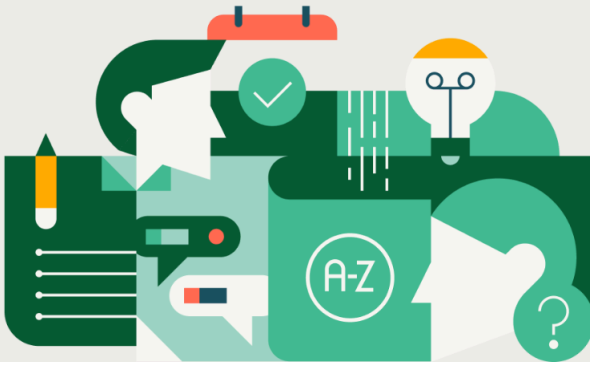
databricks

Databricks Exam 가이드

Databricks Certified 데이터 엔지니어 Associate



시험 가이드 피드백 제공

정보 업데이트

업데이트	날짜
시험 출시	2026년 5월 4일
시험 가이드 업데이트	2026년 8월 6일

시험 날짜 2주 전에 당사 웹 페이지의 시험 가이드를 확인하여 현재 버전의 시험을 준비하고 있는지 확인하십시오.

이 시험 가이드의 목적

이 시험 가이드의 목적은 시험 및 시험 범위에 대한 개요를 제공하여 시험 준비 상태를 파악하는 데 도움을 드리는 것입니다. 이 문서는 시험에 변경 사항이 있을 때마다 업데이트되며(변경 사항이 적용되는 시기도 함께 표시됩니다), 이를 통해 확신을 갖고 준비하실 수 있습니다. 이 가이드는 위에 명시된 발효일 기준으로 현재 시행 중인 시험 버전을 다룹니다.

대상 독자 설명

Databricks Certified Data Engineer Associate 인증 시험은 Databricks Platform을 사용하여 기본적인 데이터 엔지니어링 작업을 수행하는 개인의 역량을 평가합니다. 여기에는 Databricks Platform, 워크스페이스, 아키텍처 및 기능에 대한 이해와 함께 데이터 수집, 데이터 로딩, 데이터 변환 및 모델링과 관련된 작업이 포함됩니다. 예를 들어 PySpark를 사용하여 추출, 변환, 로드(ETL) 작업을 수행하고, Lakeflow Jobs 및 CI/CD를 활용하는 능력이 이에 해당합니다. 또한 이 시험은 문제 해결, 모니터링 및 최적화 기법에 대한 이해와 Databricks Platform 내에서 거버넌스 및 보안을 달성하는 방법에 대한 지식도 평가합니다.

시험 소개

항목	세부 정보
채점 문항 수	채점되는 객관식 45문항
채점 제외 문항 포함 가능성	최대 10개의 채점 제외 문항이 포함될 수 있음(아래 "채점 제외 콘텐츠" 참조)
표시될 수 있는 총 문항 수	채점 문항 45개 및 채점 제외 문항
제한 시간	90분(채점 제외 문항을 고려한 추가 시간이 이미 반영됨)
응시료	USD 200, 현지 법률에 따라 부과되는 세금 별도
시험 응시 방식	온라인 감독 또는 시험 센터
제공 언어	영어, 일본어, 브라질 포르투갈어(pt-BR), 한국어
시험 보조 도구	허용되지 않음
사전 요건	필수 요건 없음, 관련 교육 이수 적극 권장
권장 경험	명시된 시험 개요에 따라 Databricks Platform에서의 6개월 실무 경험이 적극 권장됩니다
유효 기간	2년

재인증**:** 인증된 상태를 유지하려면 **2**년마다 재인증이 필요합니다**.** 재인증을 받으려면 **Data Engineer Associate Renewal Exam** 또는 현재 정규 시험을 응시할 수 있습니다**.**

채점되지 않는 콘텐츠**:** 시험에는 향후 활용을 위한 통계적 정보 수집을 목적으로 채점되지 않는 문항이 포함될 수 있습니다**.** 해당 문항은 시험에서 별도로 표시되지 않으며 점수에 영향을 미치지 않고**,** 이를 어카운트에 반영하여 제한 시간에 추가 시간이 포함됩니다**.**

권장 준비 사항

학습

- 강사 주도**: Databricks**를 활용한 데이터 엔지니어링
- 자기 주도 학습 **(Databricks Academy**에서 이용 가능**):**
    - LakeFlow Connect를 사용한 데이터 수집
    - Lakeflow Jobs로 워크로드 배포하기
    - 데이터 엔지니어링을 위한 DevOps 핵심 개념
    - Apache Spark™ Declarative Pipelines로 데이터 파이프라인 구축
    - 거버넌스 시작하기

AI Prep 가이드 - 모든 Databricks 시험

Databricks AI Prep Guide 가이드는 이미 사용 중인 AI 챗봇을 Databricks 자격증 시험을 위한 맞춤형 튜터로 활용하는 방법을 안내합니다. 공식 시험 가이드를 기반으로 프롬프트를 설정하고, AI가 제공하는 오래된 답변을 걸러내며, 취약한 부분을 파악하고, Databricks 시험에서 요구하는 실습으로 넘어갑니다.

실습으로 넘어갑니다

무엇을 연습해야 할지 모르시겠습니까? AI Prep 가이드(위의 AI Prep 가이드 섹션 참조)에는 "5단계 — 실습 최소 요건" 아래에 AI 챗봇이 이 시험의 목표에 맞춰 6~10개의 실습 과제로 구성된 맞춤형 체크리스트를 생성하는 프롬프트가 포함되어 있으며, 각 과제는 Databricks Free Edition에서 완료할 수 있을 만큼 작은 단위로 구성되어 있습니다.

Databricks 문서

준비하는 동안, 이 시험에서 다루는 플랫폼 영역에 대한 공식 Databricks 문서를 학습하십시오. 이는 가장 최신의 권위 있는 소스입니다. 각 주제를 얼마나 깊이 학습할지 판단하고, 오래된 자료가 아닌 최신의 더 이상 사용되지 않는 기능을 학습하고 있는지 확인하는 데 활용하십시오.Databricks documentation

시험 개요

섹션 1: Databricks Intelligence Platform (6%)

- 아키텍처, Delta Lake, Unity Catalog 등 Databricks Platform의 핵심 구성 요소를 이해합니다.
- Databricks Platform의 compute 서비스(특성, 제한 사항, 비용 모델 포함)를 이해하고, 각 워크로드 사용 사례에 가장 적합한 옵션을 선택합니다.

섹션 2: 데이터 수집 및 로딩 (21%)

- 배치, 스트리밍, 증분 로딩을 포함한 데이터 수집 패턴을 활성화하고 상세히 설명하며, 로컬 파일, LakeFlow Connect 표준 커넥터, LakeFlow Connect 관리형 커넥터와 같은 소스에서 데이터를 가져옵니다.
- COPY INTO 명령을 사용하여 클라우드 오브젝트 스토리지(ADLS/S3/GCS)에서 Unity Catalog 관리 테이블로 파일을 증분 로드합니다.
- 배치 모드(예: 디렉터리 목록 또는 파일 알림)에서 스키마 준수 및 스키마 진화를 적용한 Auto Loader를 사용하여 Unity Catalog가 관리하는 테이블에 데이터를 적재합니다.
- 다양한 엔터프라이즈 소스에서 Unity Catalog가 관리하는 테이블로 데이터를 안정적으로 수집하도록 Lakeflow Connect를 구성합니다.

- Notebooks에서 JDBC/ODBC 또는 REST 클라이언트를 사용하여 데이터를 클라우드 스토리지에 저장하거나 Unity Catalog로 관리되는 테이블에 직접 저장하며, 일반적으로 Lakeflow Jobs를 통해 오케스트레이션 및 스케줄링됩니다.
- 데이터 볼륨, 수집 빈도, 데이터 유형, Unity Catalog를 활용한 거버넌스 요구 사항 등의 기술적 요구 사항을 기반으로 Auto Loader, Lakeflow Connect(표준 및 관리형 커넥터), 파트너 커넥터, 기타 수집 방법 간의 우선순위를 결정합니다.
- Lakeflow Connect 및 기타 관리형 커넥터를 통해 반정형 및 비정형 데이터(예: JSON 및 중첩 데이터)를 Unity Catalog가 관리하는 Delta 테이블로 수집합니다.

Section 3: Data Transformation and Modeling (22%)

- PySpark/SQL을 사용하여 브론즈 테이블을 읽고, null 값을 정리하고, 데이터 유형을 표준화한 후 새로운 실버 테이블에 쓰는 방식으로 데이터 정제를 구현합니다.
- Inner join, left join, broadcast join, multiple keys, cross join, union, union all 등의 운영을 사용하여 DataFrames를 결합합니다.
- 열 추가, 삭제, 분할, 이름 변경, 필터 적용, 배열 분해(exploding) 등을 통해 열, 행 및 테이블 구조를 조작합니다.
- DataFrames에서 count, approximate count distinct, mean, summary 등의 데이터 중복 제거 운영 및 집계 운영을 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 views, views, Streaming Tables, 테이블과 같은 골드 레이어 객체 간의 차이점을 이해하고 이를 구축하는 방법을 학습합니다.
- 데이터 품질 검사 및 유효성 검사 규칙을 신청하여 신뢰할 수 있는 실버 및 골드 데이터셋을 보장합니다.

섹션 4: Lakeflow Jobs 활용 (16%)

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

섹션 5: CI/CD 구현 (10%)

- Databricks Git folders(이전 명칭: Databricks Repos)에서 브랜치 생성 및 전환, 변경 사항 커밋 및 푸시, Databricks Git 통합을 사용한 pull request 생성 등 Databricks 워크스페이스 UI 내에서 코드 개발 워크플로를 관리합니다.
- 동일한 코드베이스를 dev, test, prod 대상으로 승격하면서 Declarative Automation Bundle(이전 명칭: Databricks Asset Bundles) 변수 및 재정의를 사용하여 환경별 구성을 이해합니다.
- Declarative Automation Bundles(이전 명칭: Databricks Asset Bundles)를 사용하여 Lakeflow Jobs, Lakeflow Declarative Pipelines 및 기타 워크스페이스 자산을 패키지화하고, 구성하며, dev, test, prod 환경 전반에 걸쳐 배포합니다.
- 자동화된 CI/CD Workflows에서 Declarative Automation Bundles(이전 명칭: Databricks Asset Bundles) 및 기타 워크스페이스 자산을 검증, 배포, 관리하기 위해 Databricks CLI를 이해합니다.

섹션 6: 문제 해결, 모니터링 및 최적화 (10%)

- Lakeflow Jobs 실행 이력 뷰를 사용하여 현재 실행 시간을 과거 기준선과 비교함으로써 작업 성능의 추세를 파악합니다.
- Lakeflow Jobs UI를 사용하여 작업 상태를 해석하고, DAG 기반 태스크 그래프를 통해 업스트림 블로커를 파악하며, 파이프라인 실행 시간과 실패율을 추적함으로써 파이프라인 상태를 모니터링합니다.
- Spark UI의 스테이지 수준 메트릭을 해석하여 데이터 스큐, 셔플링, 디스크 스필링과 같은 일반적인 성능 병목 현상을 식별합니다.
- Liquid 클러스터링과 Predictive Optimization의 기능을 이해합니다.
- 클러스터 스타트업 실패, 라이브러리 충돌 및 메모리 부족 문제를 진단합니다.

섹션 7: 거버넌스 및 보안 (15%)

- Unity Catalog에서 관리형 테이블과 외부 테이블의 차이를 구분하고, 이에 대한 기본 운영(생성, 수정, 삭제, 관리형 테이블과 외부 테이블 간 변환)을 수행합니다.
- UI와 SQL을 사용하여 보안 계층 구조의 적절한 수준에서 주체(사용자, 그룹, 서비스 주체)에게 GRANT, REVOKE, DENY 권한을 적용하여 액세스 제어를 구성합니다.
- 사용자 그룹을 기반으로 데이터 가시성을 제한하기 위해 열 수준 마스킹과 행 수준 보안을 이해합니다.
- 민감한 데이터에 대한 행 수준 필터링 및 열 마스킹을 중앙에서 제어하기 위해 Unity Catalog ABAC 정책을 이해합니다.

샘플 문제

샘플 문제의 목적은 시험 가이드에 명시된 목표를 보여주고 각 목표에 부합하는 샘플 문제를 제공하는 것입니다. 시험 가이드에는 시험에서 다룰 수 있는 목표가 나열되어 있으며, 가장 좋은 준비 방법은 위의 시험 날짜에 해당하는 전체 시험 개요를 검토하는 것입니다.

질문 **1**

목표**: Spark UI**에서 스테이지 수준 메트릭을 해석하여 데이터 스큐**,** 셔플링**,** 디스크 스필링과 같은 일반적인 성능 병목 현상을 식별합니다**.**

데이터 엔지니어가 새로운 데이터 소스가 온보딩된 후 배치 작업의 기간이 두 배로 늘어난 것을 발견했습니다. 해당 작업은 `customer_id` 키에 대해 wide groupBy 집계를 수행합니다. Spark UI에서 가장 긴 스테이지를 확인하면 대부분의 태스크는 30초 이내에 완료되지만, 하나의 태스크는 10분 이상 실행됩니다. 해당 스테이지의 태스크 요약에서 Min/Median shuffle read는 약 400 MB인 반면, Max shuffle read는 5 GB를 초과합니다. Adaptive Query Execution은 이미 default 설정으로 활성화되어 있지만, 데이터 편향 태스크는 계속 지속됩니다.

어떤 솔루션이 작업 런타임을 줄입니까?

A. 느린 태스크가 더 빨리 완료될 수 있도록 클러스터 크기를 늘려 executor를 추가합니다
B. AQE 스큐 조인 처리에 의존하여 런타임에 과도하게 큰 파티션을 자동으로 분할합니다
C. spark.sql.shuffle.partitions를 줄여 더 적은 태스크로 더 많은 작업을 병합합니다
D. 집계 키에 솔트 키를 추가하여 편향된 값이 여러 태스크에 분산되도록 합니다

질문 2

목표**: Databricks Platform**의 아키텍처**, Delta Lake, Unity Catalog** 등 핵심 구성 요소를 이해합니다**.**

데이터 엔지니어는 잘못된 수집 이후 안정적인 롤백을 유지하고, 규정 준수를 위한 감사 추적을 보장하며, AI 및 BI 워크로드 모두에 단일 소스 오브 트루스에 대한 일관된 액세스를 제공하면서 신속한 파이프라인 반복 작업을 수행해야 합니다.

데이터 엔지니어가 이러한 요구 사항을 충족하기 위해 사용해야 하는 전략은 무엇입니까?

A. 수동 파일 버전 관리 및 롤백을 위한 야간 복사본을 사용하는 DBFS CSV 스토리지.
B. Delta Lake ACID 트랜잭션 및 시간 여행 기능을 Unity Catalog를 통해 일관된 액세스와 리니지를 관리합니다.
C. 클라우드 오브젝트 스토리지만 사용하며, 복구 및 거버넌스를 위한 임시 SQL 쿼리를 지원합니다.
D. 감사 추적 및 BI 배포를 위한 임시 인메모리 DataFrames입니다.

질문 3

목표**:** 배치**,** 스트리밍**,** 증분 로딩을 포함한 데이터 수집 패턴을 활성화하고 상세히 설명하며**,** 로컬 파일**, LakeFlow Connect** 표준 커넥터**, LakeFlow Connect** 관리형 커넥터 등의 소스에서 데이터를 가져옵니다**.**

데이터 엔지니어가 고객 소유의 S3 버킷에서 Databricks 감사 로그를 사용하는 다운스트림 파이프라인을 구축하고 있습니다. 스키마 추론 및 체크포인팅을 구현하기 전에, 전달 형식, 일반적인 수집 지연 시간, 그리고 파일이 덮어쓰여질 수 있는지 여부를 파악하고자 합니다.

Databricks 감사 로그 스토리지 동작은 무엇입니까?

A. 파일은 JSON으로 전달되며, 일반적인 이벤트 로깅은 전달 시작 후 15분 이내에 이루어지고, 새로운 전달이 기존 파일을 덮어쓸 수 있습니다.
B. 파일은 1분 미만의 지연 시간 보장과 함께 CSV로 전달되며, 불변성을 유지하기 위해 파일이 작성된 후에는 덮어쓰기가 발생하지 않습니다.
C. 파일은 Parquet 형식으로 제공되며 24시간 이후에는 최종 일관성이 보장되고, 스트리밍 수집을 단순화하기 위해 덮어쓰기가 비활성화되어 있습니다.
D. 파일은 JSON 형식으로 제공되며 주 단위 배치 주기로 전달되고, 덮어쓰기 시 이전 콘텐츠가 추가 없이 완전히 교체됩니다.

질문 4

목표**: Databricks Platform**의 **compute** 서비스**(**특성**,** 제한 사항**,** 비용 모델 포함**)**를 이해하고**,** 각 워크로드 사용 사례에 가장 적합한 옵션을 선택합니다**.**

데이터 엔지니어링 팀은 하루 종일 정제된 Delta 테이블에 대해 임시 SQL 쿼리를 실행하는 여러 비즈니스 분석가를 지원합니다. 팀은 효율적인 쿼리 성능, 빠른 클러스터 스타트업, 다수의 동시 사용자 지원을 보장하는 동시에, 매우 큰 클러스터로의 불필요한 확장을 방지하여 비용을 관리해야 합니다.

이러한 요구 사항을 충족하는 클러스터 구성은 무엇입니까?

A. 예약된 ETL Workflows를 위해 Autoscaling이 설계된 작업 클러스터
B. 고정된 수의 작업자 노드로 구성된 범용 클러스터
C. Autoscaling이 활성화된 Serverless SQL Warehouse
D. 경량 개발 작업을 위해 구성된 단일 노드 클러스터

질문 5

목표**: Declarative Automation Bundles(**이전 명칭**: Databricks Asset Bundles)**를 사용하여 **Lakeflow Jobs, Lakeflow Declarative Pipelines** 및 기타 워크스페이스 자산을 패키지화하고**,** 구성하며**, dev, test, prod** 환경 전반에 걸쳐 프로모션합니다**.**

한 팀이 Databricks에서 ETL 파이프라인을 배포, 버전 관리 및 오케스트레이션하는 모듈식 방법을 원하며, CI/CD와 반복 가능성을 지원하고자 합니다.

이 요구 사항을 지원하는 기능은 무엇입니까?

A. Unity Catalog의 모델을 사용하여 ETL 작업을 나타내며, 각 모델은 파이프라인 코드 아티팩트를 저장하고 CI/CD는 작업 태스크에 연결된 모델 별칭을 업데이트하여 버전을 승격합니다.
B. 변환 로직을 Unity Catalog Volumes에 저장된 wheel 라이브러리로 패키지화하고, 이를 Lakeflow Jobs 태스크에 바인딩하여 환경 전반에 걸쳐 결정론적 배포를 보장합니다.
C. 패키지 API 로직을 Volume에 마운트된 노트북 안에 넣고, Jobs API v2를 사용하여 노트북을 트리거하되, 버전 관리 시스템으로서 노트북 수정 이력에 의존합니다.
D. Declarative Automation Bundles(DABs)를 사용하여 리소스와 코드 자산을 정의하고, Git에서 버전을 관리하며, 자동화된 CI/CD 작업을 통해 환경 전반에 걸쳐 배포를 승격합니다.

Question 6

목표**:** 자동화된 **CI/CD** 워크플로에서 **DAB** 및 기타 워크스페이스 자산을 검증**,** 배포 및 관리하기 위한 **Databricks CLI** 사용 방법을 이해합니다**.**

데이터 엔지니어가 Declarative Automation Bundles를 사용하여 외부 데이터베이스에 연결하는 작업을 배포하고 있습니다. 보안 모범 사례를 따르기 위해, 데이터베이스 비밀번호는 databricks.yml 파일이나 노트북 소스 코드에 하드코딩되어서는 안 됩니다.

데이터 엔지니어는 자동화된 배포 중에 이 시크릿을 안전하게 처리하기 위해 번들을 어떻게 구성해야 합니까?

A. databricks.yml의 variables 블록에 변수를 정의하고, `BUNDLE_VAR_`를 접두사로 하는 환경 변수를 통해 해당 값을 제공합니다.
B. 번들 루트에 passwords.txt라는 일반 텍스트 파일에 비밀번호를 저장하고 배포에 포함시킵니다.
C. Git에 커밋하기 전에 Databricks bundle encrypt 명령을 사용하여 문자열을 난독화합니다.
D. 노트북 내부에서 직접 표준 Spark 구성인 spark.conf.set("db_pass", "...")을 사용합니다.

질문 7

목표**:** 사용자의 그룹에 따라 데이터 가시성을 제한하기 위해 열 수준 마스킹 및 행 수준 보안을 신청합니다**.**

데이터 엔지니어는 'finance_team' 그룹의 멤버만 'credit_card_number' 열의 전체 값을 볼 수 있도록 해야 합니다. 다른 모든 사용자는 마스킹된 버전을 볼 수 있어야 합니다.

이 요구 사항을 구현하려면 어떤 Unity Catalog 기능과 구성을 사용해야 합니까?

A. 각 사용자 그룹에 대해 값이 마스킹된 별도의 Delta 테이블을 생성합니다.
B. Delta 테이블 속성 내 특정 열에서 AWS IAM 역할을 활성화합니다
C. is_account_group_member('finance_team') 로직을 사용하여 열 마스킹 함수를 신청합니다
D. 재무 팀을 제외한 모든 사용자에 대해 특정 열에 DENY 권한을 설정합니다

Question 8

목표**: Unity Catalog**에서 관리형 테이블과 외부 테이블의 차이를 구분하고 기본 운영을 수행합니다**.**

데이터 엔지니어가 외부 테이블을 관리형 테이블로 변환합니다. 얼마 후, 팀은 문제를 발견하고 변환을 되돌려야 합니다.

최신 런타임에서 테이블을 이전 유형으로 복원하는 작업은 무엇입니까?

A. ALTER TABLE ... UNSET MANAGED를 실행하여 외부 테이블로 되돌립니다
B. 롤백이 지원되지 않으므로 관리형 테이블을 삭제하고 원래 LOCATION을 사용하여 외부 테이블로 다시 생성합니다.
C. 데이터를 이동하지 않고 테이블 유형을 되돌리려면 ALTER TABLE ... SET EXTERNAL을 실행합니다.
D. RESTORE TABLE ... TO VERSION AS OF를 사용하여 Delta 시간 여행 이력에서 테이블을 복원합니다.

질문 9

목표**: Lakeflow Jobs UI**를 사용하여 작업 상태를 해석하고 태스크 그래프를 확인함으로써 파이프라인 상태를 모니터링합니다**.**

Databricks 작업 태스크가 45분 실행 후 '시간 초과(Timed Out)' 상태로 실패합니다. 데이터 엔지니어는 처리할 데이터 양이 증가하여 해당 태스크가 성공적으로 완료되는 데 최대 60분이 걸릴 수 있다고 판단하며, 향후 실행이 자동으로 종료되지 않고 완료되기를 원합니다.

데이터 엔지니어가 목표를 달성하기 위해 수정해야 하는 구성은 무엇입니까?

A. 업스트림 태스크에 대한 종속성 조건
B. 태스크 실행 기간의 임계값
C. 작업의 최대 동시의 실행 수 설정
D. 컴퓨트 리소스의 클러스터 자동 종료 설정

질문 10

목표**:** 트리거 유형에 대한 이해를 바탕으로 **Lakeflow Jobs**를 사용하여 작업 일정을 구현합니다**.**

데이터 엔지니어링 팀은 서드파티 분석 벤더로부터 클릭스트림 내보내기 파일을 받습니다. 벤더는 압축된 JSON 파일을 예측할 수 없는 간격으로 외부 클라우드 스토리지 위치에 업로드합니다. 한 시간 내에 여러 파일이 업로드되기도 하고, 하루 대부분 동안 아무것도 업로드되지 않기도 합니다. Lakeflow Job은 Auto Loader로 새 파일을 읽고, 데이터를 보강한 후 결과를 Delta 테이블에 병합하는 배치 노트북을 실행합니다. 팀은 새 파일이 도착한 직후 처리되기를 원하며, 운영자가 스토리지 위치를 모니터링하지 않아도 되고, 긴 유휴 기간 동안 compute 비용이 발생하거나 새 데이터가 없는 상태에서 실행이 시작되는 일이 없기를 원합니다.

이 Lakeflow Job에는 어떤 트리거 유형을 구성해야 합니까?

A. 새 내보내기 파일을 폴링하기 위해 5분마다 실행되는 예약된 트리거

B. 공급업체의 외부 스토리지 위치에 대한 파일 도착 트리거
C. 작업을 항상 실행 상태로 유지하는 연속 트리거
D. 공급업체 업로드 알림이 도착할 때마다 운영자가 수동으로 시작하는 트리거

정답 키 및 해설

#	정답	설명
1	D	중앙값 약 400MB 대비 최대 셔플 읽기가 5GB라는 것은 전형적인 키 스큐(skew)로, 하나의 핫한 customer_id가 단일 파티션을 지배하는 경우입니다. AQE의 스큐 처리는 스큐된 조인(sort-merge/shuffle-hash)만 최적화하고 groupBy 집계는 최적화하지 않으므로, AQE가 켜져 있어도 이 스테이지를 해소하지 못합니다(B 제거). 실행기(executor)를 추가해도 거대한 단일 태스크가 혼자 실행되는 상태는 그대로이고(A), shuffle.partitions를 낮추면 파티션이 더 커져 스큐가 악화됩니다(C). 집계 키에 솔팅(salting)을 적용하면 핫 키가 여러 태스크로 분산되며, 이후 2차 패스로 부분 결과를 결합합니다.
2	B	Delta Lake는 ACID 트랜잭션과 타임 트래블(time travel)을 제공하여 잘못된 수집 이후 안정적인 롤백과 완전한 감사 이력을 제공하며, Unity Catalog는 액세스와 계보(lineage)를 관리하여 AI 및 BI 워크로드가 단일 진실 공급원(single source of truth)을 읽도록 합니다. 다른 선택지는 수동 복사본이나 거버넌스가 적용되지 않은 스토리지에 의존합니다.
3	A	Databricks는 감사 로그를 JSON 파일로 전달하며, 전달이 시작되면 일반적으로 이벤트 발생 후 15분 이내에 제공되고, 이후 전달이 기존 파일을 덮어쓸 수 있습니다. 따라서 수집 파이프라인은 파일이 불변이라고 가정하지 말고 덮어쓰기를 허용해야 합니다.
4	C	자동 확장이 적용된 서버리스 SQL 웨어하우스는 다수의 동시 대화형 SQL 사용자를 위해 설계되어, 빠른 시작과 쿼리 격리를 제공하면서 수요에 따라서만 확장됩니다. 잡 클러스터(Job cluster)는 예약된 ETL에 적합하며, 고정형 또는 단일 노드 클러스터는 동시성과 비용의 균형을 맞출 수 없습니다.
5	D	Declarative Automation Bundles는 잡, 파이프라인, 코드 자산을 Git에서 버전 관리되는 구성으로 선언하므로, 동일한 정의가 자동화된 CI/CD를 통해 개발, 테스트, 운영 환경으로 승격됩니다. 다른 선택지는 아티팩트 저장소나 노트북 수정 이력에 의존하며, 둘 다 오케스트레이션 자체를 버전 관리하지 않습니다.
6	A	DABs는 variables 블록을 통한 매개변수화를 지원합니다. BUNDLE_VAR_<variable_name> 접두사를 사용하면 CLI가 배포 시점에 환경(예: GitHub Secrets)에서 값을 번들에 자동으로 주입하여, 비밀 정보가 소스 코드나 YAML 파일에 노출되지 않도록 보장합니다.
7	C	is_account_group_member() 함수를 사용하는 열 마스킹(column masking)이 올바른 선택입니다. 동일한 테이블을 여러 사용자가 쿼리하더라도 그룹 멤버십에 따라 민감한 데이터를 선택적으로 가리는 동적 세분화 보안을 제공하기 때문입니다.
8	A	UNSET MANAGED는 문서화된 롤백 방법입니다. 테이블 메타데이터를 원래의 외부 위치를 가리키도록 업데이트하며, 변환 후 14일 이내에만 사용할 수 있습니다. 변환 이후 관리형 위치에 기록된 쓰기는 보존되며, 변환과 롤백 사이의 커밋은 버전 기준 타임 트래블은 지원하지만 타임스탬프 기준은 지원하지 않습니다. Databricks는 롤백 후 7일이 지나면 관리형 위치의 데이터를 삭제하며, 스트리밍 잡은 다시 시작해야 합니다.
9	B	"Timed Out" 상태는 실행이 태스크에 설정된 제한 시간 임계값(여기서는 45분)을 초과했음을 의미합니다. 이를 60분 이상(여유를 두어 예: 90분)으로 높이면 이후 실행이 중단되지 않고 완료될 수 있습니다.

10	B	파일 도착 트리거(file arrival trigger)는 외부 위치를 모니터링하여 새 파일이 나타날 때만 실행을 시작합니다. 자동화되어 있고 지연 시간이 짧으며, 위치가 변경되지 않는 동안에는 컴퓨팅 자원을 사용하지 않습니다.