databricks

Databricks 시험 가이드

Databricks 인증 데이터 엔지니어 어소시에이트

시험 가이드 피드백 제공

이 시험 가이드의 목적

이 시험 가이드의 목적은 시험의 개요를 제공하고 시험에서 다루어질 주요 내용을 안내하여 시험에 대비할 수 있도록 돕기 위한 것입니다. 이 문서는 시험에 변경 사항이 있을 때마다(그리고 해당 변경 사항이 시험에 적용될 때) 업데이트되므로 그에 맞춰 준비할 수 있습니다. **이 버전은 2024 년 1 월 1 일 현재 제공 중인 라이브 시험에 적용됩니다. 시험일 2 주 전에 다시 확인하여 최신 버전을 확인하시기 바랍니다.**

오디언스 설명

Databricks 인증 데이터 엔지니어 어소시에이트 인증 시험은 Databricks 레이크하우스 플랫폼을 활용하여 입문 수준의 데이터 엔지니어링 작업을 완료하는 개인의 역량을 평가합니다. 여기에는 레이크하우스 플랫폼 및 해당 워크스페이스, 아키텍처 및 기능에 대한 이해가 포함됩니다. 또한 배치 및 증분 처리 패러다임 모두에서 Apache Spark SQL 및 Python 를 사용하여 다중 홉 아키텍처 ETL 작업을 수행하는 역량을 평가합니다. 시험에서는 마지막으로 엔터티 권한을 유지하면서 기본 ETL 파이프라인과 Databricks SQL 쿼리 및 대시보드를 프로덕션에 배치하는 테스터의 역량을 평가합니다. 이 인증 시험을 통과한 개인은 Databricks 및 관련 도구를 사용하여 기본 데이터 엔지니어링 작업을 완료할 수 있는 역량을 갖춘 것으로 기대됩니다.

시험에 대한 정보

- 항목 수: 45 개의 객관식 문제
- 제한 시간: 90 분
- 등록비: 미화 200 달러 + 현지법에 따라 적용 가능한 세금
- 제공 방법: 온라인 감독
- 테스트 보조: 허용되지 않음.
- 선수 조건: 요구 조건 없음. 강의 출석 및 6 개월의 Databricks 사용 실습이 권장됨
- 유효 기간: 2 년
- 채점되지 않는 콘텐츠: 시험에는 향후 사용을 위한 통계 정보를 수집하기 위해 채점되지 않는 항목이 포함될 수 있습니다. 이러한 항목은 양식에서 식별되지 않으며 점수에 영향을 미치지 않습니다. 이러한 콘텐츠에 대해서는 적절한 추가 시간이 고려됩니다.

권장 교육 과정

- 강사 주도형: Databricks 를 이용한 데이터 엔지니어링
- 자기 주도형: Databricks 를 이용한 데이터 엔지니어링(Databricks Academy 에서 사용 가능)

시험 개요

섹션 1: Databricks 레이크하우스 플랫폼

- 데이터 레이크하우스와 데이터 웨어하우스의 관계를 설명합니다.
- 데이터 레이크 대비 데이터 레이크하우스의 데이터 품질 향상을 확인합니다.
- 실버 테이블과 골드 테이블을 비교 및 대조하여 브론즈 테이블을 소스로 사용할 워크로드와 골드 테이블을 소스로 사용할 워크로드를 확인합니다.
- Databricks 플랫폼 아키텍처의 요소를 파악하여, Compute Plane 에 위치한 요소와 Control Plane 에 위치한 요소를 비교하고 고객의 클라우드 계정에 존재하는 요소를 확인합니다.
- 다목적 클러스터와 작업 클러스터를 구분합니다.
- Databricks 런타임을 사용한 클러스터 소프트웨어의 버전 관리 방법을 확인합니다.
- 클러스터를 필터링하여 사용자가 액세스할 수 있는 클러스터를 표시하는 방법을 확인합니다.
- 클러스터를 종료하는 방법과 클러스터 종료가 미치는 영향을 설명합니다.
- 클러스터 재시작이 도움이 되는 시나리오를 확인합니다.
- 동일 노트북 내에서 여러 언어를 사용하는 방법을 설명합니다.
- 노트북을 다른 노트북 내에서 실행하는 방법을 확인합니다.
- 노트북을 다른 사용자와 공유하는 방법을 확인합니다.
- Databricks Repos 로 Databricks 내 워크플로 활성화 방법을 설명합니다.
- Databricks Repos 를 통해 사용 가능한 Git 기능을 확인합니다.
- Repos 와 관련하여 Databricks 노트북 버전 제어 기능의 제한사항을 확인합니다.

섹션 2: Apache Spark 를 사용한 ELT

- 단일 파일 및 파일 디렉터리에서 데이터를 추출합니다.
- 데이터 유형으로 FROM 키워드 다음에 포함된 접두사를 확인합니다.
- 파일에 대한 참조로 view, temporary view 및 CTE 를 만듭니다.
- 외부 소스의 테이블은 Delta Lake 테이블이 아님을 확인합니다.
- JDBC 연결 및 외부 CSV 파일에서 테이블을 만듭니다.
- count_if 함수와 count where x is null 의 사용 방법을 확인합니다.
- count(row)가 NULL 값을 건너뛰는 방법을 확인합니다.
- 기존 Delta Lake 테이블의 중복 행을 삭제합니다.
- 중복 행을 삭제하면서 기존 테이블에서 새 테이블을 만듭니다.
- 특정 열을 기반으로 중복 행을 삭제합니다.
- 기본 키가 모든 행 전반에서 고유한지 확인합니다.
- 필드가 또 다른 필드의 단일 고윳값과 연결되어 있는지 확인합니다.
- 특정 필드에 값이 없는지 확인합니다.

- 열을 타임스탬프로 캐스팅합니다.
- 타임스탬프에서 캘린더 데이터를 추출합니다.
- 기존 문자열 열에서 특정 패턴을 추출합니다.
- 도트 구문을 활용하여 중첩 데이터 필드를 추출합니다.
- 배열 함수 사용의 이점을 확인합니다.
- JSON 문자열을 구조체로 파싱합니다.
- join query 를 기반으로 반환되는 결과를 확인합니다.
- flatten 함수 대비 explode 함수의 사용 시나리오를 확인합니다.
- wide format 에서 long format 으로의 데이터 변환 방법으로 PIVOT 절을 확인합니다.
- SQL UDF 를 정의합니다.
- 함수의 위치를 확인합니다.
- SQL UDF 공유를 위한 보안 모델을 설명합니다.
- SQL 코드에 CASE/WHEN 을 사용합니다.
- 사용자 지정 제어 흐름에 CASE/WHEN 을 활용합니다.

섹션 3: 증분 데이터 처리

- Delta Lake 에서 ACID 트랜잭션을 제공하는 경우를 확인합니다.
- ACID 트랜잭션의 이점을 확인합니다.
- 트랜잭션의 ACID 준수 여부를 확인합니다.
- 데이터와 메타데이터를 비교 및 대조합니다.
- 관리형 테이블과 외부 테이블을 비교 및 대조합니다.
- 외부 테이블 사용을 위한 시나리오를 확인합니다.
- 관리형 테이블을 만듭니다.
- 테이블의 위치를 확인합니다.
- Delta Lake 파일의 디렉터리 구조를 검사합니다.
- 테이블의 이전 버전 작성자를 확인합니다.
- 테이블 트랜잭션 기록을 검토합니다.
- 테이블을 이전 버전으로 롤백합니다.
- 테이블을 이전 버전으로 롤백할 수 있다는 것을 확인합니다.
- 테이블의 특정 버전을 query 합니다.
- Zordering 이 Delta Lake 테이블에 유리한 이유를 확인합니다.
- vacuum 이 delete 를 커밋하는 방법을 확인합니다.
- Optimize 에서 압축하는 파일의 종류를 확인합니다.
- CTAS 를 솔루션으로서 확인합니다.
- 생성된 열을 만듭니다.
- 테이블 코멘트를 추가합니다.
- CREATE OR REPLACE TABLE 및 INSERT OVERWRITE 를 사용합니다.
- CREATE OR REPLACE TABLE 및 INSERT OVERWRITE 를 비교 및 대조합니다.
- MERGE 를 사용해야 하는 시나리오를 확인합니다.
- 작성 시 데이터 중복을 제거하는 명령으로서 MERGE 를 확인합니다.
- MERGE 명령의 이점을 설명합니다.

- COPY INTO 문으로 대상 테이블에 데이터가 복제되지 않는 이유를 확인합니다.
- COPY INTO 를 사용해야 하는 시나리오를 확인합니다.
- COPY INTO 를 사용하여 데이터를 삽입합니다.
- 새로운 DLT 파이프라인을 만드는 데 필요한 구성요소를 확인합니다.
- 파이프라인 생성 시 대상과 노트북 라이브러리의 목적을 확인합니다.
- 비용 및 지연과 관련하여 트리거 파이프라인과 연속 파이프라인을 비교 및 대조합니다.
- 자동 로더를 활용하는 소스 위치를 확인합니다.
- 자동 로더가 유용한 시나리오를 확인합니다.
- 자동 로더가 JSON 소스의 모든 데이터를 STRING 으로 추론하는 이유를 확인합니다.
- 제약 위반의 기본 동작을 확인합니다.
- 제약 위반에 대한 ON VIOLATION DROP ROW 및 ON VIOLATION FAIL UPDATE 의 영향을 확인합니다.
- CDC(Change Data Capture) 및 APPLY CHANGES INTO 의 동작을 설명합니다.
- 이벤트 로그를 query 하여 메트릭을 얻고 감사 로깅을 수행하며 리니지를 조사합니다.
- DLT 구문 문제 해결: DLT 파이프라인에서 오류가 발생한 노트북을 확인하고, create 문에서 LIVE 의 필요성을 확인하며 from 절에서 STREAM 의 필요성을 확인합니다.

섹션 4: 프로덕션 파이프라인

- 작업에서 여러 태스크 사용의 이점을 확인합니다.
- 작업에서 이전 태스크를 설정합니다.
- 이전 태스크를 설정해야 하는 시나리오를 확인합니다.
- 태스크의 실행 기록을 검토합니다.
- 스케줄링 기회로서 CRON 을 확인합니다.
- 실패한 태스크를 디버그합니다.
- 실패 시 재시도 정책을 설정합니다.
- 실패한 작업의 경우 알림을 생성합니다.
- 이메일을 통해 알림을 보낼 수 있는지 확인합니다.

섹션 5: 데이터 거버넌스

- 데이터 거버넌스의 네 가지 영역 중 하나를 확인합니다.
- 메타스토어와 카탈로그를 비교 및 대조합니다.
- Unity Catalog 보안 대상을 확인합니다.
- Service Principal 을 정의합니다.
- Unity Catalog 와 호환되는 클러스터 보안 모드를 확인합니다.
- UC-지원 범용 클러스터를 생성합니다.
- DBSQL 웨어하우스를 생성합니다.
- 3 계층 네임스페이스를 query 하는 방법을 확인합니다.
- 데이터 개체 액세스 제어를 구현합니다.
- 모범 사례로 워크스페이스와 메타스토어를 코로케이션하는 방법을 확인합니다.
- 모범 사례로 커넥션을 위해 Service Principal 을 사용하는 방법을 설명합니다.
- 모범 사례로 카탈로그 전반에서 비즈니스 단위 분리 방법을 확인합니다.

샘플 질문

다음 질문은 이전 버전의 시험에서 삭제된 문제입니다. 목적은 시험 가이드에 명시된 대로 목표를 보여주고 목표에 맞는 샘플 문제를 제공하는 것입니다. 시험 가이드에는 시험에서 다룰 수 있는 목표가 나열되어 있습니다. 인증 시험에 대비하는 가장 좋은 방법은 시험 가이드의 시험 개요를 검토하는 것입니다.

질문 1

목표: 기존 데이터 웨어하우스 대비 데이터 레이크하우스의 이점을 설명합니다.

기존 데이터 웨어하우스에서 제공되지 않는 데이터 레이크하우스의 이점은 무엇인가요?

A. 데이터 레이크하우스는 관계형 데이터 관리 시스템을 제공한다.
B. 데이터 레이크하우스는 버전 제어를 위해 데이터 스냅샷을 캡처한다.
C. 데이터 레이크하우스는 완전한 제어를 위해 스토리지와 컴퓨팅을 연결한다.
D. 데이터 레이크하우스는 데이터를 위한 전용 스토리지 형식을 활용한다.
E. 데이터 레이크하우스는 배치 및 스트리밍 분석을 모두 지원한다.

질문 2

목표: query 최적화 기법을 확인합니다.

데이터 엔지니어링 팀이 모두 동일 조건을 충족하는 행을 추출하기 위해 Delta 테이블을 query 해야 합니다. 그러나 팀은 query 가 느리게 실행되고 있음을 확인했습니다. 팀에서는 이미 데이터 파일의 크기를 조정했습니다. 조사 결과, 팀은 조건을 충족하는 행이 각 데이터 파일 전반에 걸쳐 드물게 위치한다는 결론을 내렸습니다.

query 를 가속화할 수 있는 최적화 기법은 무엇인가요?

A. Data skipping
B. Z-Ordering
C. Bin-packing
D. Parquet 파일로 쓰기
E. 파일 크기 튜닝

질문 3
목표: 실버 테이블을 소스로 활용하는 데이터 워크로드를 확인합니다.

어느 데이터 워크로드에서 실버 테이블을 소스로 활용하나요?

A. 타임스탬프를 사람이 읽을 수 있는 형식으로 파싱하여 데이터를 보강하는 작업
B. 이미 대시보드에 피드된 집계된 데이터를 query 하는 작업
C. 스트리밍 소스의 가공되지 않은 데이터를 Lakehouse 로 수집하는 작업
D. 표준 요약 통계를 생성하는 정리된 데이터를 집계하는 작업
E. 잘못된 형식의 레코드를 제거하여 데이터를 정리하는 작업

질문 4
목표: refresh 스케줄 구성 방법을 설명

엔지니어링 관리자가 Databricks SQL query 를 사용하여
고객 보고 버그 관련 픽스에 대한 팀의 진행 상황을 모니터링합니다. 관리자는 query 의 결과를 매일 검토하지만, 이들은 query 를 매일 수동으로 재실행하며 결과를 기다립니다.

query 결과가 매일 업데이트되도록 하려면 query 를 어떻게 스케줄링해야 할까요?

A. Databricks SQL 의 query 페이지에서 12 시간마다 refresh
B. Databricks SQL 의 query 페이지에서 매일 refresh
C. Jobs UI 에서 12 시간마다 실행
D. Databricks SQL 의 SQL 웨어하우스 페이지에서 매일 refresh
E. Databricks SQL 의 SQL 웨어하우스 페이지에서 12 시간마다 refresh

질문 5
목표: 적절한 권한을 허용하기 위한 명령 확인

한 회사에서 신입 데이터 엔지니어가 업무를 시작했습니다. 이 데이터 엔지니어는 최근 회사의 Databricks 워크스페이스에 new.engineer@company.com 으로 추가되었습니다. 이 데이터 엔지니어는 retail 데이터베이스에서 sale 테이블을 query 할 수 있어야 합니다. 이 신입 데이터 엔지니어에게는 이미 retail 데이터베이스에 대한 USAGE 권한이 부여되었습니다.

이 신입 데이터 엔지니어에게 적절한 권한을 허용하려면 어떤 명령을 사용해야 할까요?

A. GRANT USAGE ON TABLE sales TO new.engineer@company.com;
B. GRANT CREATE ON TABLE sales TO new.engineer@company.com;
C. GRANT SELECT ON TABLE sales TO new.engineer@company.com;
D. GRANT USAGE ON TABLE new.engineer@company.com TO sales;
E. GRANT SELECT ON TABLE new.engineer@company.com TO sales;

정답

질문 1: E

질문 2: B

질문 3: D

질문 4: B

질문 5: C