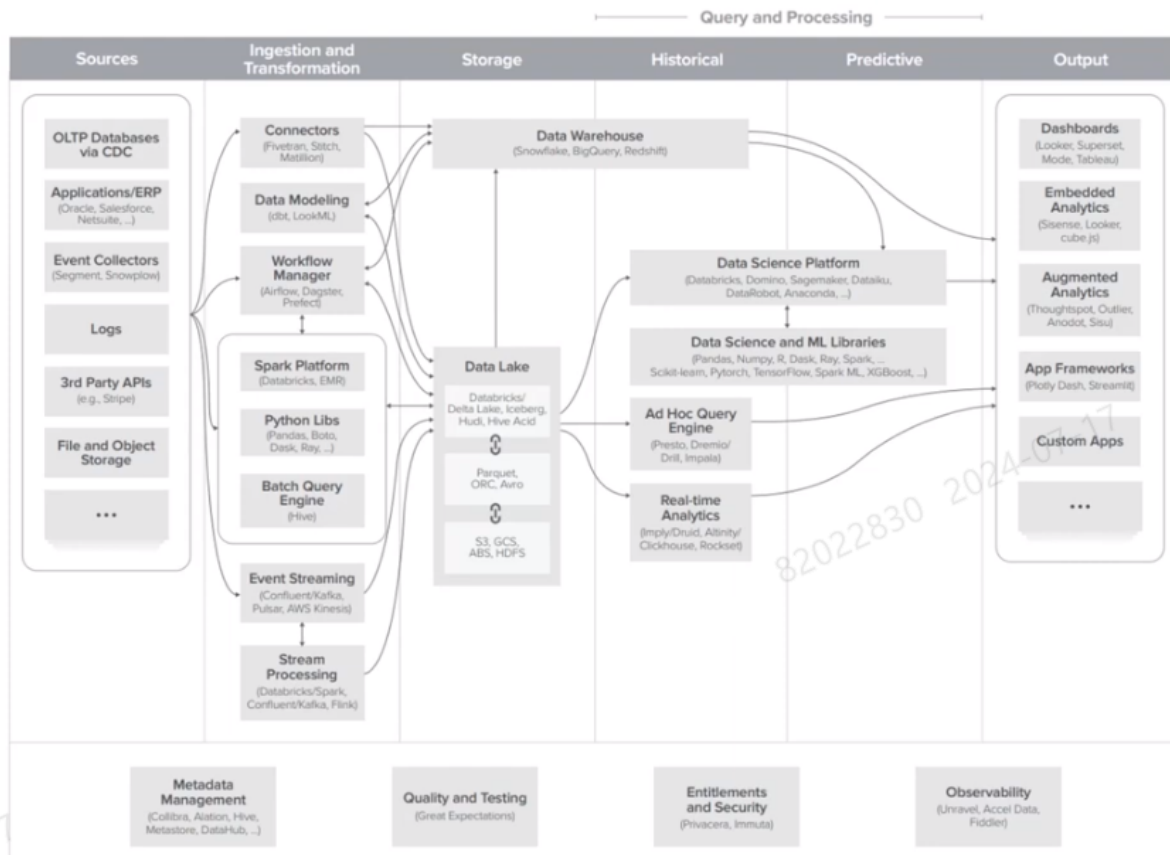


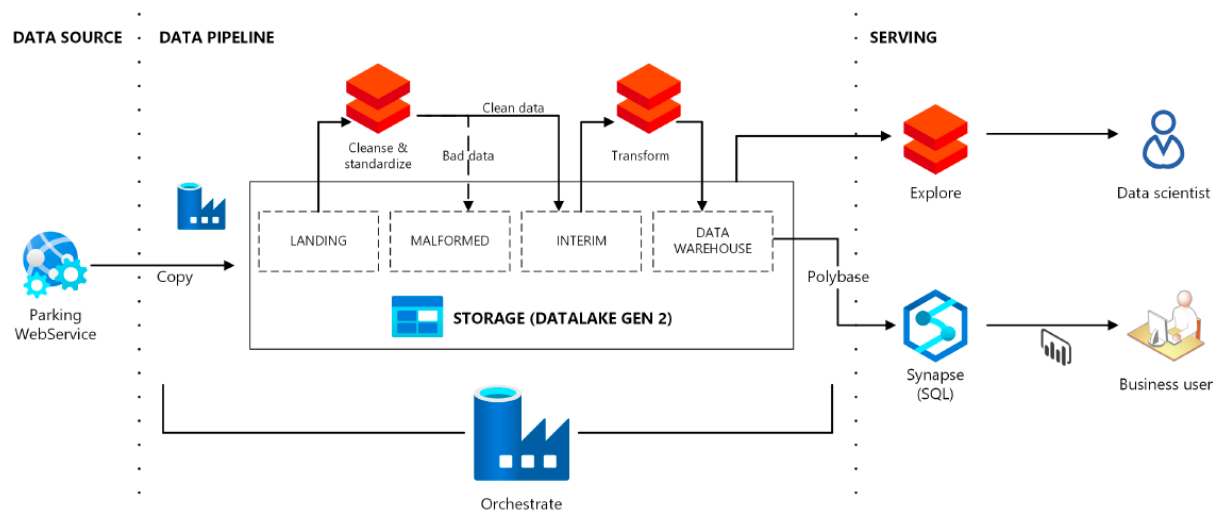
기본개념-팀자료

-17

A Unified Data Infrastructure Architecture



DataOps



Azure Databricks의 MLOps 워크플로

이 기사에서

1.

MLOps란 무엇인가?

2.

MLOps에 대한 일반 권장 사항

3.

추천 MLOps 워크플로

4.

개발 단계

5.

스테이징 단계

6.

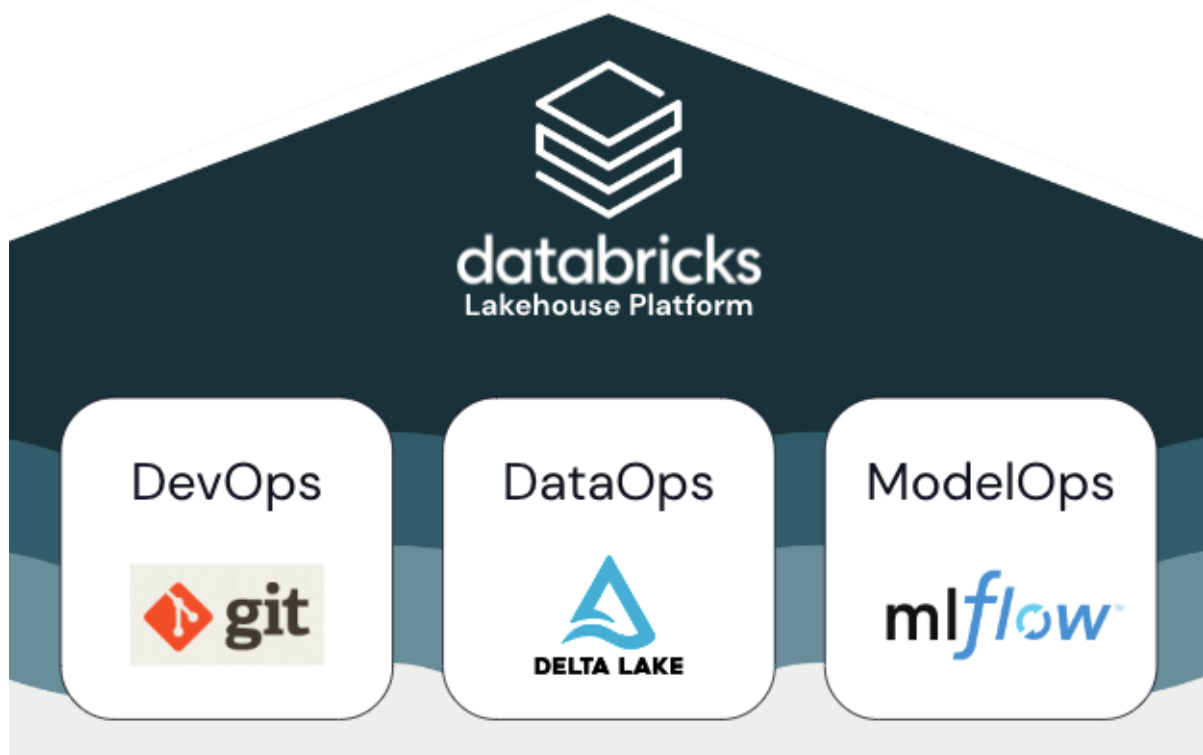
생산 단계 **Show less**

이 문서에서는 Databricks 플랫폼에서 MLOps를 사용하여 머신 러닝(ML) 시스템의 성능과 장기적 효율성을 최적화하는 방법을 설명합니다. 여기에는 MLOps 아키텍처에 대한 일반적인 권장 사항이 포함되어 있으며 ML 개발에서 프로덕션까지의 프로세스에 대한 모델로 사용할 수 있는 Databricks 플랫폼을 사용하는 일반화된 워크플로를 설명합니다. LLMOps 애플리케이션에 대한 이 워크플로 수정 사항은 LLMOps 워크플로를 참조하세요 .

자세한 내용은 The Big Book of MLOps를 참조하세요 .

MLOps란 무엇인가?

MLOps는 ML 시스템의 성능, 안정성 및 장기적 효율성을 개선하기 위해 코드, 데이터 및 모델을 관리하기 위한 프로세스 및 자동화된 단계의 집합입니다. DevOps, DataOps 및 ModelOps를 결합합니다.



코드, 데이터, 모델과 같은 ML 자산은 엄격한 액세스 제한이 없고 엄격하게 테스트되지 않은 초기 개발 단계부터 중간 테스트 단계를 거쳐 엄격하게 제어되는 최종 프로덕션 단계로 진행되는 단계로 개발됩니다. Databricks 플랫폼을 사용하면 통합 액세스 제어를 통해 단일 플랫폼에서 이러한 자산을 관리할 수 있습니다. 동일한 플랫폼에서 데이터 애플리케이션과 ML 애플리케이션을 개발하여 데이터를 이동하는 데 따른 위험과 지연을 줄일 수 있습니다.

MLOps에 대한 일반 권장 사항

이 섹션에는 Databricks의 MLOps에 대한 몇 가지 일반적인 권장 사항과 자세한 정보에 대한 링크가 포함되어 있습니다.

각 단계에 대해 별도의 환경을 만듭니다.

실행 환경은 모델과 데이터가 코드에 의해 생성되거나 소비되는 장소입니다. 각 실행 환경은 컴퓨팅 인스턴스, 런타임 및 라이브러리, 자동화된 작업으로 구성됩니다.

Databricks는 ML 코드와 모델 개발의 여러 단계에 대해 명확하게 정의된 단계 간 전환을 통해 별도의 환경을 만드는 것을 권장합니다. 이 문서에서 설명하는 워크플로는 단계에 대한 일반적인 이름을 사용하여 이 프로세스를 따릅니다.

- 개발
- 각색
- 생산

귀하의 조직의 특정 요구 사항을 충족하기 위해 다른 구성을 사용할 수도 있습니다.

액세스 제어 및 버전 관리

액세스 제어 및 버전 관리가 모든 소프트웨어 운영 프로세스의 핵심 구성 요소입니다. Databricks는 다음을 권장합니다.

- **버전 제어를 위해 Git을 사용하세요.** Databricks Git 폴더를 개발자 도구 및 지침을

파이프라인과 코드는 버전 제어를 위해 Git에 저장해야 합니다. ML 로직을 단계 간에 이동하는 것은 개발 브랜치에서 스테이징 브랜치로, 릴리스 브랜치로 코드를 이동하는 것으로 해석할 수 있습니다.

사용하여 Git 공급자와 통합하고 노트북과 소스 코드를 Databricks 작업 공간과 동기화하세요. Databricks는 또한 Git 통합 및 버전 제어를 위한 추가 도구를 제공합니다.

참조하세요 .

- **델타 테이블을 사용하여 레이크하우스 아키텍처에 데이터를 저장합니다.** 레이크하우스 아키텍처에 저장해야 합니다. 원시 데이터와 피쳐 테이블은 모두 누가 읽고 수정할 수 있는지 확인하기 위한 액세스 제어가 있는 델타 테이블

데이터는 클라우드 계정의

로 저장해야 합니다 .

- **MLflow로 모델 개발을 관리하세요.** MLflow를

사용하여 모델 개발 프로세스를 추적하고 코드 스냅샷, 모델 매개변수, 메트릭 및 기타 메타 데이터를 저장할 수 있습니다.

- **Unity Catalog의 모델을 사용하여 모델 라이프사이클을 관리합니다.** Unity Catalog의 모델을

사용하여 모델 버전 관리, 거버넌스 및 배포 상태를 관리합니다.

모델이 아닌 코드를 배포하세요

대부분의 상황에서 Databricks는 ML 개발 프로세스 중에 *모델이 아닌 코드*를 한 환경에서 다른 환경으로 승격시키는 것을 권장합니다. 이런 방식으로 프로젝트 자산을 이동하면 ML 개발 프로세스의 모든 코드가 동일한 코드 검토 및 통합 테스트 프로세스를 거치게 됩니다. 또한 모델의 프로덕션 버전이 프로덕션 코드에서 학습되도록 합니다. 옵션과 트레이드오프에 대한 자세한 내용은 모델 배포 패턴을 참조하세요 .

추천 MLOps 워크플로

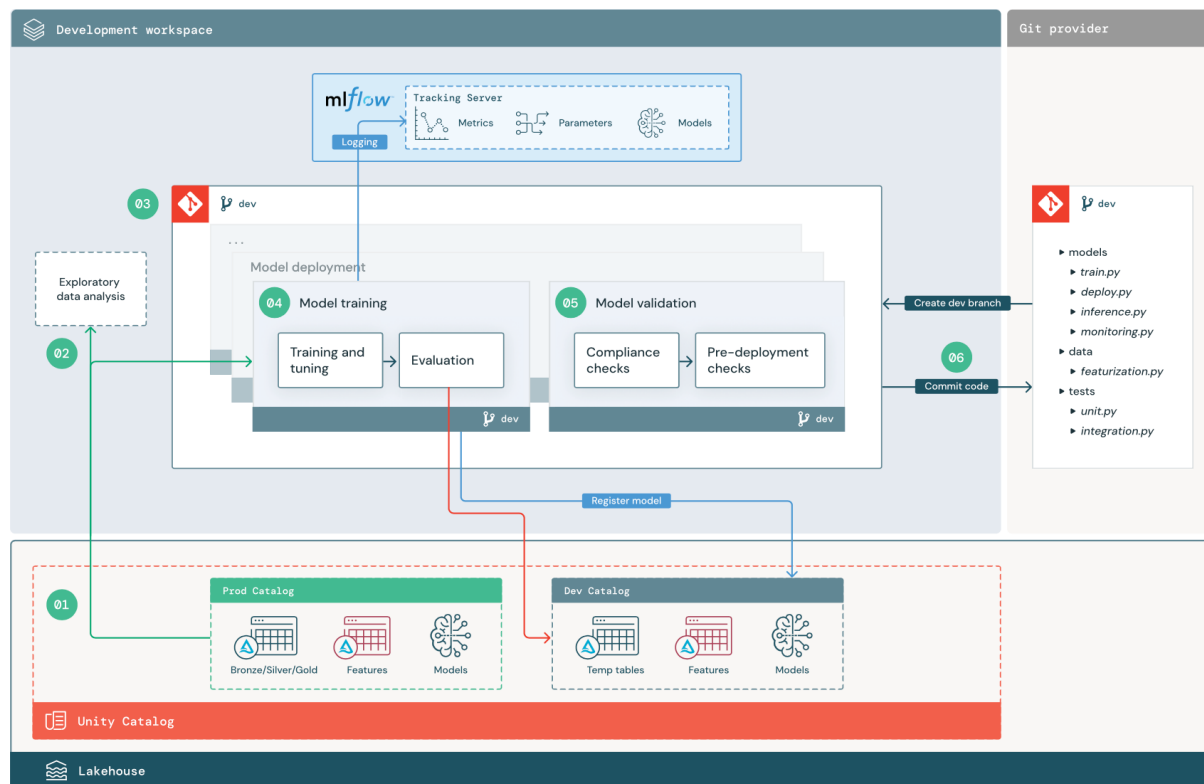
다음 섹션에서는 개발, 스테이징, 프로덕션의 세 단계를 다루는 일반적인 MLOps 워크플로를 설명합니다.

이 섹션에서는 "데이터 과학자"와 "ML 엔지니어"라는 용어를 전형적인 페르소나로 사용합니다. MLOps 워크플로의 구체적인 역할과 책임은 팀과 조직마다 다릅니다.

개발 단계

개발 단계의 초점은 실험입니다. 데이터 과학자는 기능과 모델을 개발하고 실험을 실행하여 모델 성능을 최적화합니다. 개발 프로세스의 출력은 기능 계산, 모델 학습, 추론 및 모니터링을 포

함할 수 있는 ML 파이프라인 코드입니다.



번호가 매겨진 단계는 다이어그램에 표시된 숫자에 해당합니다.

1. 데이터 소스

개발 환경은 Unity Catalog의 dev 카탈로그로 표현됩니다. 데이터 과학자는 개발 작업 공간에서 임시 데이터와 기능 테이블을 만들 때 dev 카탈로그에 대한 읽기-쓰기 액세스 권한을 갖습니다. 개발 단계에서 만든 모델은 dev 카탈로그에 등록됩니다.

이상적으로, 개발 작업 공간에서 작업하는 데이터 과학자는 prod 카탈로그의 프로덕션 데이터에 대한 읽기 전용 액세스 권한도 가지고 있습니다. 데이터 과학자에게 prod 카탈로그의 프로덕션 데이터, 추론 테이블 및 메트릭 테이블에 대한 읽기 액세스를 허용하면 현재 프로덕션 모델 예측 및 성능을 분석할 수 있습니다. 데이터 과학자는 실험 및 분석을 위해 프로덕션 모델을 로드할 수도 있어야 합니다.

프로덕션 카탈로그에 읽기 전용 액세스 권한을 부여할 수 없는 경우 프로덕션 데이터의 스냅샷을 개발 카탈로그에 작성하여 데이터 과학자가 프로젝트 코드를 개발하고 평가할 수 있습니다.

2. 탐색적 데이터 분석(EDA)

데이터 과학자는 노트북을 사용하여 대화형 반복 프로세스에서 데이터를 탐색하고 분석합니다. 목표는 사용 가능한 데이터가 비즈니스 문제를 해결할 잠재력이 있는지 평가하는 것입니다. 이 단계에서 데이터 과학자는 모델 교육을 위한 데이터 준비 및 기능화 단계를 식별하기 시작합니다. 이 임시 프로세스는 일반적으로 다른 실행 환경에 배포될 파이프라인의 일부가 아닙니다.

Databricks AutoML은 데이터세트에 대한 기준 모델을 생성하여 이 프로세스를 가속화합니다. AutoML은 일련의 시행을 수행하고 기록하며 각 시행에 대한 소스 코드가 있는 Python 노트북을 제공하므로 코드를 검토, 재생산 및 수정할 수 있습니다. AutoML은 또한 데이터세트에 대한 요약 통계를 계산하고 이 정보를 검토할 수 있는 노트북에 저장합니다.

3. 코드

코드 저장소에는 ML 프로젝트의 모든 파이프라인, 모듈 및 기타 프로젝트 파일이 들어 있습니다. 데이터 과학자는 프로젝트 저장소의 개발("dev") 브랜치에서 새 파이프라인이나 업데이트된 파이프라인을 만듭니다. EDA와 프로젝트의 초기 단계부터 데이터 과학자는 저장소에서 작업하여 코드를 공유하고 변경 사항을 추적해야 합니다.

4. 모델 학습(개발)

데이터 과학자는 개발 또는 프로덕션 카탈로그의 테이블을 사용하여 개발 환경에서 모델 학습 파이프라인을 개발합니다.

이 파이프라인에는 2가지 작업이 포함됩니다.

- **학습 및 튜닝.** 학습 프로세스는 모델 매개변수, 메트릭 및 아티팩트를 MLflow 추적 서버에 로깅합니다. 하이퍼파라미터를 학습하고 튜닝한 후 최종 모델 아티팩트는 추적 서버에 로깅되어 모델, 학습된 입력 데이터 및 모델을 생성하는 데 사용된 코드 간의 링크를 기록합니다.
- **평가.** 보류된 데이터에 대한 테스트를 통해 모델 품질을 평가합니다. 이러한 테스트의 결과는 MLflow Tracking 서버에 기록됩니다. 평가의 목적은 새로 개발된 모델이 현재 프로덕션 모델보다 성능이 더 좋은지 확인하는 것입니다. 충분한 권한이 주어지면 prod 카탈로그에 등록된 모든 프로덕션 모델을 개발 작업 공간에 로드하여 새로 훈련된 모델과 비교할 수 있습니다.

조직의 거버넌스 요구 사항에 모델에 대한 추가 정보가 포함된 경우 MLflow 추적을 사용하여 저장할 수 있습니다. 일반적인 아티팩트는 일반 텍스트 설명 및 SHAP에서 생성된 플롯과 같은 모델 해석입니다. 특정 거버넌스 요구 사항은 데이터 거버넌스 책임자 또는 비즈니스 이해 관계자에게서 나올 수 있습니다.

모델 학습 파이프라인의 출력은 개발 환경의 MLflow 추적 서버에 저장된 ML 모델 아티팩트입니다. 파이프라인이 스테이징 또는 프로덕션 작업 공간에서 실행되는 경우 모델 아티팩트는 해당 작업 공간의 MLflow 추적 서버에 저장됩니다.

모델 학습이 완료되면 Unity Catalog에 모델을 등록합니다. 파이프라인 코드를 설정하여 모델 파이프라인이 실행된 환경에 해당하는 카탈로그에 모델을 등록합니다. 이 예에서는 dev 카탈로그입니다.

권장 아키텍처를 사용하면 첫 번째 작업이 모델 학습 파이프라인이고 그 다음에 모델 검증 및 모델 배포 작업이 이어지는 멀티태스킹 Databricks 워크플로를 배포합니다. 모델 학습 작업은 모델 검증 작업에서 사용할 수 있는 모델 URI를 생성합니다. 작업 값을 사용하여 이 URI를 모델에 전달할 수 있습니다.

5. 모델 검증 및 배포(개발)

모델 학습 파이프라인 외에도 모델 검증, 모델 배포 파이프라인 등 다른 파이프라인도 개발 환경에 포함되어 있습니다.

- **모델 검증.** 모델 검증 파이프라인은 모델 학습 파이프라인에서 모델 URI를 가져와 Unity Catalog에서 모델을 로드하고 검증 검사를 실행합니다.

검증 검사는 맥락에 따라 달라집니다. 여기에는 형식 및 필수 메타데이터 확인과 같은 기본적인 검사와 사전 정의된 규정 준수 검사 및 선택된 데이터 슬라이스에서 모델 성능 확인과 같이 엄격하게 규제되는 산업에 필요할 수 있는 보다 복잡한 검사가 포함될 수 있습니다.

모델 검증 파이프라인의 주요 기능은 모델을 배포 단계로 진행해야 하는지 여부를 결정하는 것입니다. 모델이 배포 전 검사를 통과하면 Unity Catalog에서 "Challenger" 별칭을 할당할 수 있습니다. 검사에 실패하면 프로세스가 종료됩니다. 워크플로를 구성하여 사용자에게 검증 실패를 알릴 수 있습니다. 작업 이벤트에 대한 이메일 및 시스템 알림 추가를 참조하세요.

- **모델 배포.** 모델 배포 파이프라인은 일반적으로 별칭 업데이트를 사용하여 새로 훈련된 "Challenger" 모델을 "Champion" 상태로 직접 승격시키거나 기존 "Champion" 모델과 새 "Challenger" 모델 간의 비교를 용이하게 합니다. 이 파이프라인은 Model Serving 엔드포인트와 같은 필요한 추론 인프라를 설정할 수도 있습니다. 모델 배포 파이프라인에 포함된 단계에 대한 자세한 내용은 Production을 참조하세요.

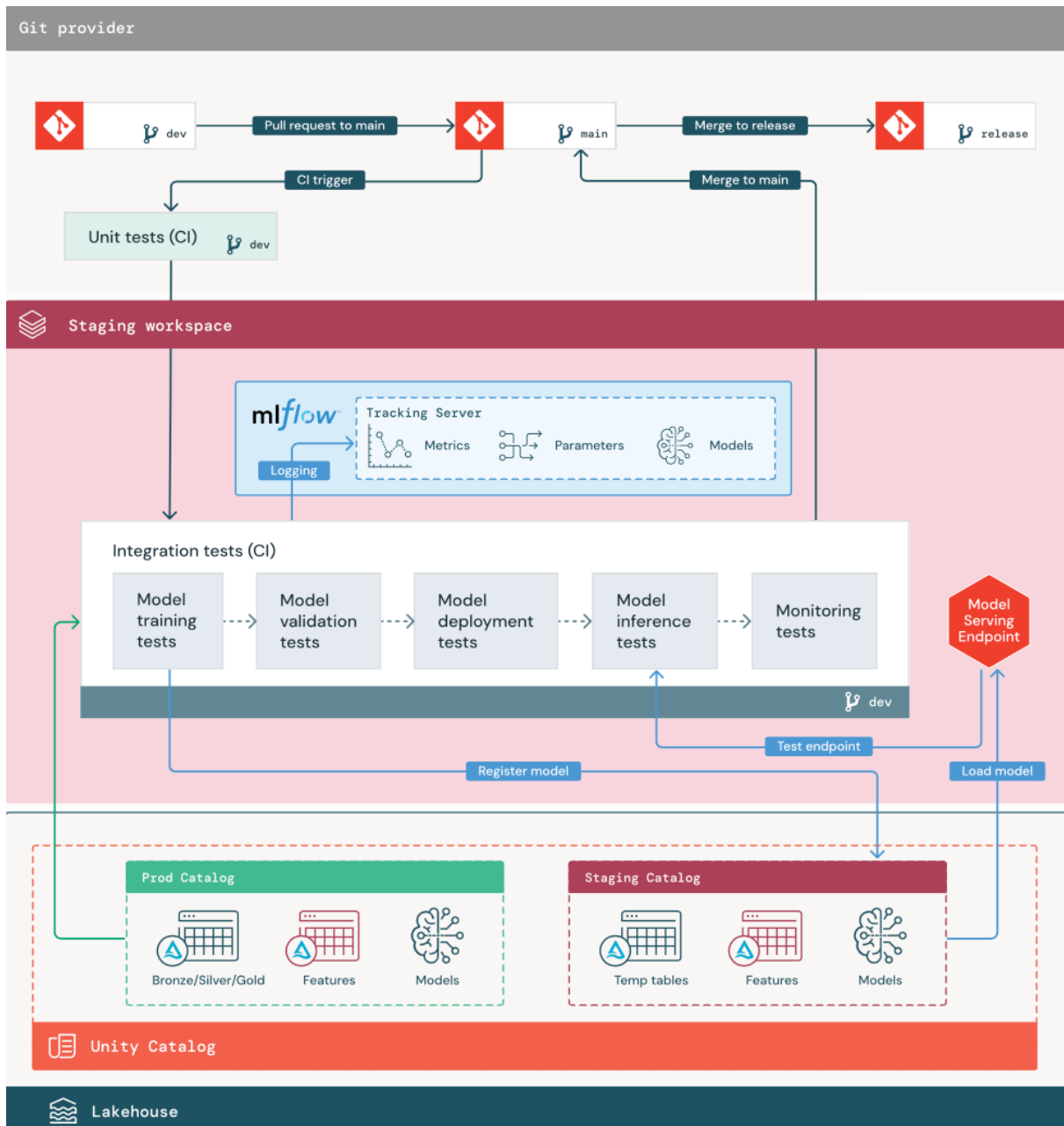
6. 코드 커밋

데이터 과학자나 ML 엔지니어는 훈련, 검증, 배포 및 기타 파이프라인을 위한 코드를 개발한 후 개발 브랜치 변경 사항을 소스 제어에 커밋합니다.

스테이징 단계

이 단계의 초점은 ML 파이프라인 코드를 테스트하여 프로덕션에 사용할 준비가 되었는지 확인하는 것입니다. 이 단계에서는 모델 학습을 위한 코드, 기능 엔지니어링 파이프라인, 추론 코드 등을 포함하여 모든 ML 파이프라인 코드를 테스트합니다.

ML 엔지니어는 이 단계에서 실행되는 단위 및 통합 테스트를 구현하기 위해 CI 파이프라인을 만듭니다. 스테이징 프로세스의 출력은 CI/CD 시스템이 프로덕션 단계를 시작하도록 트리거하는 릴리스 브랜치입니다.



1. 데이터

스테이징 환경은 ML 파이프라인을 테스트하고 Unity Catalog에 모델을 등록하기 위해 Unity Catalog에 자체 카탈로그를 가져야 합니다. 이 카탈로그는 다이어그램에서 "스테이징" 카탈로그로 표시됩니다. 이 카탈로그에 작성된 자산은 일반적으로 일시적이며 테스트가 완료될 때까지만 보관됩니다. 개발 환경은 디버깅 목적으로 스테이징 카탈로그에 대한 액세스가 필요할 수도 있습니다.

2. 코드 병합

데이터 과학자는 개발 또는 프로덕션 카탈로그의 테이블을 사용하여 개발 환경에서 모델 학습 파이프라인을 개발합니다.

- **풀 리퀘스트.** 배포 프로세스는 소스 컨트롤에서 프로젝트의 메인 브랜치에 대한 풀 리퀘스트가 생성될 때 시작됩니다.
- **단위 테스트(CI).** 풀 리퀘스트는 자동으로 소스 코드를 빌드하고 단위 테스트를 트리거합니다. 단위 테스트가 실패하면 풀 리퀘스트가 거부됩니다.

단위 테스트는 소프트웨어 개발 프로세스의 일부이며 모든 코드 개발 중에 지속적으로 실행되고 코드베이스에 추가됩니다. CI 파이프라인의 일부로 단위 테스트를 실행하면 개발 브랜치에서 변경한 내용이 기존 기능을 손상시키지 않도록 할 수 있습니다.

3. 통합 테스트(CI)

그런 다음 CI 프로세스는 통합 테스트를 실행합니다. 통합 테스트는 모든 파이프라인(기능 엔지니어링, 모델 학습, 추론 및 모니터링 포함)을 실행하여 함께 올바르게 작동하는지 확인합니다. 스테이징 환경은 합리적으로 가능한 한 프로덕션 환경과 일치해야 합니다.

실시간 추론을 사용하여 ML 애플리케이션을 배포하는 경우 스테이징 환경에서 제공 인프라를 만들고 테스트해야 합니다. 여기에는 스테이징 환경에서 제공 엔드포인트를 만들고 모델을 로드하는 모델 배포 파이프라인을 트리거하는 것이 포함됩니다.

통합 테스트를 실행하는 데 필요한 시간을 줄이기 위해 일부 단계는 테스트의 충실성과 속도 또는 비용 사이에서 균형을 이룰 수 있습니다. 예를 들어, 모델을 학습하는 데 비용이 많이 들거나 시간이 많이 걸리는 경우 작은 데이터 하위 집합을 사용하거나 학습 반복 횟수를 줄일 수 있습니다. 모델 제공의 경우 프로덕션 요구 사항에 따라 통합 테스트에서 전체 규모의 부하 테스트를 수행하거나 작은 일괄 작업 또는 임시 엔드포인트에 대한 요청만 테스트할 수 있습니다.

4. 스테이징 브랜치에 병합

모든 테스트가 통과하면 새 코드가 프로젝트의 메인 브랜치에 병합됩니다. 테스트가 실패하면 CI/CD 시스템은 사용자에게 알리고 풀 리퀘스트에 결과를 게시해야 합니다.

메인 브랜치에서 주기적 통합 테스트를 예약할 수 있습니다. 브랜치가 여러 사용자의 동시 풀 요청으로 자주 업데이트되는 경우 좋은 생각입니다.

5. 릴리스 브랜치 생성

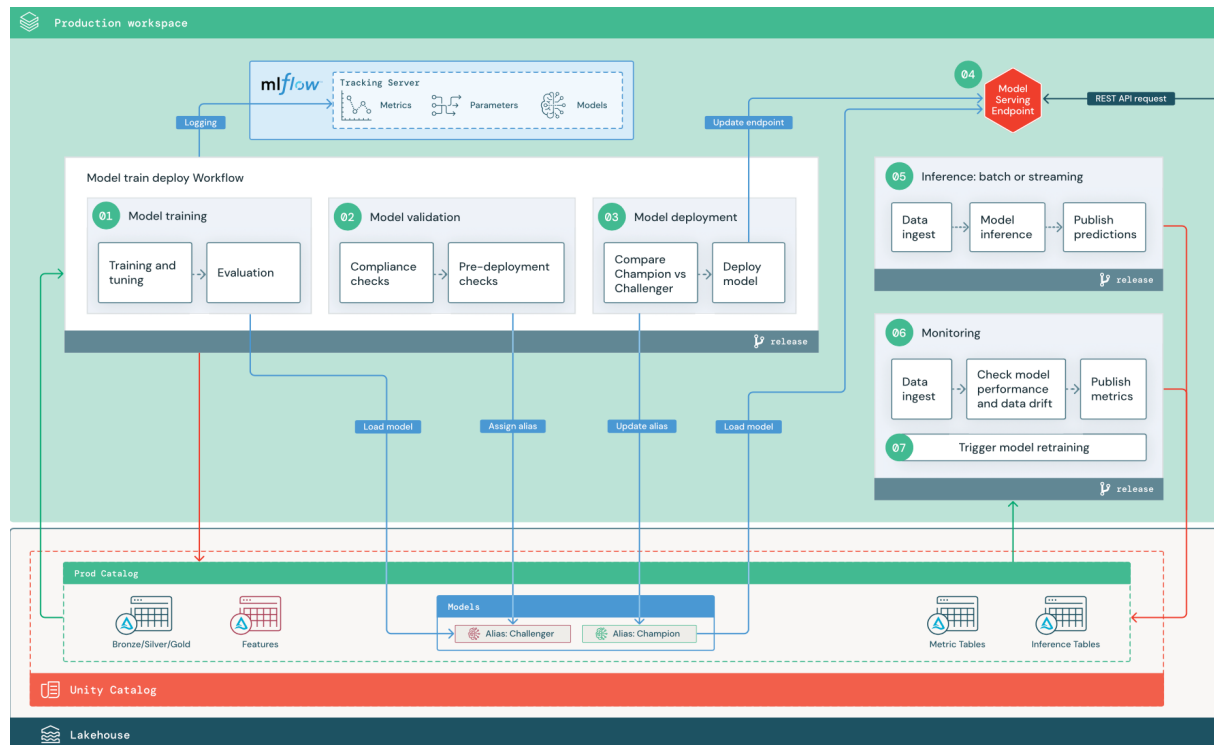
CI 테스트를 통과하고 개발 브랜치가 메인 브랜치에 병합되면 ML 엔지니어가 릴리스 브랜치를 생성하고, 이를 통해 CI/CD 시스템이 프로덕션 작업을 업데이트합니다.

생산 단계

ML 엔지니어는 ML 파이프라인이 배포되고 실행되는 프로덕션 환경을 소유합니다. 이러한 파이프라인은 모델 학습을 트리거하고, 새로운 모델 버전을 검증하고 배포하고, 다운스트림 테이블이나 애플리케이션에 예측을 게시하고, 전체 프로세스를 모니터링하여 성능 저하와 불안정성을 방지합니다.

데이터 과학자는 일반적으로 프로덕션 환경에서 쓰기 또는 계산 액세스 권한이 없습니다. 그러나 테스트 결과, 로그, 모델 아티팩트, 프로덕션 파이프라인 상태 및 모니터링 테이블에 대한 가시성이 있는 것이 중요합니다. 이러한 가시성을 통해 프로덕션에서 문제를 식별하고 진단하고

새 모델의 성능을 현재 프로덕션 중인 모델과 비교할 수 있습니다. 이러한 목적을 위해 프로덕션 카탈로그의 자산에 대한 읽기 전용 액세스 권한을 데이터 과학자에게 부여할 수 있습니다.



번호가 매겨진 단계는 다이어그램에 표시된 숫자에 해당합니다.

1. 모델 학습

이 파이프라인은 코드 변경이나 자동화된 재교육 작업에 의해 트리거될 수 있습니다. 이 단계에서는 프로덕션 카탈로그의 테이블이 다음 단계에 사용됩니다.

- **훈련 및 튜닝.** 훈련 프로세스 동안 로그는 프로덕션 환경 MLflow Tracking 서버에 기록됩니다. 이러한 로그에는 모델 메트릭, 매개변수, 태그 및 모델 자체가 포함됩니다. 피쳐 테이블을 사용하는 경우 모델은 Databricks Feature Store 클라이언트를 사용하여 MLflow에 기록되며, 이 클라이언트는 추론 시간에 사용되는 피쳐 조회 정보와 함께 모델을 패키징합니다.

개발 중에 데이터 과학자는 많은 알고리즘과 하이퍼파라미터를 테스트할 수 있습니다. 프로덕션 트레이닝 코드에서는 성능이 가장 좋은 옵션만 고려하는 것이 일반적입니다. 이런 식으로 튜닝을 제한하면 시간을 절약하고 자동 재교육에서 튜닝으로 인한 분산을 줄일 수 있습니다.

데이터 과학자가 프로덕션 카탈로그에 대한 읽기 전용 액세스 권한이 있는 경우 모델에 대한 최적의 하이퍼파라미터 세트를 결정할 수 있습니다. 이 경우 프로덕션에 배포된 모델 학습 파이프라인은 일반적으로 파이프라인에 구성 파일로 포함된 선택된 하이퍼파라미터 세트를 사용하여 실행할 수 있습니다.

- **평가.** 모델 품질은 보류된 프로덕션 데이터에 대한 테스트를 통해 평가됩니다. 이러한 테스트의 결과는 MLflow 추적 서버에 기록됩니다. 이 단계에서는 개발 단계에서 데이터 과학자가 지정한 평가 지표를 사용합니다. 이러한 지표에는 사용자 지정 코드가 포함될 수 있습니다.
- **모델 등록.** 모델 학습이 완료되면 모델 아티팩트는 Unity Catalog의 프로덕션 카탈로그에 지정된 모델 경로에 등록된 모델 버전으로 저장됩니다. 모델 학습 작업은 모델 검증 작업에서 사용할 수 있는 모델 URI를 생성합니다. 작업 값을 사용 하여 이 URI를 모델에 전달할 수 있습니다.

2. 모델 검증

이 파이프라인은 1단계의 모델 URI를 사용하고 Unity Catalog에서 모델을 로드합니다. 그런 다음 일련의 유효성 검사를 실행합니다. 이러한 검사는 조직과 사용 사례에 따라 달라지며 기본 형식 및 메타데이터 유효성 검사, 선택한 데이터 슬라이스에 대한 성능 평가, 태그 또는 문서에 대한 규정 준수 검사와 같은 조직 요구 사항 준수와 같은 사항을 포함할 수 있습니다.

모델이 모든 검증 검사를 성공적으로 통과하면 Unity Catalog에서 모델 버전에 "Challenger" 별칭을 할당할 수 있습니다. 모델이 모든 검증 검사를 통과하지 못하면 프로세스가 종료되고 사용자에게 자동으로 알림을 보낼 수 있습니다. 태그를 사용하여 이러한 검증 검사의 결과에 따라 키-값 속성을 추가할 수 있습니다. 예를 들어, "model_validation_status" 태그를 만들고 테스트가 실행될 때 값을 "PENDING"으로 설정한 다음 파이프라인이 완료되면 "PASSED" 또는 "FAILED"로 업데이트할 수 있습니다.

모델이 Unity Catalog에 등록되어 있으므로 개발 환경에서 작업하는 데이터 과학자는 프로덕션 카탈로그에서 이 모델 버전을 로드하여 모델이 검증에 실패했는지 조사할 수 있습니다. 결과에 관계없이 결과는 모델 버전에 대한 주석을 사용하여 프로덕션 카탈로그의 등록된 모델에 기록됩니다.

3. 모델 배포

검증 파이프라인과 마찬가지로 모델 배포 파이프라인은 조직과 사용 사례에 따라 달라집니다. 이 섹션에서는 새로 검증된 모델에 "Challenger" 별칭을 지정했고 기존 프로덕션 모델에 "Champion" 별칭을 지정했다고 가정합니다. 새 모델을 배포하기 전 첫 번째 단계는 최소한 현재 프로덕션 모델만큼 잘 수행되는지 확인하는 것입니다.

- **"CHALLENGER"와 "CHAMPION" 모델을 비교합니다.** 이 비교는 오프라인 또는 온라인으로 수행할 수 있습니다. 오프라인 비교는 보류된 데이터 세트에 대해 두 모델을 평가하고 MLflow Tracking 서버를 사용하여 결과를 추적합니다. 실시간 모델 제공의 경우 A/B 테스트 또는 새 모델의 점진적 롤아웃과 같이 더 오래 실행되는 온라인 비교를 수행할 수 있습니다. "Challenger" 모델 버전이 비교에서 더 나은 성과를 거두면 현재 "Champion" 별칭을 대체합니다.

Mosaic AI Model Serving과 Databricks Lakehouse Monitoring을 사용하면 엔드포인트에 대한 요청 및 응답 데이터가 포함된 추론 테이블을 자동으로 수집하고 모니터링할 수 있습니다.

기존의 "챔피언" 모델이 없다면 "챌린저" 모델을 비즈니스 휴리스틱이나 다른 임계값과 기준으로 비교할 수 있습니다.

여기에 설명된 프로세스는 완전히 자동화되어 있습니다. 수동 승인 단계가 필요한 경우 모델 배포 파이프라인에서 워크플로 알림 또는 CI/CD 콜백을 사용하여 이를 설정할 수 있습니다.

- **모델 배포.** 배치 또는 스트리밍 추론 파이프라인은 "Champion" 별칭으로 모델을 사용하도록 설정할 수 있습니다. 실시간 사용 사례의 경우 모델을 REST API 엔드포인트로 배포하기 위한 인프라를 설정해야 합니다. Mosaic AI Model Serving을 사용하여 이 엔드포인트를 만들고 관리할 수 있습니다. 엔드포인트가 현재 모델에 이미 사용 중인 경우 새 모델로 엔드포인트를 업데이트할 수 있습니다. Mosaic AI Model Serving은 새 구성이 준비될 때까지 기존 구성을 실행하여 다운타임 없는 업데이트를 실행합니다.

4. 모델 제공

모델 제공 엔드포인트를 구성할 때 Unity Catalog의 모델 이름과 제공할 버전을 지정합니다. 모델 버전이 Unity Catalog의 테이블에서 피처를 사용하여 학습된 경우 모델은 피처 및 함수에 대한 종속성을 저장합니다. 모델 제공은 추론 시간에 이 종속성 그래프를 자동으로 사용하여 적절한 온라인 스토어에서 피처를 조회합니다. 이 접근 방식은 데이터 전처리를 위한 함수를 적용하거나 모델 스코어링 중에 주문형 피처를 계산하는 데에도 사용할 수 있습니다.

여러 모델을 사용하여 단일 엔드포인트를 만들고 해당 모델 간의 엔드포인트 트래픽 분할을 지정하여 온라인에서 "Champion" 대 "Challenger" 비교를 수행할 수 있습니다.

5. 추론: 배치 또는 스트리밍

추론 파이프라인은 프로덕션 카탈로그에서 최신 데이터를 읽고, 주문형 기능을 계산하는 함수를 실행하고, "Champion" 모델을 로드하고, 데이터를 평가하고, 예측을 반환합니다. 일괄 또는 스트리밍 추론은 일반적으로 더 높은 처리량, 더 높은 대기 시간 사용 사례에 가장 비용 효율적인 옵션입니다. 저지연 예측이 필요하지만 예측을 오프라인에서 계산할 수 있는 시나리오의 경우 이러한 일괄 예측은 DynamoDB 또는 Cosmos DB와 같은 온라인 키-값 저장소에 게시될 수 있습니다.

Unity Catalog에 등록된 모델은 별칭으로 참조됩니다. 추론 파이프라인은 "Champion" 모델 버전을 로드하고 적용하도록 구성되어 있습니다. "Champion" 버전이 새 모델 버전으로 업데이트되면 추론 파이프라인은 다음 실행에 자동으로 새 버전을 사용합니다. 이런 방식으로 모델 배포 단계가 추론 파이프라인에서 분리됩니다.

배치 작업은 일반적으로 프로덕션 카탈로그의 테이블, 플랫폼 파일 또는 JDBC 연결을 통해 예측을 게시합니다. 스트리밍 작업은 일반적으로 Unity Catalog 테이블 또는 Apache Kafka와 같은 메시지 큐에 예측을 게시합니다.

6. 레이크하우스 모니터링

Lakehouse Monitoring은 입력 데이터와 모델 예측의 데이터 드리프트 및 모델 성능과 같은 통계적 속성을 모니터링합니다. 이러한 지표를 기반으로 알림을 만들거나 대시보드에 게시할 수

있습니다.

- **데이터 수집.**

이 파이프라인은 일괄, 스트리밍 또는 온라인 추론에서 로그를 읽습니다.

- **정확도와 데이터 드리프트를 확인합니다.**

파이프라인은 입력 데이터, 모델의 예측 및 인프라 성능에 대한 메트릭을 계산합니다. 데이터 과학자는 개발 중에 데이터 및 모델 메트릭을 지정하고 ML 엔지니어는 인프라 메트릭을 지정합니다. Lakehouse Monitoring으로 사용자 지정 메트릭을 정의할 수도 있습니다.

- **메트릭을 게시하고 알림을 설정합니다.**

파이프라인은 분석 및 보고를 위해 프로덕션 카탈로그의 테이블에 씁니다. 이러한 테이블은 개발 환경에서 읽을 수 있도록 구성해야 데이터 과학자가 분석에 액세스할 수 있습니다. Databricks SQL을 사용하여 모니터링 대시보드를 만들어 모델 성능을 추적하고 모니터링 작업 또는 대시보드 도구를 설정하여 메트릭이 지정된 임계값을 초과할 때 알림을 발행할 수 있습니다.

- **모델 재교육 트리거.**

모니터링 메트릭이 성능 문제나 입력 데이터의 변경을 나타낼 때, 데이터 과학자는 새로운 모델 버전을 개발해야 할 수 있습니다. 이런 일이 발생하면 데이터 과학자에게 알리기 위해 SQL 알림을 설정할 수 있습니다.

7. 재교육

이 아키텍처는 위의 동일한 모델 학습 파이프라인을 사용하여 자동 재학습을 지원합니다.

Databricks는 예약된 주기적 재학습으로 시작하여 필요할 때 트리거된 재학습으로 전환할 것을 권장합니다.

- **예약됨.** 예약된 작업을 생성할 수 있습니다.

정기적으로 새 데이터가 제공되는 경우, 최신 사용 가능한 데이터에서 모델 학습 코드를 실행하기 위한

- **트리거됨.**

모니터링 파이프라인이 모델 성능 문제를 식별하고 알림을 보낼 수 있는 경우 재교육을 트리거할 수도 있습니다. 예를 들어, 들어오는 데이터의 분포가 크게 변경되거나 모델 성능이 저하되는 경우 자동 재교육 및 재배포를 통해 최소한의 인적 개입으로 모델 성능을 높일 수 있습니다. 이는 메트릭이 비정상적인지 확인하는 SQL 알림을 통해 달성할 수 있습니다(예: 임계값에 대해 드리프트 또는 모델 품질 확인). 알림은 웹훅 대상을 사용하도록 구성할 수 있으며, 이는 이후에 교육 워크플로를 트리거할 수 있습니다.

재교육 파이프라인이나 다른 파이프라인에 성능 문제가 나타나면 데이터 과학자는 문제를 해결하기 위해 추가 실험을 위해 개발 환경으로 돌아가야 할 수도 있습니다.

1. Datalake 신규 구축시 발생할 수 있는 이슈와 hedge 방안 제시
 - 기존 정보계의 역할 대체 방안 및 이슈 해결 작성
1. 온/오프라인 연계 실시간성 CDP(Customer Data Platform) 구축시 아키텍처 제시 및 이슈 해결방안 도출
 - 고객행동정보 실시간 통합을 위한 아키텍처 구성 방안 및 데이터 보정 방안

만든 문제]

○○○는 2022년과 2023년 동안 A기업과 B기업을 위한 대규모 차세대 프로젝트를 수행했습니다. 이 프로젝트는 Datalake와 실시간 CDP(Customer Data Platform)를 구축하여, 기업의 데이터 활용 능력을 극대화하고 고객 맞춤형 서비스를 제공하는 것을 목표로 하고 있습니다. 데이터 수집부터 처리, 그리고 사용자에게 제공되는 레포트 생성까지의 전체 데이터 흐름을 고려한 구체적인 방안을 제시해야 합니다.

문제 상황 1: Datalake 신규 구축 및 데이터 수집부터 레포트 제공까지의 프로세스 설계

A기업은 기존의 정보계 시스템을 대체하고, 비정형 데이터를 포함한 대규모 데이터를 처리할 수 있는 새로운 Datalake를 구축하고자 합니다. 이 Datalake는 데이터 수집, 처리, 분석, 그리고 최종 사용자에게 제공되는 레포트 생성까지 모든 과정을 포함해야 합니다. 이 과정에서 여러 가지 기술적, 전략적 문제가 발생할 수 있습니다.

1. Datalake를 통한 기존 정보계 시스템 대체 방안:

- A기업의 기존 정보계 시스템은 주로 구조화된 데이터를 처리해왔습니다. 이제 Datalake를 통해 비정형 데이터, 스트리밍 데이터 등 다양한 데이터 유형을 통합하려고 합니다. 기존 정보계 시스템의 역할을 어떻게 Datalake로 대체할 수 있을지 방안을 제시하십시오.
- Datalake로 전환 시 데이터 중복, 일관성 문제, 성능 저하 등을 해결하기 위한 구체적인 방안을 설명하십시오.

2. 데이터 수집부터 처리, 레포트 생성까지의 통합 프로세스 설계:

- Datalake에서 다양한 데이터 소스로부터 데이터를 수집하고, 이를 처리하여 분석할 수 있는 전체 프로세스를 설계하십시오. 데이터의 수집, 저장, 처리, 분석, 레포트 생성까지의 흐름을 상세히 설명하십시오.

- 데이터 수집 과정에서 발생할 수 있는 품질 저하, 데이터 손실 등을 방지하기 위한 방법과, 최종 사용자에게 제공되는 레포트의 정확성을 보장하기 위한 데이터 검증 및 보정 방안을 제시하십시오.

3. Hedge 방안 및 이슈 해결:

- Datalake 구축 과정에서 발생할 수 있는 대표적인 이슈(예: 데이터 품질 저하, 보안 문제, 비용 증가 등)를 예측하고, 이를 사전에 예방하거나 해결할 수 있는 hedge 방안을 제시하십시오.
- 데이터 수집 및 처리 과정에서 발생할 수 있는 지연, 오류 등의 문제를 해결하기 위한 구체적인 전략을 설명하십시오.

문제 상황 2: 온/오프라인 연계 실시간성 CDP(Customer Data Platform) 구축 및 데이터 통합 프로세스

B기업은 고객 행동 정보를 실시간으로 수집, 처리하여 통합된 데이터를 기반으로 고객 맞춤형 서비스를 제공하기 위해 CDP(Customer Data Platform)를 구축하려고 합니다. 이 플랫폼은 온/오프라인 데이터를 실시간으로 연계하여, 수집된 데이터를 처리하고 최종적으로 사용자에게 제공되는 인사이트와 레포트를 포함하는 전체 프로세스를 설계해야 합니다.

1. 실시간 CDP 아키텍처 구성 및 데이터 통합 프로세스 설계:

- 온/오프라인에서 발생하는 고객 행동 정보를 실시간으로 수집, 처리, 통합할 수 있는 CDP 아키텍처를 제시하십시오. 이 아키텍처는 데이터의 수집부터 처리, 분석, 그리고 사용자에게 제공되는 인사이트와 레포트 생성까지의 전체 흐름을 포함해야 합니다.
- 실시간 데이터 통합 과정에서 발생할 수 있는 데이터 지연, 일관성 부족 등의 문제를 해결하기 위한 구체적인 기술적 방안을 설명하십시오.

2. 데이터 보정 및 레포트 제공 과정에서의 이슈 해결 방안:

- 다양한 소스로부터 수집된 고객 행동 데이터를 보정하여 일관성을 유지하는 방안을 제시하십시오. 데이터 보정 과정에서 발생할 수 있는 문제(예: 데이터 형식 차이, 중복 데이터 처리, 누락된 데이터 복구 등)를 해결하기 위한 구체적인 방안을 설명하십시오.
- CDP에서 생성된 데이터를 기반으로 최종 사용자에게 제공되는 레포트의 신뢰성과 정확성을 높이기 위한 검증, 모니터링, 보정 방안을 제시하십시오.

3. 데이터 보안 및 개인정보 보호 이슈 해결:

- CDP 구축 시 발생할 수 있는 데이터 보안 문제와 개인정보 보호 이슈를 해결하기 위한 방안을 제시하십시오. 특히, 실시간으로 수집되고 처리되는 고객 데이터의 보호를 위해 적용할 수 있는 기술적, 관리적 대책을 구체적으로 설명하십시오.

추가 지시사항:

- 제시된 방안이 A기업과 B기업의 현재 및 미래의 비즈니스 요구를 어떻게 충족할 수 있는지를 논리적으로 설명하십시오.
- 문제 해결을 위한 전제 조건이나 가정을 설정하고, 해당 전제 조건 또는 가정이 문제 해결에 어떻게 영향을 미치는지를 명확히 기술하십시오.
- 아키텍처 구성 방안 및 다이어그램을 텍스트로 설명하거나, 시각적 다이어그램으로 포함해도 좋습니다.

[GPT 4.o 답변 : 문제에 대한 답변]

만든 문제에 대한 상세 답변 요청

문제 상황 1: Datalake 신규 구축 및 데이터 수집부터 레포트 제공까지의 프로세스 설계

A기업은 기존의 정보계 시스템을 대체하기 위해 새로운 Datalake를 구축하고자 합니다. 이 과정에서 데이터 수집, 처리, 분석, 그리고 최종 사용자에게 제공되는 레포트 생성까지의 모든 과정을 포함한 통합 프로세스를 설계해야 합니다.

1. Datalake를 통한 기존 정보계 시스템 대체 방안

A기업의 기존 정보계 시스템은 주로 구조화된 데이터를 처리해왔습니다. 이제 Datalake를 도입함으로써 비정형 데이터, 스트리밍 데이터 등 다양한 데이터 유형을 수용하고, 실시간 데이터를 처리할 수 있는 유연한 환경을 구축하려고 합니다.

- 데이터 통합 및 분류:
 - 구조화된 데이터: 기존 정보계에서 처리하던 구조화된 데이터(예: 관계형 데이터베이스에 저장된 트랜잭션 데이터)를 Datalake로 마이그레이션합니다. 이를 위해 ETL(Extract, Transform, Load) 도구를 사용하여 데이터를 추출하고, 변환한 후 Datalake에 저장합니다.

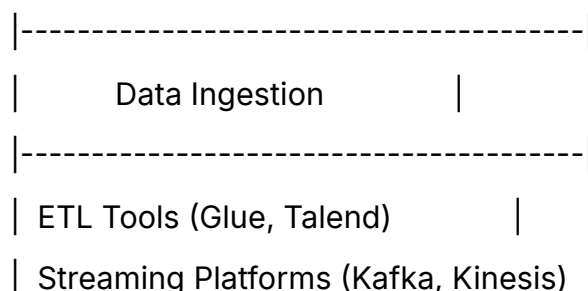
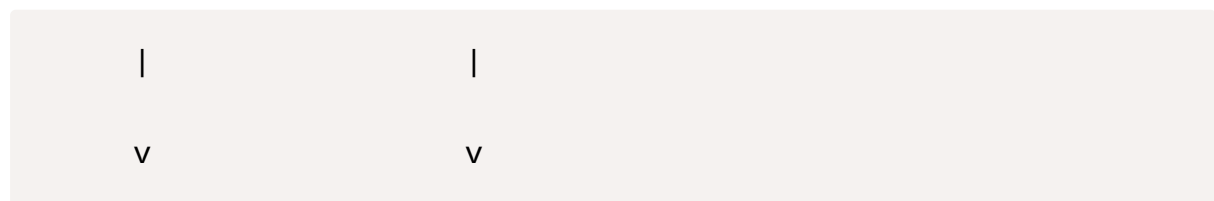
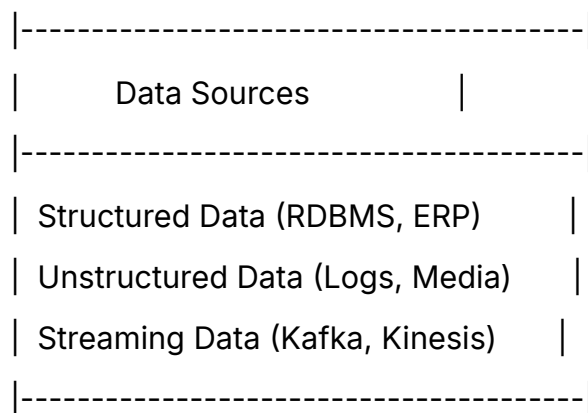
- 비정형 데이터: Datalake는 비정형 데이터(예: 로그 파일, 이미지, 텍스트 데이터)를 대규모로 저장할 수 있습니다. 비정형 데이터를 수집하여 Datalake의 적절한 저장소에 저장하고, 필요 시 데이터를 구조화하여 분석에 활용할 수 있습니다.
- 실시간 데이터: 스트리밍 데이터를 실시간으로 수집하고 처리하기 위해 Kafka 또는 Kinesis와 같은 스트리밍 플랫폼을 도입하여 Datalake와 통합합니다. 실시간 데이터는 이후 분석 및 레포트 생성에 사용될 수 있습니다.
- 데이터 중복 및 일관성 문제 해결:
 - 데이터 중복 제거: 데이터 중복을 방지하기 위해 데이터 디듀PLICATION(de-duplication) 알고리즘을 적용합니다. Datalake에 데이터를 저장하기 전에 중복 데이터를 식별하고 제거하는 프로세스를 포함시켜 일관된 데이터 저장을 보장합니다.
 - 데이터 일관성 관리: 데이터 일관성을 유지하기 위해 ACID(Atomicity, Consistency, Isolation, Durability) 특성을 준수하는 분산 트랜잭션 관리 시스템을 도입합니다. 이를 통해 다중 소스에서 동일한 데이터를 처리할 때 발생할 수 있는 일관성 문제를 해결할 수 있습니다.

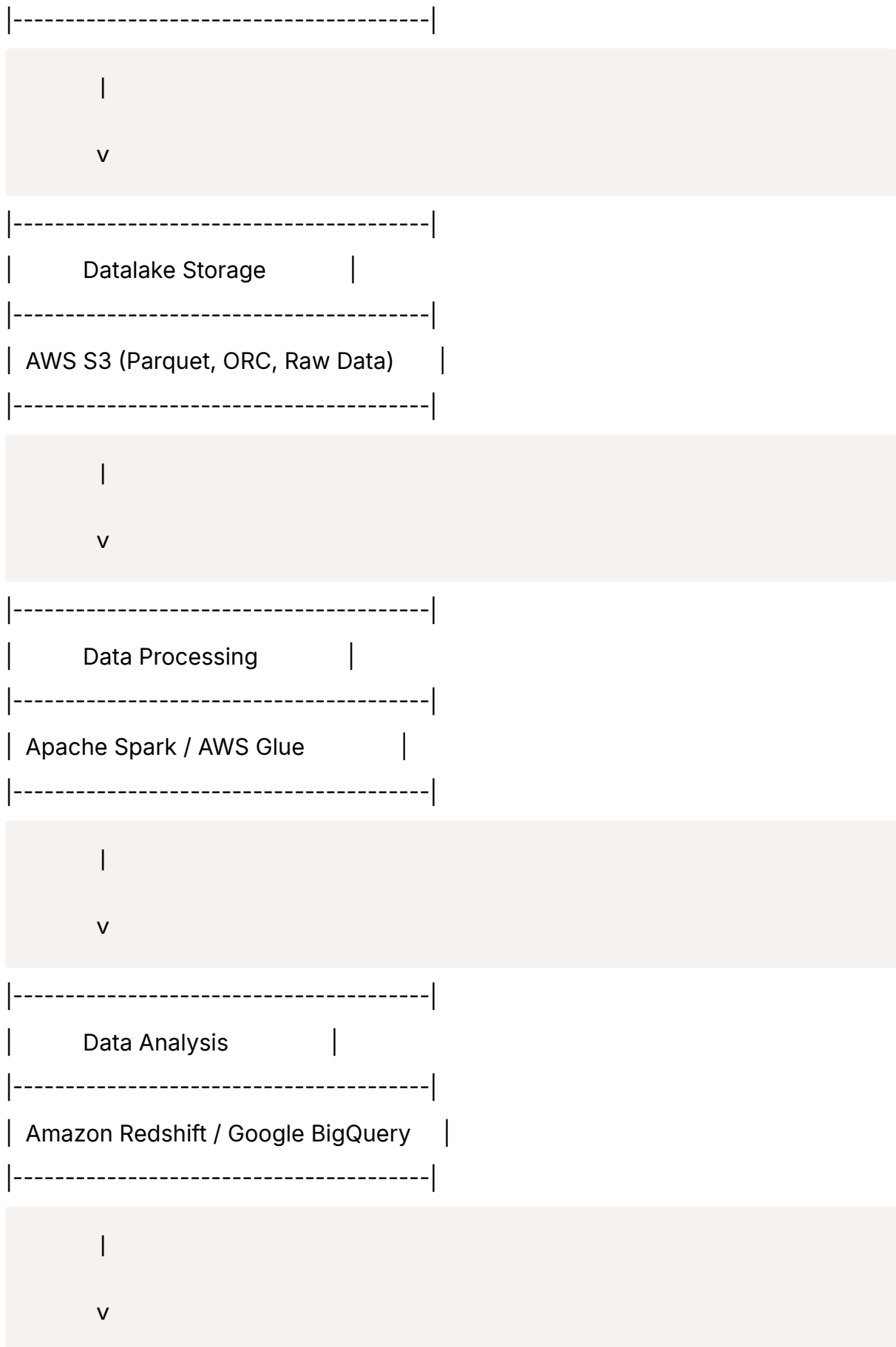
2. 데이터 수집부터 처리, 레포트 생성까지의 통합 프로세스 설계

- 데이터 수집:
 - 다양한 데이터 소스로부터 데이터를 수집하는 단계입니다. 이를 위해 ETL 도구를 사용하여 기존 정보계 시스템의 데이터를 Datalake로 이전하며, Apache Kafka나 AWS Kinesis와 같은 스트리밍 도구를 사용하여 실시간 데이터를 수집합니다.
- 데이터 저장:
 - Datalake는 AWS S3와 같은 객체 스토리지를 기반으로 구축됩니다. 구조화된 데이터는 Parquet 또는 ORC 형식으로, 비정형 데이터는 원시 형식으로 저장됩니다. 이러한 데이터는 추후 분석 및 레포트 생성에 사용됩니다.

- 데이터 처리 및 분석:
 - Apache Spark 또는 AWS Glue를 사용하여 Datalake에 저장된 데이터를 처리합니다. 데이터 처리 과정에서 필요한 변환 작업을 수행하고, 처리된 데이터를 분석 플랫폼(예: Amazon Redshift 또는 Google BigQuery)으로 전달합니다.
 - 데이터 검증: 데이터 처리 후, 데이터의 정확성과 일관성을 보장하기 위해 데이터 검증 단계를 포함합니다. 예를 들어, 데이터 품질을 평가하기 위한 데이터 프로파일링과 정합성 검사를 수행합니다.
- 레포트 생성:
 - ****BI 도구(예: Tableau, Power BI)****를 사용하여 분석된 데이터를 기반으로 레포트를 생성합니다. 이 과정에서 사용자 요구사항에 맞는 다양한 시각화 옵션을 제공하며, 데이터의 신뢰성을 보장하기 위해 생성된 레포트의 데이터를 추가 검증합니다.

Datalake 아키텍처 다이어그램:





Reporting & BI	

Tableau / Power BI	

3. Hedge 방안 및 이슈 해결

- 데이터 품질 저하 문제:
 - 데이터 품질 모니터링: 데이터 수집 단계에서부터 품질 검사를 실시하여 데이터 품질을 지속적으로 모니터링합니다. 품질 저하가 발생할 경우 자동으로 알림을 생성하여 신속하게 문제를 해결할 수 있도록 합니다.
 - 데이터 정제: 수집된 데이터를 Datalake에 저장하기 전에 데이터 클렌징 절차를 거쳐 데이터의 정확성을 높입니다. 이를 통해 데이터 품질 저하를 사전에 방지할 수 있습니다.
- 보안 문제:
 - 데이터 암호화: Datalake에 저장된 데이터를 AES-256과 같은 강력한 암호화 알고리즘으로 보호합니다. 저장 시 암호화뿐만 아니라 전송 중에도 TLS/SSL 암호화를 적용하여 데이터를 보호합니다.
 - 액세스 제어: 데이터에 대한 접근 권한을 **IAM(Identity and Access Management)**을 통해 엄격히 관리하고, 데이터에 접근할 수 있는 사용자와 권한을 최소화하여 보안을 강화합니다.
- 비용 증가 문제:
 - 데이터 수명 주기 관리: 저장된 데이터를 주기적으로 검토하여, 오래된 데이터는 Glacier와 같은 저비용 스토리지로 이동시키거나 삭제합니다. 이를 통해 Datalake 운영 비용을 절감할 수 있습니다.
 - 자원 최적화: 데이터 처리에 필요한 컴퓨팅 자원을 효율적으로 사용하기 위

해 자동 스케일링 기능을 도입하여, 필요할 때만 자원을 사용하도록 최적화합니다.

문제 상황 2: 온/오프라인 연계 실시간성 CDP(Customer Data Platform) 구축 및 데이터 통합 프로세스

B기업은 고객 행동 정보를 실시간으로 수집하고 통합하여, 개인 맞춤형 서비스를 제공하기 위해 CDP(Customer Data Platform)를 구축하려고 합니다. 이 과정에서 온/오프라인 데이터를 연계하여 실시간 데이터를 처리하고, 사용자에게 제공되는 인사이트와 레포트까지의 전체 프로세스를 설계해야 합니다.

1. 실시간 CDP 아키텍처 구성 및 데이터 통합 프로세스 설계

- 온/오프라인 데이터 수집:
 - 온라인 데이터: 웹사이트와 모바일 앱에서 발생하는 고객 행동 데이터를 Google Analytics 또는 Adobe Analytics와 같은 도구를 통해 수집합니다. 실시간 데이터를 CDP로 전송하기 위해 Kafka 또는 AWS Kinesis를 사용합니다.
 - 오프라인 데이터: 매장 방문 데이터, POS(Point of Sale) 시스템 데이터 등을 IoT 센서와 API를 통해 수집합니다. 오프라인 데이터는 주기적으로 CDP로 전송되며, 실시간성을 높이기 위해 스트리밍 처리도 지원됩니다.
- 데이터 저장 및 처리:
 - 수집된 데이터는 CDP의 중앙 저장소에 저장되며, AWS S3와 같은 객체 스토리지 또는 Amazon Redshift와 같은 데이터 웨어하우스에 저장됩니다. 이 데이터를 처리하기 위해 AWS Lambda와 Apache Spark를 사용하여 실시간 데이터를 처리하고, 정제된 데이터를 저장합니다.
- 데이터 통합: 다양한 소스에서 수집된 데이터를 ETL 과정을 통해 통합합니다. 실시간 데이터를 처리하면서, 데이터의 일관성을 유지하기 위해 데이터 스키마를 적용하고, 중복 데이터를 제거하는 작업을 수행합니다.
-

문제 상황 2: 데이터 분석 및 인사이트 도출

데이터 분석 및 인사이트 도출

- 데이터 분석:
 - 통합된 데이터를 분석하기 위해 Amazon Athena를 사용합니다. Athena는 S3에 저장된 데이터를 SQL 쿼리를 통해 분석할 수 있는 서버리스 분석 서비스로, 별도의 데이터 이동 없이 대규모 데이터를 빠르게 쿼리할 수 있습니다.
 - 실시간 분석을 위해 Amazon Kinesis Analytics를 도입하여, 스트리밍 데이터를 실시간으로 분석하고 패턴을 탐지합니다. 이를 통해 실시간으로 변동하는 고객 행동 데이터를 빠르게 분석하고, 인사이트를 도출할 수 있습니다.
- 인사이트 도출 및 예측 분석:
 - Amazon SageMaker를 사용하여 고객 행동 데이터를 기반으로 예측 모델을 구축합니다. SageMaker는 기계 학습 모델을 쉽게 개발, 훈련, 배포할 수 있는 서비스로, 고객 이탈 예측, 개인화 추천 모델 등을 생성하는 데 사용됩니다.
 - 고객 세분화를 위해 클러스터링 알고리즘을 사용하여 고객군을 정의하고, 각 군집에 맞는 맞춤형 서비스를 제공할 수 있는 전략을 도출합니다. 이를 통해 고객의 행동 패턴에 맞는 최적의 마케팅 캠페인을 설계할 수 있습니다.
- 레포트 생성 및 시각화:
 - 분석 결과를 Amazon QuickSight를 통해 시각화하여 사용자에게 제공할 레포트를 생성합니다. QuickSight는 AWS의 BI 도구로, 대시보드를 통해 실시간 데이터를 시각화하고, 다양한 차트를 활용하여 데이터를 효과적으로 전달할 수 있습니다.
 - 레포트는 주기적으로 생성되어 관련 부서에 자동으로 공유되며, 실시간 인사이트는 대시보드를 통해 실시간으로 모니터링할 수 있습니다. 이를 통해 마케팅 팀은 즉각적인 조치를 취할 수 있고, 경영진은 전략적 의사 결정을 지원받을 수 있습니다.

CDP 아키텍처 다이어그램:





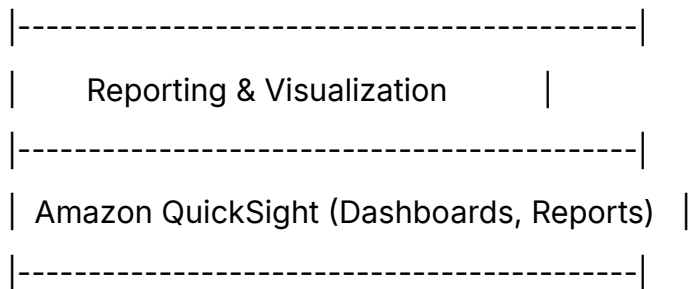
|

v



|

v



데이터 보정 및 레포트 제공 과정에서의 이슈 해결 방안

- 데이터 보정:
 - 데이터 정규화: 온/오프라인에서 수집된 다양한 형태의 데이터를 표준화하여 통합된 데이터 모델로 변환합니다. 이를 통해 데이터 형식 차이로 발생할 수 있는 일관성 문제를 해결합니다.
 - 중복 데이터 제거: 데이터 수집 과정에서 중복된 데이터를 제거하기 위해 데이터 디듀플리케이션 알고리즘을 적용합니다. 중복 데이터가 발생하는 시나리오를 미리 정의하고, 이를 실시간으로 처리할 수 있는 규칙 기반 필터링 시스템을 구축합니다.
 - 데이터 누락 복구: 누락된 데이터를 복구하기 위해 데이터 보강 기술을 적용합니다. 예를 들어, 누락된 값을 근사치로 대체하는 임putation 기법을 사용하거나, 이전 데이터 패턴을 기반으로 결측치를 추정하여 보완합니다.

- 레포트 신뢰성 확보:
- 데이터 검증 및 모니터링: 레포트 생성 전에 데이터의 신뢰성을 확보하기 위해 검증 단계를 추가합니다. 데이터 품질 지표(예: 정확성, 일관성, 적시성)를 설정하고, 이를 실시간으로 모니터링하여 문제가 발생하면 즉시 알림을 받도록 설정합니다.
- 자동화된 검증 프로세스: 레포트 생성 과정에서 자동화된 검증 프로세스를 구축하여 데이터 오류를 탐지하고, 수동으로 검토할 필요 없이 신뢰할 수 있는 데이터를 기반으로 레포트를 생성합니다.

데이터 보안 및 개인정보 보호 이슈 해결

- 데이터 암호화:
- 데이터 전송 중 암호화: 데이터 전송 시 TLS/SSL 암호화를 적용하여, 네트워크에서 발생할 수 있는 보안 위협으로부터 데이터를 보호합니다. 모든 데이터 전송은 HTTPS를 통해 이루어지며, 실시간 데이터 스트림은 Kinesis와 같은 암호화된 스트리밍 서비스로 처리됩니다.
- 데이터 저장 시 암호화: Datalake와 CDP에 저장되는 모든 데이터는 AES-256 암호화 알고리즘을 사용하여 암호화됩니다. AWS KMS(Key Management Service)를 사용하여 암호화 키를 관리하며, 데이터에 접근할 수 있는 권한을 최소화합니다.
- 접근 제어 및 감사:
- IAM(Identity and Access Management): 데이터에 접근할 수 있는 사용자를 엄격히 관리하기 위해 AWS IAM을 사용하여 세분화된 권한을 설정합니다. 역할 기반 접근 제어(RBAC)를 통해 민감한 데이터에 대한 접근을 제한하며, 필요에 따라 **MFA(Multi-Factor Authentication)**를 적용합니다.
- 감사 및 로그 관리: AWS CloudTrail을 사용하여 모든 데이터 액세스 및 변경 사항을 기록하고, 이를 기반으로 정기적인 감사를 수행합니다. 감사 로그를 통해 비정상적인 액세스나 보안 위협을 식별하고 대응할 수 있습니다.

결론

제안된 Datalake와 CDP 구축 방안은 A기업과 B기업의 데이터 관리 및 분석 역량을 크게 향상시킬 수 있는 종합적인 해결책입니다. Datalake는 구조화된 데이터와 비정형 데이터를 유연하

게 관리할 수 있는 인프라를 제공하며, CDP는 고객 데이터를 실시간으로 통합하여 개인 맞춤형 서비스를 제공하는 데 중요한 역할을 합니다. 각 단계에서 발생할 수 있는 기술적, 보안적 문제를 사전에 해결하기 위한 구체적인 전략을 통해, 기업의 데이터 활용도가 극대화되고 비즈니스 경쟁력이 강화될 것입니다.

카프카를 활용한 고객 데이터 처리: 실시간 vs. 배치 처리

문제 1. 어떤 방식을 선택해야 할까요?

실시간 처리 vs. 배치 처리, 어떤 방식을 선택할지는 고객 데이터의 특성과 활용 목적에 따라 달라 집니다.

실시간 처리 방식

- **장점:**
 - **즉각적인 반응:** 데이터가 생성되는 즉시 처리하여 실시간으로 인사이트를 얻을 수 있습니다.
 - **낮은 지연 시간:** 실시간 분석을 통해 빠르게 의사 결정을 내릴 수 있습니다.
 - **사용자 맞춤형 서비스:** 실시간 데이터를 기반으로 개인화된 서비스를 제공할 수 있습니다.
- **단점:**
 - **높은 시스템 자원 소모:** 실시간으로 데이터를 처리하기 위해 많은 시스템 자원이 필요합니다.
 - **복잡한 시스템 구축:** 실시간 처리 시스템을 구축하고 관리하는 것이 복잡합니다.

배치 처리 방식

- **장점:**
 - **낮은 시스템 자원 소모:** 실시간 처리에 비해 시스템 자원을 적게 사용합니다.
 - **복잡도 감소:** 실시간 처리 시스템에 비해 시스템 구축 및 관리가 상대적으로 간단합니다.
 - **오프라인 분석:** 대량의 데이터를 한 번에 처리하여 심층적인 분석을 수행할 수 있습니다.

- **단점:**
 - **지연:** 데이터 처리에 시간이 걸리기 때문에 실시간으로 데이터를 반영하기 어렵습니다.
 - **유연성 부족:** 실시간으로 변화하는 데이터에 유연하게 대응하기 어렵습니다.

선택 기준

- **데이터의 특성:**
 - **데이터 생성 속도:** 데이터 생성 속도가 빠르면 실시간 처리가 적합합니다.
 - **데이터 양:** 데이터 양이 매우 크면 배치 처리가 적합할 수 있습니다.
 - **데이터의 중요도:** 실시간으로 데이터를 분석하여 빠른 의사 결정이 필요한 경우 실시간 처리가 적합합니다.
- **활용 목적:**
 - **실시간 분석:** 실시간으로 데이터를 분석하여 인사이트를 얻고 싶다면 실시간 처리가 적합합니다.
 - **오프라인 분석:** 대량의 데이터를 분석하여 새로운 패턴을 발견하고 싶다면 배치 처리가 적합합니다.
- **시스템 환경:**
 - **시스템 자원:** 시스템 자원이 충분하지 않다면 배치 처리가 적합할 수 있습니다.
 - **시스템 복잡도:** 시스템 관리에 대한 부담이 크다면 배치 처리가 적합할 수 있습니다.

결론

최적의 방식은 위의 요소들을 종합적으로 고려하여 결정해야 합니다. 예를 들어, 고객 행동을 실시간으로 분석하여 개인화된 마케팅을 수행하고 싶다면 실시간 처리가 적합하고, 대량의 고객 데이터를 분석하여 고객 세분화를 수행하고 싶다면 배치 처리가 적합할 수 있습니다.

문제 2. Azure 기반 구축

Azure에서 카프카를 활용하여 고객 데이터를 처리하기 위한 몇 가지 방법을 소개합니다.

- **Azure Event Hubs:**
 - **장점:** 완전 관리형 서비스로 빠르게 구축 가능, 높은 처리량, 다양한 언어 지원
 - **단점:** 카프카와 호환되지 않는 기능 존재
- **Azure HDInsight:**
 - **장점:** 카프카 클러스터를 직접 관리, 유연성 높음
 - **단점:** 클러스터 관리에 대한 전문 지식 필요, 비용이 상대적으로 높음
- **Azure Synapse Analytics:**

- **장점:** 카프카와 통합하여 실시간 및 배치 처리 모두 가능, 서버리스 컴퓨팅 지원
- **단점:** 복잡한 시스템 구성 필요

구축 시 고려 사항

- **데이터 양 및 처리량:** 처리해야 하는 데이터 양과 처리량에 따라 적절한 서비스를 선택해야 합니다.
- **실시간성:** 실시간 처리가 필요한 경우 Azure Event Hubs 또는 Azure Synapse Analytics를 고려할 수 있습니다.
- **유연성:** 유연한 시스템 구성이 필요한 경우 Azure HDInsight를 고려할 수 있습니다.
- **비용:** 각 서비스의 가격 모델을 비교하여 예산에 맞는 서비스를 선택해야 합니다.

Azure 기반 구축 시 아키텍처 예시

- **실시간 처리:** IoT 디바이스에서 생성된 데이터를 Azure Event Hubs로 전송하고, Azure Stream Analytics를 통해 실시간으로 분석하여 Azure Cosmos DB에 저장
- **배치 처리:** Azure Event Hubs에 저장된 데이터를 Azure Databricks를 이용하여 배치로 처리하고, Azure Synapse Analytics에 저장

결론

Azure는 카프카를 활용하여 고객 데이터를 처리하기 위한 다양한 서비스를 제공합니다. 위에서 제시된 정보를 바탕으로 고객 데이터의 특성과 활용 목적에 맞는 최적의 솔루션을 선택하여 구축할 수 있습니다.

주의: 위 내용은 일반적인 정보이며, 실제 구축 시에는 고객의 요구사항 및 환경에 맞춰 상세한 설계 및 구현이 필요합니다.

더 자세한 정보가 필요하시면 언제든지 문의해주세요.

- 어떤 부분이 궁금하신가요?
- 어떤 종류의 고객 데이터를 처리하고 싶으신가요?
- 어떤 분석을 수행하고 싶으신가요?

다음과 같은 질문을 통해 더 구체적인 답변을 얻을 수 있습니다.

- 고객 데이터의 종류는 무엇인가요? (예: 웹 로그, 앱 사용 로그, IoT 데이터)
- 데이터 생성 속도는 어떠한가요? (예: 초당 수천 건, 분당 수십만 건)
- 데이터의 양은 얼마나 되나요? (예: 테라바이트, 페타바이트)
- 어떤 종류의 분석을 수행하고 싶으신가요? (예: 실시간 예측, 배치 분석, 머신러닝)
- 시스템의 요구사항은 무엇인가요? (예: 고가용성, 확장성, 보안)

이러한 정보를 바탕으로 더욱 맞춤형 솔루션을 제안해 드릴 수 있습니다.

데이터 처리 및 데이터베이스

- **1.데이터베이스 인덱스의 역할과 유형을 설명하고, 인덱스를 사용할 때의 장단점에 대해 논하세요.**
 - ○예시 답변: 인덱스는 데이터베이스에서 데이터를 빠르게 검색할 수 있도록 도와줍니다. B-트리 인덱스, 해시 인덱스 등 다양한 유형이 있으며, 인덱스를 사용하면 조회 속도가 빨라지지만, 인덱스를 유지하는 비용과 인덱스 생성 시간이 소요됩니다.
- **2.SQL과 NoSQL 데이터베이스의 차이점을 설명하고, 각각의 장단점을 비교하세요.**
 - ○예시 답변: SQL 데이터베이스는 관계형 데이터베이스로, 정형화된 스키마와 ACID 특성을 가집니다. NoSQL 데이터베이스는 비관계형으로, 스키마가 유연하며 대규모 데이터 처리에 적합합니다. SQL은 데이터 무결성과 복잡한 쿼리에 강점을 가지지만, NoSQL은 확장성과 성능에 유리합니다.

데이터 파이프라인 구축

- **3.데이터 파이프라인의 주요 구성 요소를 설명하고, 각 구성 요소의 역할을 기술하세요.**
 - ○예시 답변: 데이터 파이프라인의 주요 구성 요소는 데이터 소스, 데이터 수집, 데이터 변환, 데이터 저장, 데이터 분석 및 시각화입니다. 데이터 소스는 원천 데이터를 제공하며, 데이터 수집은 데이터를 추출합니다. 데이터 변환은 데이터를 정제하고 가공하며, 데이터 저장은 데이터를 적재합니다. 마지막으로 데이터 분석 및 시각화는 데이터를 분석하고 인사이트를 도출합니다.
- **4.ETL(Extract, Transform, Load) 과정의 각 단계를 설명하고, 각 단계에서 발생할 수 있는 문제점과 해결 방법을 논하세요.**
 - ○예시 답변: 추출 단계에서는 다양한 소스에서 데이터를 가져오며, 데이터의 불완전성이나 연결 문제 등이 발생할 수 있습니다. 변환 단계에서는 데이터 정제와 변환을 수행하며, 데이터 손실이나 변환 오류가 발생할 수 있습니다. 적재 단계에서는 데이터를 대상 시스템에 저장하며, 성능 저하나 데이터 정합성 문제가 발생할 수 있습니다. 이러한 문제들은 데이터 검증, 오류 로그 분석, 성능 모니터링 등을 통해 해결할 수 있습니다.

빅데이터 기술

- **5.Hadoop과 Spark의 차이점을 설명하고, 각각의 장단점에 대해 논하세요.**
- ○예시 답변: Hadoop은 분산 파일 시스템(HDFS)과 MapReduce 프로그래밍 모델을 사용하며, 대규모 데이터 처리에 적합하지만, 처리 속도가 느릴 수 있습니다. Spark는 인메모리 컴퓨팅을 활용하여 빠른 데이터 처리가 가능하며, 다양한 연산을 지원하지만, 메모리 사용량이 많을 수 있습니다.
- **6.데이터 엔지니어링에서 사용되는 주요 빅데이터 기술과 도구들을 나열하고, 각 도구의 용도와 특징을 설명하세요.**
- ○예시 답변: 주요 빅데이터 기술과 도구에는 Hadoop, Spark, Kafka, Flink, Airflow 등이 있습니다. Hadoop은 분산 저장과 처리, Spark는 빠른 데이터 처리, Kafka는 실시간 데이터 스트리밍, Flink는 실시간 데이터 스트리밍 및 배치 처리, Airflow는 워크플로우 관리에 사용됩니다.

데이터 품질 및 거버넌스

- **7.데이터 품질 관리의 중요성을 설명하고, 데이터 품질을 유지하기 위한 주요 방법론을 기술하세요.**
- ○예시 답변: 데이터 품질 관리는 정확한 데이터 분석과 의사결정을 위해 중요합니다. 주요 방법론에는 데이터 프로파일링, 데이터 정제, 데이터 검증, 데이터 표준화, 데이터 모니터링 등이 있습니다.
- **8.데이터 거버넌스의 개념을 설명하고, 데이터 거버넌스 프레임워크의 구성 요소를 나열하세요.**
- ○예시 답변: 데이터 거버넌스는 데이터 관리와 보안, 데이터 사용 규정을 정립하는 체계입니다. 주요 구성 요소에는 데이터 정책, 데이터 표준, 데이터 품질 관리, 데이터 보안, 데이터 아키텍처, 데이터 관리 책임 등이 포함됩니다.

빅데이터 플랫폼 구축/ 운영관련

~~ 고객의 환경에서, 네트워크 제약, 비용 등 이슈로 6개 사이트 동일한 빅데이터 플랫폼 구축하고 있습니다. 일부 구축, 일부 구축 중인 상태입니다.

○ 구성 요소 : 수집/워크플로우, 저장소 하둡, 데이터 처리 HIVE , GREENPLUM, 마트 DB, 시각화 Spotfire, 앱가상화, VM 환경(sandbox용) , 분석 플랫폼 등

시각화를 통한 정형/ 비정형 리포트를 제공하고 있음

○ 글로벌 데이터 수집/처리, 스토리지, data mart, 분석환경(앱 가상화), 시각화 툴, 분석툴, VM Sandbox, 멀티 instance 구성

○ data source - data staging, data 프로세싱/store/ analytics / 연동 등 전체적으로 구성을 제공해주고

ETL 단위 job 개발시 (파란색선) 에 대한 솔루션 선정 방안 제시

- 1안) 다양한 원천 데이터를 지원하는 단일 ETL 솔루션(EX. DATA STAGE, Informatica, teraStream등) 으로 단위 job 개발
- 2안)

안을 선정하고, 선정한 사유 등 등 기술

A기업과 B기업을 위한 대규모 정보계 차세대시스템 구축 프로젝트를 수행합니다. 이 프로젝트는 Datalake와 실시간 CDP(Customer Data Platform)를 구축하여, 기업의 데이터 활용 능력을 극대화하고 고객 맞춤형 서비스를 제공하는 것을 목표로 하고 있습니다. 데이터 수집부터 처리, 그리고 사용자에게 제공되는 레포트 생성까지의 전체 데이터 흐름을 고려한 구체적인 방안을 제시해야 합니다.

문제 상황 1: Datalake 신규 구축 및 데이터 수집부터 레포트 제공까지의 프로세스 설계

A기업은 기존의 정보계 시스템을 대체하고, 비정형 데이터를 포함한 대규모 데이터를 처리할 수 있는 새로운 Datalake를 구축하고자 합니다. 이 Datalake는 데이터 수집, 처리, 분석, 그리고 최종 사용자에게 제공되는 레포트 생성까지 모든 과정을 포함해야 합니다. 이 과정에서 여러 가지 기술적, 전략적 문제가 발생할 수 있습니다.

1. Datalake를 통한 기존 정보계 시스템 대체 방안:

- A기업의 기존 정보계 시스템은 주로 구조화된 데이터를 처리해왔습니다. 이제 Datalake를 통해 비정형 데이터, 스트리밍 데이터 등 다양한 데이터 유형을 통합하려고 합니다. 기존 정보계 시스템의 역할을 어떻게 Datalake로 대체할 수 있을지 방안을 제시하십시오.
- Datalake로 전환 시 데이터 중복, 일관성 문제, 성능 저하 등을 해결하기 위한 구체적인 방안을 설명하십시오.

2. 데이터 수집부터 처리, 레포트 생성까지의 통합 프로세스 설계:

- Datalake에서 다양한 데이터 소스로부터 데이터를 수집하고, 이를 처리하여 분석할 수 있는 전체 프로세스를 설계하십시오. 데이터의 수집, 저장, 처리, 분석, 레포트 생성까지의 흐름을 상세히 설명하십시오.
- 데이터 수집 과정에서 발생할 수 있는 품질 저하, 데이터 손실 등을 방지하기 위한 방법과, 최종 사용자에게 제공되는 레포트의 정확성을 보장하기 위한 데이터 검증 및 보정 방안을 제시하십시오.

3. 예상 이슈 및 이슈 해결 방안:

- Datalake 구축 과정에서 발생할 수 있는 대표적인 이슈(예: 데이터 품질 저하, 보안 문제, 비용 증가 등)를 예측하고, 이를 사전에 예방하거나 해결할 수 있는 완화 방안을 제시하십시오.

- 데이터 수집 및 처리 과정에서 발생할 수 있는 지연, 오류 등의 문제를 해결하기 위한 구체적인 전략을 설명하십시오.

문제 상황 2: 온/오프라인 연계 실시간성 CDP(Customer Data Platform) 구축 및 데이터 통합 프로세스

B기업은 고객 행동 정보를 실시간으로 수집, 처리하여 통합된 데이터를 기반으로 고객 맞춤형 서비스를 제공하기 위해 CDP(Customer Data Platform)를 구축하려고 합니다. 이 플랫폼은 온/오프라인 데이터를 실시간으로 연계하여, 수집된 데이터를 처리하고 최종적으로 사용자에게 제공되는 인사이트와 레포트를 포함하는 전체 프로세스를 설계해야 합니다.

1. 실시간 CDP 아키텍처 구성 및 데이터 통합 프로세스 설계:

- 온/오프라인에서 발생하는 고객 행동 정보를 실시간으로 수집, 처리, 통합할 수 있는 CDP 아키텍처를 제시하십시오. 이 아키텍처는 데이터의 수집부터 처리, 분석, 그리고 사용자에게 제공되는 인사이트와 레포트 생성까지의 전체 흐름을 포함해야 합니다.
- 실시간 데이터 통합 과정에서 발생할 수 있는 데이터 지연, 일관성 부족 등의 문제를 해결하기 위한 구체적인 기술적 방안을 설명하십시오.

2. 데이터 보정 및 레포트 제공 과정에서의 이슈 해결 방안:

- 다양한 소스로부터 수집된 고객 행동 데이터를 보정하여 일관성을 유지하는 방안을 제시하십시오. 데이터 보정 과정에서 발생할 수 있는 문제(예: 데이터 형식 차이, 중복 데이터 처리, 누락된 데이터 복구 등)를 해결하기 위한 구체적인 방안을 설명하십시오.
- CDP에서 생성된 데이터를 기반으로 최종 사용자에게 제공되는 레포트의 신뢰성과 정확성을 높이기 위한 검증, 모니터링, 보정 방안을 제시하십시오.

3. 데이터 보안 및 개인정보 보호 이슈 해결:

- CDP 구축 시 발생할 수 있는 데이터 보안 문제와 개인정보 보호 이슈를 해결하기 위한 방안을 제시하십시오. 특히, 실시간으로 수집되고 처리되는 고객 데이터의 보호를 위해 적용할 수 있는 기술적, 관리적 대책을 구체적으로 설명하십시오.

추가 지시사항:

- 제시된 방안이 A기업과 B기업의 현재 및 미래의 비즈니스 요구를 어떻게 충족할 수 있는지를 논리적으로 설명하십시오.
- 문제 해결을 위한 전제 조건이나 가정을 설정하고, 해당 전제 조건 또는 가정이 문제 해결에 어떻게 영향을 미치는지를 명확히 기술하십시오.
- 아키텍처 구성 방안 및 다이어그램을 텍스트로 설명하거나, 시각적 다이어그램으로 포함해도 좋습니다.

답안 표시

A기업과 B기업의 차세대 정보계 시스템 구축 방안

A기업: Datalake 기반 정보계 시스템 구축

1. Datalake를 통한 기존 정보계 시스템 대체 방안

- **기존 데이터 마이그레이션:** 기존 데이터베이스의 데이터를 Datalake로 이전하고, 필요한 변환 작업을 수행합니다.
- **데이터 레이어링:** Datalake를 여러 레이어로 구성하여 원시 데이터, 가공 데이터, 분석 결과 등을 분리하여 관리합니다.
- **메타데이터 관리 강화:** 모든 데이터에 대한 상세한 메타데이터를 관리하여 데이터 검색 및 활용 효율성을 높입니다.
- **데이터 중복, 일관성 문제 해결:**
 - **데이터 중복 검증:** 해싱, 비교 등을 통해 중복 데이터를 제거합니다.
 - **데이터 품질 규칙 설정:** 데이터 품질 기준을 정의하고 자동으로 검증하는 시스템을 구축합니다.
 - **마스터 데이터 관리:** 중요한 데이터는 마스터 데이터 관리 시스템을 통해 일관성을 유지합니다.
- **성능 저하 해결:**
 - **분산 저장 시스템:** 대용량 데이터를 분산하여 저장하여 성능 저하를 방지합니다.
 - **인덱싱:** 자주 사용되는 데이터에 대한 인덱스를 생성하여 검색 성능을 향상시킵니다.
 - **캐싱:** 자주 사용되는 데이터를 메모리에 캐싱하여 응답 시간을 단축합니다.

2. 데이터 수집부터 레포트 생성까지의 통합 프로세스 설계

- **데이터 수집:** 다양한 소스(RDBMS, NoSQL, 로그 파일 등)에서 데이터를 수집하고, 데이터 파이프라인을 통해 Datalake로 전송합니다.
- **데이터 저장:** Parquet, ORC 등의 형식으로 데이터를 저장하여 압축 및 효율적인 컬럼 기반 저장을 지원합니다.
- **데이터 처리:** ETL(Extract, Transform, Load) 프로세스를 통해 데이터를 정제, 변환하고 데이터 웨어하우스로 로드합니다.
- **데이터 분석:** SQL, Python, R 등을 활용하여 데이터를 분석하고 시각화합니다.
- **레포트 생성:** 분석 결과를 기반으로 다양한 형태의 보고서를 생성합니다.
- **데이터 품질 확보:**
 - **데이터 프로파일링:** 데이터의 통계적 특성을 분석하여 이상값을 탐지합니다.
 - **데이터 클렌징:** 오류 데이터를 수정하고 누락된 데이터를 보완합니다.
 - **데이터 품질 규칙:** 데이터 품질 기준을 정의하고 자동으로 검증합니다.

3. 예상 이슈 및 해결 방안

- **데이터 품질 저하:** 데이터 프로파일링, 클렌징, 품질 규칙 설정을 통해 해결합니다.
- **보안 문제:** 접근 권한 관리, 암호화, 침입 탐지 시스템 도입 등을 통해 해결합니다.
- **비용 증가:** 오픈소스 솔루션 활용, 클라우드 서비스 활용 등을 통해 비용을 절감합니다.
- **지연 및 오류:** 실시간 처리 시스템 도입, 오류 처리 메커니즘 구축 등을 통해 해결합니다.

B기업: 온/오프라인 연계 실시간 CDP 구축

1. 실시간 CDP 아키텍처 구성 및 데이터 통합 프로세스 설계

- **데이터 수집:** 웹 로그, 앱 로그, CRM 데이터 등 다양한 소스에서 실시간으로 데이터를 수집합니다.
- **데이터 처리:** Kafka, Kinesis 등의 스트리밍 플랫폼을 활용하여 실시간으로 데이터를 처리하고, 데이터 레이크에 저장합니다.
- **데이터 통합:** 데이터를 실시간으로 통합하고, 고객 ID를 기준으로 연결하여 360도 고객 프로필을 생성합니다.
- **인사이트 도출:** 분석 엔진을 활용하여 고객 행동 패턴을 분석하고 인사이트를 도출합니다.
- **레포트 생성:** 대시보드를 통해 실시간으로 고객 데이터를 시각화하고, 맞춤형 레포트를 생성합니다.
- **실시간 데이터 통합:**
 - **스트리밍 처리 플랫폼:** Kafka, Kinesis 등을 활용하여 실시간 데이터 처리를 수행합니다.
 - **데이터 복제:** 데이터를 실시간으로 복제하여 데이터 일관성을 유지합니다.

2. 데이터 보정 및 레포트 제공 과정에서의 이슈 해결 방안

- **데이터 보정:** 데이터 품질 규칙을 설정하고, 데이터 클렌징 프로세스를 자동화합니다.
- **레포트 신뢰성:** 데이터 검증, 모니터링 시스템을 구축하고, 정기적인 검증을 수행합니다.

3. 데이터 보안 및 개인정보 보호

- **데이터 암호화:** 모든 데이터를 암호화하여 저장하고 전송합니다.
- **접근 권한 관리:** 사용자별로 필요한 데이터에 대한 접근 권한을 세분화합니다.
- **개인정보 보호 규정 준수:** GDPR, CCPA 등 관련 법규를 준수합니다.

추가 지시사항에 대한 답변

- **비즈니스 요구 충족:** 위에서 제시된 방안은 각 기업의 비즈니스 목표에 맞춰 데이터 활용 능력을 극대화하고, 고객 맞춤형 서비스 제공을 가능하게 합니다.

- **전제 조건:**

- 충분한 IT 인프라 구축
- 데이터 품질 관리 체계 마련
- 보안 시스템 구축
- 사용자 교육

- **전제 조건의 영향:** 전제 조건이 충족되지 않으면 시스템 구축 및 운영에 어려움이 발생할 수 있습니다.

결론

A기업과 B기업의 차세대 정보계 시스템 구축은 단순한 시스템 교체를 넘어, 데이터 기반 의사 결정을 위한 핵심 인프라를 구축하는 것입니다. 위에서 제시된 방안을 바탕으로 각 기업의 특성과 요구사항에 맞는 최적의 시스템을 구축하여 데이터 활용 가치를 극대화할 수 있을 것입니다.

주요 키워드	문제	예시답변
데이터 처리 및 데이터베이스	데이터베이스 인덱스의 역할과 유형을 설명하고, 인덱스를 사용할 때의 장단점에 대해 논하세요.	인덱스는 데이터베이스에서 데이터를 빠르게 검색할 수 있도록 도와줍니다. B-트리 인덱스, 해시 인덱스 등 다양한 유형이 있으며, 인덱스를 사용하면 조회 속도가 빨라지지만, 인덱스를 유지하는 비용과 인덱스 생성 시간이 소요됩니다.
	SQL과 NoSQL 데이터베이스의 차이점을 설명하고, 각각의 장단점을 비교하세요.	SQL 데이터베이스는 관계형 데이터베이스로, 정형화된 스키마와 ACID 특성을 가집니다. NoSQL 데이터베이스는 비관계형으로, 스키마가 유연하며 대규모 데이터 처리에 적합합니다. SQL은 데이터 무결성과 복잡한 쿼리에 강점을 가지지만, NoSQL은 확장성과 성능에 유리합니다.
데이터 파이프라인 구축	데이터 파이프라인의 주요 구성 요소를 설명하고, 각 구성 요소의 역할을 기술하세요.	데이터 파이프라인의 주요 구성 요소는 데이터 소스, 데이터 수집, 데이터 변환, 데이터 저장, 데이터 분석 및 시각화입니다. 데이터 소스는 원천 데이터를 제공하며, 데이터 수집은 데이터를 추출합니다. 데이터 변환은 데이터를 정제하고 가공하며, 데이터 저장은 데이터를 적재합니다. 마지막으로 데이터 분석 및 시각화는 데이터를 분석하고 인사이트를 도출합니다.
	ETL(Extract, Transform, Load) 과정의 각 단계를 설명하고, 각 단계에서 발생할 수 있는 문제점과 해결 방법을 논하세요.	추출 단계에서는 다양한 소스에서 데이터를 가져오며, 데이터의 불완전성이나 연결 문제 등이 발생할 수 있습니다. 변환 단계에서는 데이터 정제와 변환을 수행하며, 데이터 손실이나 변환 오류가 발생할 수 있습니다. 적재 단계에서는 데이터를 대상 시스템에 저장하며, 성능 저하나 데이터 정합성 문제가 발생할 수 있습니다.

		이러한 문제들은 데이터 검증, 오류 로그 분석, 성능 모니터링 등을 통해 해결할 수 있습니다.
빅데이터 기술	Hadoop과 Spark의 차이점을 설명하고, 각각의 장단점에 대해 논하세요.	Hadoop은 분산 파일 시스템(HDFS)과 MapReduce 프로그래밍 모델을 사용하며, 대규모 데이터 처리에 적합하지만, 처리 속도가 느릴 수 있습니다. Spark는 인 메모리 컴퓨팅을 활용하여 빠른 데이터 처리가 가능하며, 다양한 연산을 지원하지만, 메모리 사용량이 많을 수 있습니다.
	데이터 엔지니어링에서 사용되는 주요 빅데이터 기술과 도구들을 나열하고, 각 도구의 용도와 특징을 설명하세요.	주요 빅데이터 기술과 도구에는 Hadoop, Spark, Kafka, Flink, Airflow 등이 있습니다. Hadoop은 분산 저장과 처리, Spark는 빠른 데이터 처리, Kafka는 실시간 데이터 스트리밍, Flink는 실시간 데이터 스트리밍 및 배치 처리, Airflow는 워크플로우 관리에 사용됩니다.
데이터 품질 및 거버넌스	데이터 품질 관리의 중요성을 설명하고, 데이터 품질을 유지하기 위한 주요 방법론을 기술하세요.	데이터 품질 관리는 정확한 데이터 분석과 의사결정을 위해 중요합니다. 주요 방법론에는 데이터 프로파일링, 데이터 정제, 데이터 검증, 데이터 표준화, 데이터 모니터링 등이 있습니다.
	데이터 거버넌스의 개념을 설명하고, 데이터 거버넌스 프레임워크의 구성 요소를 나열하세요.	데이터 거버넌스는 데이터 관리와 보안, 데이터 사용 규정을 정립하는 체계입니다. 주요 구성 요소에는 데이터 정책, 데이터 표준, 데이터 품질 관리, 데이터 보안, 데이터 아키텍처, 데이터 관리 책임 등이 포함됩니다.

이름	No.	질문	답안	주요 키워드
박성근	1	해외 지사를 두고 있는 글로벌 기업에서 각 지사에서 발생하는 데이터를 통합하여 데이터레이크를 구성하려고 할 때 IT 관점에서 고려해야 되는 사항은 무엇인가요?	1.데이터 거버넌스 (Data Governance): ○정책 및 규제 준수: 각 국가 및 지역의 데이터 보호 법규 (예: GDPR, CCPA)를 준수하기 위한 정책을 수립하고 준수해야 합니다. ○데이터 품질 관리: 일관된 데이터 품질을	데이터레이크 구성방안

유지하기 위해 데이터 정제, 데이터 중복 제거 및 데이터 표준화를 실시해야 합니다.

2.데이터 보안 (Data Security):

○암호화: 데이터 저장 시와 전송 시의 암호화를 적용하여 데이터 보안을 강화해야 합니다.

○액세스 제어: 민감 데이터에 대한 접근 권한을 제어하고, 역할 기반 접근 제어 (RBAC)를 구현해야 합니다.

○침입 탐지 및 대응: 실시간으로 데이터 접근을 모니터링하고, 잠재적인 침입 시도를 탐지하여 대응할 수 있는 시스템을 갖추어야 합니다.

3.데이터 인프라 (Data Infrastructure):

○클라우드 기반 데이터 레이크: 글로벌 스케일의 데이터 저장 및 처리를 위해 클라우드 기반 솔루션 (예: AWS S3, Azure Data Lake, Google Cloud Storage)을 고려해야 합니다.

○데이터 전송: 지사에서 중앙 데이터 레이크로 데이터를 안전하고 효율적으로 전송할 수 있는 네트워크 인프라를 구축해야 합니다.

○지연 최소화: 데이터 전송 시의 지연을 최소화하고, 전송 속도를 최적화할 수 있는 방법을 찾아야 합니다.

4.데이터 통합 (Data Integration):

○ETL 프로세스: 데이터를 추출(Extract), 변환(Transform), 적재(Load)하는 ETL 파이프라인을 설계하고 자동화해야 합니다.

○실시간 데이터 스트리밍: 실시간 데이터를 처리하기 위한 스트리밍 플랫폼 (예:

Apache Kafka)을 구축하여 실시간 데이터 통합을 지원해야 합니다.

5.데이터 카탈로그 및 메타데이터 관리

(Data Catalog and Metadata

Management):

○데이터 카탈로그: 모든 데이터를 쉽게 검색하고 활용할 수 있도록 데이터 카탈로그를 구축하여 메타데이터를 관리해야 합니다.

○데이터 계보 (Data Lineage): 데이터의 출처, 이동 경로 및 변환 내역을 추적할 수 있는 시스템을 도입하여 데이터 계보를 관리해야 합니다.

6.성능 및 확장성 (Performance and Scalability):

○확장성: 데이터 양이 증가함에 따라 쉽게 확장할 수 있는 인프라를 설계해야 합니다.

○성능 최적화: 데이터 처리 및 쿼리 성능을 최적화하기 위해 적절한 인덱싱, 파티셔닝 및 캐싱 전략을 적용해

			<p>야 합니다.</p> <p>7.백업 및 복구 (Backup and Recovery):</p> <p>○정기 백업: 데이터의 정기적인 백업을 통해 데이터 손실에 대비해야 합니다.</p> <p>○재해 복구: 재해 발생 시 신속하게 데이터 복구가 가능하도록 재해 복구 계획을 수립하고 테스트해야 합니다.</p> <p>8.데이터 분석 및 활용 (Data Analytics and Utilization):</p> <p>○분석 플랫폼 통합: 데이터 레이크와 분석 플랫폼 (예: Apache Spark, Presto, Apache Flink)을 통합하여 데이터 분석을 지원해야 합니다.</p> <p>○머신러닝 및 AI: 데이터 레이크 상에서 머신러닝 및 AI 모델을 구축하고 운영할 수 있는 환경을 마련해야 합니다.</p>	
박성근	2	빅데이터 실시간 처리 데이터 아키텍처 수립 시 고려해야 되는 사항은 무엇인가요?	<p>1.데이터 수집 (Data Ingestion):</p> <p>○다양한 데이터 소스 지원: 여러 소스(예: IoT 디바이스, 웹 로그, 소셜 미디어 등)에서 데이터를 실시간으로 수집할 수 있는 인프라를 구축해야 합니다.</p> <p>○스트리밍 플랫폼: Apache Kafka, Apache Pulsar, Amazon Kinesis와 같은 스트리밍 플랫폼을 사용하여 높은 처리량과 저지연 데이터 수</p>	실시간 처리

집을 보장해야 합니다.

2.데이터 처리 (Data Processing):

○스트림 처리 엔진:

Apache Flink,
Apache Spark
Streaming, Apache
Storm과 같은 스트림
처리 엔진을 사용하여
실시간 데이터 처리를
구현해야 합니다.

○저지연 처리: 데이터
를 신속하게 처리하기
위해 저지연 처리 프레
임워크와 기술을 선택
해야 합니다.

○상태 관리: 실시간
데이터 처리 중 상태를
유지하고 관리할 수 있
는 방법을 고려해야 합
니다.

3.데이터 저장 (Data Storage):

○빠른 읽기/쓰기를 지
원하는 저장소:

HBase, Cassandra,
Redis와 같은 NoSQL
데이터베이스를 사용
하여 실시간 데이터를
저장하고 검색할 수 있
어야 합니다.

○시간 기반 데이터 저
장소: InfluxDB,
TimescaleDB와 같은
시계열 데이터베이스
를 사용하여 시간 기반
데이터를 효율적으로
저장하고 관리할 수 있
습니다.

4.데이터 분석 및 시각 화 (Data Analytics and Visualization):

○실시간 분석 도구:

실시간 분석을 지원하
는 도구 (예: Apache
Druid, ClickHouse)

를 사용하여 실시간 쿼리와 분석을 수행해야 합니다.

○시각화 도구:

Grafana, Kibana와 같은 시각화 도구를 사용하여 실시간 데이터 시각화를 제공해야 합니다.

5. 확장성

(Scalability):

○수평적 확장: 시스템이 증가하는 데이터 양을 처리할 수 있도록 수평적으로 확장 가능한 아키텍처를 설계해야 합니다.

○자동 확장:

Kubernetes와 같은 오케스트레이션 도구를 사용하여 필요에 따라 리소스를 자동으로 확장할 수 있도록 해야 합니다.

6. 신뢰성 및 가용성

(Reliability and Availability):

○데이터 손실 방지: 데이터 손실을 방지하기 위해 데이터 복제 및 백업 전략을 구현해야 합니다.

○고가용성: 장애 발생 시에도 시스템이 지속적으로 작동할 수 있도록 고가용성 아키텍처를 설계해야 합니다.

7. 보안 (Security):

○데이터 암호화: 데이터 전송 시와 저장 시 모두 암호화를 적용하여 데이터 보안을 강화해야 합니다.

○접근 제어: 역할 기반 접근 제어 (RBAC)를 사용하여 민감 데이

			<p>터에 대한 접근을 관리해야 합니다.</p> <p>○감사 및 모니터링: 실시간 데이터 접근 및 변경 내역을 모니터링하고 감사 로그를 유지해야 합니다.</p> <p>8.유지보수 및 운영 (Maintenance and Operations):</p> <p>○모니터링 및 경고 시스템: Prometheus, ELK Stack과 같은 모니터링 및 로깅 도구를 사용하여 시스템 성능을 모니터링하고 문제 발생 시 경고를 받을 수 있어야 합니다.</p> <p>○자동화된 배포: CI/CD 파이프라인을 구축하여 코드 변경 사항을 자동으로 배포하고, 지속적인 통합 및 배포를 가능하게 해야 합니다.</p> <p>9.지연 허용 범위 (Latency Tolerance):</p> <p>○저지연 처리 필요성: 애플리케이션 요구사항에 따라 허용 가능한 지연 범위를 설정하고, 이를 충족하기 위한 기술을 선택해야 합니다.</p> <p>○배치 처리와의 통합: 실시간 처리와 배치 처리를 결합하여 하이브리드 데이터 처리 아키텍처를 구현할 수 있습니다.</p>	
정경주	5	온프레미스 데이터 웨어하우스 시스템을 클라우드로 전환 시 가장 유념해	<p>1. 비용 및 ROI</p> <p>클라우드 기반 빅데이터 플랫폼은 초기 투자 비용이 적게 들 수 있지만, 장기적인 운영</p>	클라우드 전환 방안

야 할 고려 사항을
제시해 주세요.

비용은 사용량 및 데이터 규모에 따라 달라질 수 있습니다.

예상되는 비용을 정확하게 계산하고, 클라우드 서비스 제공업체의 가격 모델을 꼼꼼하게 비교 분석해야 합니다. 또한, 데이터 마이그레이션, 시스템 관리, 보안 강화 등 클라우드 전환으로 인해 발생하는 추가 비용도 고려해야 합니다.

투자 수익률(ROI)을 계산하여 클라우드 전환의 경제적 이득을 명확하게 제시해야 합니다.

2. 성능 및 확장성

클라우드 기반 빅데이터 플랫폼은 온프레미스 시스템보다 뛰어난 확장성을 제공하며, 워크로드 증가에 따라 용량을 빠르게 확장할 수 있습니다.

하지만, 네트워크 대역폭 및 레이턴시가 성능에 영향을 미칠 수 있으므로, 클라우드 서비스 제공업체의 네트워크 인프라를 평가해야 합니다.

또한, 데이터 웨어하우스 시스템의 처리 능력과 저장 용량 요구 사항을 정확하게 예측하고, 이에 맞는 클라우드 리소스를 선택해야 합니다.

3. 보안 및 규정 준수

클라우드 기반 빅데이터 플랫폼은 민감한 데

이터를 저장하기 때문에, 강력한 보안 및 규정 준수 요구 사항을 충족해야 합니다. 데이터 암호화, 액세스 제어, 데이터 손실 방지 등 다양한 보안 기능을 제공하는 클라우드 서비스 제공업체를 선택해야 합니다. 또한, 개인정보보호법 (GDPR), 산업별 규제 등 관련 규정을 준수해야 합니다.

4. 데이터 마이그레이션 및 통합
온프레미스 데이터를 클라우드로 마이그레이션하는 과정은 복잡하고 시간이 많이 걸릴 수 있습니다. 데이터 마이그레이션 전략을 수립하고, 데이터 손실 및 오류를 방지하기 위한 프로세스를 정의해야 합니다. 또한, 기존 온프레미스 시스템과 클라우드 기반 빅데이터 플랫폼을 통합하고, 데이터 흐름을 원활하게 연결해야 합니다.

5. 시스템 관리 및 운영
클라우드 기반 빅데이터 플랫폼은 온프레미스 시스템보다 관리 및 운영이 간편할 수 있습니다. 하지만, 클라우드 서비스 제공업체의 서비스 수준 계약(SLA)을 확인하고, 시스템 가용성 및 성능을 모니터링해

			<p>야 합니다.</p> <p>또한, 클라우드 환경에서 발생하는 보안 위협 및 새로운 기술 트렌드를 지속적으로 파악하고 대응해야 합니다.</p> <p>6. 인력 및 기술</p> <p>클라우드 기반 빅데이터 플랫폼을 관리 및 운영するには, 클라우드 컴퓨팅 및 관련 기술에 대한 전문 지식을 갖춘 인력이 필요합니다.</p> <p>직원 교육 및 개발을 통해 필요한 기술 능력을 갖추도록 지원해야 합니다.</p> <p>또한, 클라우드 서비스 제공업체의 지원 서비스를 활용하여 전문적인 도움을 받을 수도 있습니다.</p> <p>7. 변경 관리</p> <p>클라우드 기반 빅데이터 플랫폼으로 전환하는 것은 조직의 IT 인프라 및 운영 방식에 큰 변화를 가져올 수 있습니다.</p> <p>직원들에게 변화에 대한 교육을 제공하고, 새로운 시스템 사용에 대한 지원을 제공해야 합니다</p>	
정경주	6	<p>다양한 데이터 플랫폼 중에서 고객에게 적합한 플랫폼을 선정하기 위해 중요하게 살펴야 할 항목을 제시해 주세요.</p>	<p>1. 요구 사항:</p> <p>기능: 플랫폼이 사용자의 모든 기능 요구 사항을 충족하는지 확인해야 합니다.</p> <p>성능: 플랫폼이 사용자의 처리량, 데이터 규모, 성능 요구 사항을</p>	<p>데이터플랫폼 선정 조건</p>

처리할 수 있는지 확인해야 합니다.

확장성: 플랫폼이 사용자의 데이터 및 사용자 트래픽 증가에 따라 확장될 수 있는지 확인해야 합니다.

보안: 플랫폼이 사용자 데이터를 보호하고 액세스 제어, 암호화, 백업 및 재해 복구와 같은 강력한 보안 기능을 제공하는지 확인해야 합니다.

규정 준수: 플랫폼이 사용자의 산업 및 규제 요구 사항을 준수하는지 확인해야 합니다.

2. 비용:

초기 투자 비용: 플랫폼 라이선스, 하드웨어, 소프트웨어 및 기타 초기 비용을 고려해야 합니다.

지속적인 운영 비용: 플랫폼 사용료, 인프라 비용, 유지 관리 비용 및 기타 지속적인 비용을 고려해야 합니다.

가격 모델: 플랫폼 제공업체가 제공하는 다양한 가격 모델 (예: 사용량 기반, 구독 기반)을 비교해야 합니다.

3. 사용 편의성:

사용자 인터페이스: 플랫폼의 사용자 인터페이스가 사용하기 쉽고 직관적인지 확인해야 합니다.

문서 및 교육 자료: 플랫폼 사용을 배우고 문제를 해결하는 데 도움이 되는 충분한 문서

및 교육 자료가 있는지 확인해야 합니다.

고객 지원: 플랫폼 제공업체가 문제 해결 및 지원을 위해 우수한 고객 지원을 제공하는지 확인해야 합니다.

4. 기술 스택:

호환성: 플랫폼이 사용자의 기존 기술 스택 및 시스템과 호환되는지 확인해야 합니다.

개발 도구 및 언어: 플랫폼이 사용자의 선호하는 개발 도구 및 언어를 지원하는지 확인해야 합니다.

통합: 플랫폼이 사용자의 기존 애플리케이션 및 시스템과 통합될 수 있는지 확인해야 합니다.

5. 공급업체:

신뢰성: 플랫폼 제공업체가 신뢰할 수 있고 재무적으로 안정적인지 확인해야 합니다.

실적: 플랫폼 제공업체가 사용자와 유사한 업계에서 성공적인 실적을 가지고 있는지 확인해야 합니다.

고객 만족도: 플랫폼 제공업체의 고객 만족도 및 리뷰를 확인해야 합니다.

6. 기타 고려 사항:

장기적인 로드맵: 플랫폼 제공업체가 미래에 어떤 계획을 가지고 있는지 확인하고, 플랫폼이 사용자의 장기적인 요구 사항을 충족할 수

			<p>있는지 확인해야 합니다.</p> <p>커뮤니티: 플랫폼 사용자 커뮤니티가 활발하고 도움이 되는지 확인해야 합니다.</p> <p>오픈 소스: 플랫폼이 오픈 소스인 경우, 사용자가 코드를 검토하고 수정할 수 있는지 확인해야 합니다.</p>	
윤준구	상황제시	<p>[고객 상황]</p> <p>고객 상황 : 글로벌 패션 브랜드 '스타일 잇'은 온라인 쇼핑몰 운영 데이터를 활용하여 고객 맞춤형 마케팅 전략을 수립하고자 한다.</p> <p>Pain Point : 막대한 구매 데이터와 고객 행동 데이터를 효율적으로 분석하고 활용하기 어려움</p> <p>업무 요건 :</p> <ul style="list-style-type: none"> - 다양한 데이터 소스(웹로그, 거래 데이터, CRM 등)를 통합하고 처리할 수 있는 데이터 파이프라인 구축 - 고객 구매 패턴, 인구 통계, 선호도 분석을 위한 머신러닝 모델 개발 - 실시간 고객 추천 및 맞춤형 프로모션 자동화 시스템 구축 - 데이터 기반 마케팅 캠페인 효과 	<p>[업무 시나리오 1]</p> <ol style="list-style-type: none"> 1. 다양한 데이터 소스 통합: 웹로그, 거래 데이터, CRM 등 다양한 데이터 소스를 통합하고 처리할 수 있는 데이터 파이프라인 구축 2. 실시간 데이터 처리: 구매, 검색, 클릭 등 사용자 행동 데이터를 실시간으로 처리 및 분석 3. 데이터 저장 및 관리: 대규모 데이터 저장 및 관리를 위한 분산 데이터베이스 구축 4. 데이터 품질 관리: 데이터 누락, 오류, 불일치 등 데이터 품질 문제 해결 및 데이터 정제 5. 고객 분석: 고객 구매 패턴, 인구 통계, 선호도 등을 분석하기 위한 머신러닝 모델 개발 6. 실시간 추천: 고객 맞춤형 상품 추천을 위한 실시간 추천 시스템 구축 7. 마케팅 자동화: 고객 세분화, 타겟팅, 캠페인 자동화 등 마케팅 자동화 시스템 구축 	

		<p>측정 및 최적화 성공 지표 :</p> <ul style="list-style-type: none"> - 구매 전환율 증대 - 고객 유지율 향상 - 마케팅 투자 수익률 (ROI) 최대화 	<p>8.A/B 테스트: 다양한 마케팅 전략의 효과를 비교 분석하기 위한 A/B 테스트 플랫폼 구축</p> <p>9.데이터 시각화: 분석 결과를 시각화하고 이해하기 쉽게 제시하기 위한 데이터 시각화 도구 활용</p> <p>10.보안 및 개인정보 보호: 고객 데이터 보안 및 개인정보 보호를 위한 강력한 보안 시스템 구축</p>	
윤준구	7	<p>업무 시나리오 1에서 ①②에 대해 수집 파이프라인을 2가지 안으로 설계하세요. 각 각의 안에 대해서 선정한 사유를 기술하세요.</p>		수집파이프라인 (실시간, 저장/관리)
윤준구	8	<p>문제 1에서 설계된 수집 레이어 기반으로 ③④을 기준으로 data warehouse/data lake 레이어를 설계하세요</p>		DW, DataLake 설계방안
조항륜	9	<p>사례 문제 1번 : 온라인 소매업체 데이터 파이프라인 구축</p> <p>시나리오</p> <p>당신은 큰 온라인 소매업체에서 데이터 엔지니어로 일하고 있습니다. 이 소매업체는 다양한 소스(웹사이트, 모바일 앱, 물리적 상점 등)에서 데이터를 수집하고 있으</p>	<p>단계별 접근 방법</p> <p>1.데이터 수집 및 통합: Kafka Connect를 사용하여 다양한 데이터 소스에서 데이터를 수집하고, 이 데이터를 Kafka 토픽에 게시합니다.</p> <p>2.데이터 정제 및 변환: Apache Spark를 사용하여 수집된 데이터를 정제하고 변환합니다.</p> <p>3.실시간 데이터 스트리밍: Apache Kafka</p>	

며, 이 데이터를 활용하여 실시간 추천 시스템을 운영하고자 합니다. 현재 데이터는 여러 개의 서로 다른 데이터베이스와 파일 시스템에 분산되어 있습니다. 또한, 기존의 데이터 파이프라인은 느리고 비효율적이어서 실시간 처리가 어렵습니다.

요구 사항

- 1.데이터 통합: 모든 소스에서 데이터를 수집하여 중앙 데이터 웨어하우스로 통합해야 합니다.
 - 2.실시간 처리: 실시간 추천 시스템을 운영하기 위해 실시간 데이터 스트리밍을 구현해야 합니다.
 - 3.확장성: 데이터 파이프라인은 트래픽 증가에 대비해 확장 가능해야 합니다.
 - 4.신뢰성: 데이터 손실 없이 안정적으로 데이터를 처리해야 합니다.
- 해결해야 할 문제
- 1.데이터 소스 연결: 여러 소스에서 데이터를 수집하기 위해 필요한 기술과 도구를 선택해야 합니다.
 - 2.데이터 정제: 수

를 사용하여 실시간 데이터 스트리밍을 구현합니다.

4.데이터 저장:

Snowflake를 데이터 웨어하우스로 선택하고, 정제된 데이터를 저장합니다.

5.모니터링 및 알림:

Prometheus와 Grafana를 사용하여 데이터 파이프라인의 상태를 모니터링하고, 문제가 발생하면 알림을 받습니다.

계별 해결 솔루션의 선택 이유와 장단점

1. 데이터 수집 및 통합 선택한 솔루션: Kafka Connect

선택 이유:

●Kafka Connect는 다양한 소스에서 데이터를 쉽게 수집하고, Kafka로 데이터를 전송할 수 있는 커넥터를 제공합니다.

●다양한 커넥터를 통해 쉽게 확장할 수 있으며, 분산 처리와 스케일링이 용이합니다.

장점:

●확장성: 필요에 따라 새로운 커넥터를 추가하여 쉽게 확장할 수 있습니다.

●호환성: 다양한 데이터 소스(MySQL, PostgreSQL, Elasticsearch 등)와 호환됩니다.

●실시간 처리: 실시간으로 데이터를 수집하고 처리할 수 있습니다.

단점:

집된 데이터를 분석 가능하게 정제해야 합니다.

3.데이터 스트리밍: 실시간 처리를 위한 스트리밍 솔루션을 설계하고 구현해야 합니다.

4.데이터 저장: 통합된 데이터를 저장할 적절한 데이터 웨어하우스를 선택하고 설계해야 합니다.

5.데이터 분석 및 활용: 데이터 과학 팀이 실시간으로 데이터를 분석하고 활용할 수 있도록 지원해야 합니다.

고려할 기술 스택

●데이터 수집: Apache Nifi, Logstash, Kafka Connect

●데이터 정제: Apache Spark, Apache Flink

●데이터 스트리밍: Apache Kafka, Amazon Kinesis

●데이터 저장: Amazon Redshift, Google BigQuery, Snowflake

●모니터링 및 알림: Prometheus, Grafana

●복잡성: 다양한 커넥터를 설정하고 관리하는 데 시간이 걸릴 수 있습니다.

●자원 사용량: 많은 데이터를 처리할 때 자원 사용량이 증가할 수 있습니다.

2. 데이터 정제 및 변환

선택한 솔루션:

Apache Spark

선택 이유:

●Apache Spark는 대규모 데이터 처리를 위한 강력한 도구로, 다양한 데이터 소스에서 데이터를 읽고, 변환하고, 정제할 수 있습니다.

●배치 처리와 스트리밍 처리를 모두 지원하여 유연한 데이터 파이프라인을 구축할 수 있습니다.

장점:

●성능: 인메모리 컴퓨팅을 통해 빠른 데이터 처리가 가능합니다.

●유연성: 다양한 데이터 소스와 통합하여 데이터를 처리할 수 있습니다.

●확장성: 클러스터 환경에서 쉽게 확장할 수 있습니다.

단점:

●복잡성: 설정과 사용법이 복잡할 수 있습니다.

●자원 사용량: 대규모 데이터를 처리할 때 많은 자원이 필요할 수 있습니다.

3. 실시간 데이터 스트리밍

선택한 솔루션:

Apache Kafka와

Apache Flink

선택 이유:

●Apache Kafka는 높은 처리량과 낮은 지연 시간을 제공하며, 실시간 데이터 스트리밍에 적합합니다.

●Apache Flink는 강력한 실시간 데이터 처리 엔진으로, 복잡한 데이터 변환과 분석 작업을 실시간으로 수행할 수 있습니다.

장점:

●신뢰성: 데이터 손실 없이 높은 신뢰성으로 데이터를 전송합니다.

●확장성: 대규모 데이터 스트리밍을 처리할 수 있습니다.

●유연성: 다양한 스트리밍 애플리케이션을 쉽게 구축할 수 있습니다.

단점:

●복잡성: 설정과 운영이 복잡할 수 있습니다.

●학습 곡선: 사용법을 익히는 데 시간이 걸릴 수 있습니다.

4. 데이터 저장

선택한 솔루션:

Snowflake

선택 이유:

●Snowflake는 클라우드 기반의 데이터 웨어하우스로, 높은 확장성과 성능을 제공합니다.

●다양한 데이터 소스와 통합이 용이하며, 복잡한 데이터 쿼리를 효율적으로 처리할 수

있습니다.

장점:

- 확장성: 클라우드 기반으로 필요에 따라 자원을 쉽게 확장할 수 있습니다.

- 성능: 고성능 데이터 처리와 쿼리가 가능합니다.

- 관리 용이성: 자동화된 관리 기능을 제공하여 운영 부담이 적습니다.

단점:

- 비용: 대규모 데이터를 처리할 때 비용이 증가할 수 있습니다.

- 의존성: 클라우드 서비스에 대한 의존성이 있습니다.

5. 모니터링 및 알림
선택한 솔루션:

Prometheus와
Grafana

선택 이유:

- Prometheus는 강력한 모니터링과 경고 기능을 제공하며, Grafana와 함께 사용하여 데이터를 시각화할 수 있습니다.

- 다양한 데이터 소스와 쉽게 통합할 수 있으며, 실시간 모니터링이 가능합니다.

장점:

- 실시간 모니터링: 실시간으로 시스템 상태를 모니터링할 수 있습니다.

- 시각화: Grafana를 통해 데이터를 시각화하여 쉽게 이해할 수 있습니다.

- 확장성: 다양한 메트릭을 추가하여 쉽게 확

			<p>장할 수 있습니다.</p> <p>단점:</p> <ul style="list-style-type: none"> ●설정 복잡성: 설정과 운영이 복잡할 수 있습니다. ●학습 곡선: 사용법을 익히는 데 시간이 걸릴 수 있습니다. 	
조항문	10	<p>사례 문제 2번: 금융 기관의 데이터 레이크 구축</p> <p>문제 정의</p> <p>금융 기관은 다양한 소스(거래 시스템, 고객 관계 관리 시스템, 외부 데이터 소스 등)에서 데이터를 수집하여 통합된 데이터 레이크를 구축하려고 합니다. 데이터는 다양한 형식(구조화, 반구조화, 비구조화)으로 존재하며, 실시간 분석과 배치 분석을 모두 지원해야 합니다. 현재 시스템은 여러 데이터 사일로(silo)로 구성되어 있어 데이터 접근성과 분석이 어렵습니다.</p> <p>요구 사항</p> <p>1.데이터 통합: 다양한 소스에서 데이터를 통합하여 데이터 레이크에 저장해야 합니다.</p> <p>2.데이터 정제 및 변환: 수집된 데이터를 정제하고 변환하여 분석 가능하게 만들어야 합</p>	<p>해결 방법</p> <p>1. 데이터 수집 및 통합 선택한 솔루션:</p> <p>Apache Nifi, Kafka Connect, AWS Glue</p> <p>선택 이유:</p> <ul style="list-style-type: none"> ●Apache Nifi는 다양한 소스에서 데이터를 수집하고, 실시간으로 데이터 흐름을 관리할 수 있습니다. ●Kafka Connect는 다양한 소스로부터 데이터를 쉽게 수집하고, Kafka로 데이터를 전송할 수 있는 커넥터를 제공합니다. ●AWS Glue는 서버리스 데이터 통합 서비스로, 다양한 데이터 소스를 연결하고 ETL 작업을 수행할 수 있습니다. <p>장점:</p> <ul style="list-style-type: none"> ●유연성: 다양한 소스에서 데이터를 수집할 수 있습니다. ●확장성: 필요에 따라 쉽게 확장할 수 있습니다. ●자동화: ETL 작업을 자동화하여 효율적으로 관리할 수 있습니다. <p>단점:</p> <ul style="list-style-type: none"> ●설정 복잡성: 다양한 도구를 설정하고 관리 	

니다.

3.실시간 및 배치 처리: 실시간 스트리밍 데이터와 배치 데이터를 모두 처리할 수 있어야 합니다.

4.보안 및 규정 준수: 민감한 금융 데이터를 보호하고 규정을 준수해야 합니다.

5.확장성 및 신뢰성: 대규모 데이터를 처리할 수 있도록 확장 가능하고 신뢰성이 있어야 합니다.

해결해야 할 문제

1.데이터 소스 연결: 다양한 데이터 소스에서 데이터를 수집하기 위해 필요한 기술과 도구를 선택해야 합니다.

2.데이터 정제 및 변환: 수집된 데이터를 정제하고 변환해야 합니다.

3.데이터 저장: 데이터 레이크에 데이터를 저장할 적절한 스토리지 솔루션을 선택해야 합니다.

4.데이터 처리: 실시간 및 배치 데이터를 처리할 수 있는 파이프라인을 구축해야 합니다.

5.보안 및 규정 준수: 데이터 보안과 규정 준수를 보장해야 합니다.

하는 데 시간이 걸릴 수 있습니다.

●비용: AWS Glue와 같은 클라우드 서비스를 사용할 경우 비용이 발생할 수 있습니다.

2. 데이터 정제 및 변환 선택한 솔루션:

Apache Spark, AWS Glue

선택 이유:

●Apache Spark는 대규모 데이터 처리를 위한 강력한 도구로, 다양한 데이터 소스에서 데이터를 읽고, 변환하고, 정제할 수 있습니다.

●AWS Glue는 서버리스 ETL 서비스로, 데이터 정제와 변환을 자동화할 수 있습니다.

장점:

●성능: 인메모리 컴퓨팅을 통해 빠른 데이터 처리가 가능합니다.

●유연성: 다양한 데이터 소스와 통합하여 데이터를 처리할 수 있습니다.

●자동화: ETL 작업을 자동화하여 효율적으로 관리할 수 있습니다.

단점:

●복잡성: 설정과 사용법이 복잡할 수 있습니다.

●자원 사용량: 대규모 데이터를 처리할 때 많은 자원이 필요할 수 있습니다.

3. 데이터 저장 선택한 솔루션: AWS S3, Hadoop HDFS, Google Cloud

기술 스택

- 데이터 수집: Apache Nifi, Kafka Connect, AWS Glue
- 데이터 정제 및 변환: Apache Spark, AWS Glue
- 데이터 저장: AWS S3, Hadoop HDFS, Google Cloud Storage
- 데이터 처리: Apache Spark, Apache Flink, AWS Lambda
- 보안 및 규정 준수: AWS IAM, AWS KMS, Azure Active Directory
- 모니터링 및 알림: Prometheus, Grafana, AWS CloudWatch

Storage

선택 이유:

- AWS S3는 확장성이 뛰어난 클라우드 스토리지 솔루션으로, 대규모 데이터를 저장할 수 있습니다.

- Hadoop HDFS는 분산 파일 시스템으로, 대규모 데이터 저장에 적합합니다.

- Google Cloud Storage는 클라우드 기반의 스토리지 솔루션으로, 높은 확장성과 성능을 제공합니다.

장점:

- 확장성: 대규모 데이터를 저장할 수 있습니다.

- 비용 효율성: 사용한 만큼 비용을 지불하는 유연한 요금제를 제공합니다.

- 통합성: 다양한 분석 도구와 쉽게 통합할 수 있습니다.

단점:

- 의존성: 클라우드 서비스에 대한 의존성이 있습니다.

- 비용: 대규모 데이터를 장기간 저장할 경우 비용이 증가할 수 있습니다.

4. 데이터 처리 선택한 솔루션: Apache Spark, Apache Flink, AWS Lambda

선택 이유:

- Apache Spark는 배치 및 스트리밍 데이터를 모두 처리할 수 있는 강력한 데이터 처리 엔진입니다.

- Apache Flink는 실

시간 스트리밍 데이터 처리를 위한 도구로, 높은 처리량과 낮은 지연 시간을 제공합니다.

●AWS Lambda는 서버리스 컴퓨팅 서비스로, 이벤트 기반 데이터 처리를 자동화할 수 있습니다.

장점:

●유연성: 배치 및 실시간 데이터를 모두 처리할 수 있습니다.

●확장성: 클러스터 환경에서 쉽게 확장할 수 있습니다.

●자동화: 이벤트 기반으로 데이터 처리를 자동화할 수 있습니다.

단점:

●복잡성: 설정과 운영이 복잡할 수 있습니다.

●학습 곡선: 사용법을 익히는 데 시간이 걸릴 수 있습니다.

5. 보안 및 규정 준수
선택한 솔루션: AWS IAM, AWS KMS, Azure Active Directory

선택 이유:

●AWS IAM은 세분화된 권한 관리를 통해 데이터 접근을 제어할 수 있습니다.

●AWS KMS는 데이터 암호화를 통해 민감한 데이터를 보호할 수 있습니다.

●Azure Active Directory는 사용자 인증 및 권한 관리를 위한 강력한 도구입니다.

장점:

●보안성: 민감한 데이터를 보호할 수 있습니다.

●규정 준수: 다양한 보안 규정을 준수할 수 있습니다.

●관리 용이성: 권한 관리와 데이터 보호를 중앙에서 관리할 수 있습니다.

단점:

●복잡성: 설정과 관리가 복잡할 수 있습니다.

●비용: 보안 솔루션을 사용하는 데 비용이 발생할 수 있습니다.

6. 모니터링 및 알림
선택한 솔루션:

Prometheus,
Grafana, AWS
CloudWatch

선택 이유:

●Prometheus는 강력한 모니터링과 경고 기능을 제공하며, Grafana와 함께 사용하여 데이터를 시각화할 수 있습니다.

●AWS CloudWatch는 AWS 리소스를 모니터링하고 로그를 수집할 수 있는 관리형 서비스입니다.

장점:

●실시간 모니터링: 실시간으로 시스템 상태를 모니터링할 수 있습니다.

●시각화: Grafana를 통해 데이터를 시각화하여 쉽게 이해할 수 있습니다.

●통합성: 다양한 서비스와 쉽게 통합할 수 있습니다.

			<p>단점:</p> <ul style="list-style-type: none"> ●설정 복잡성: 설정과 운영이 복잡할 수 있습니다. ●학습 곡선: 사용법을 익히는 데 시간이 걸릴 수 있습니다. <p>이와 같은 이유로 각 단계별 솔루션을 선택하고, 해당 솔루션의 장단점을 고려하여 데이터 레이크를 설계하고 구현합니다.</p>	
조항문	11	<p>사례 문제 3번: 통신사의 데이터 파이프라인 구축 문제 정의</p> <p>통신사는 다양한 소스(콜 로그, 네트워크 트래픽, 고객 데이터 등)에서 데이터를 수집하여 통합된 데이터 파이프라인을 구축하려고 합니다. 데이터는 실시간 분석과 배치 분석을 모두 지원해야 하며, 네트워크 운영, 고객 행동 분석, 마케팅 캠페인 최적화 등에 활용될 것입니다. 현재 시스템은 여러 데이터 사일로(silo)로 구성되어 있어 데이터 접근성과 분석이 어렵습니다.</p> <p>요구 사항</p> <p>1.데이터 통합: 다양한 소스에서 데이터를 통합하여 중앙 데이터 웨어하우스에 저장해야</p>	<p>해결 방법</p> <p>1. 데이터 수집 및 통합 선택한 솔루션: Apache Nifi, Kafka Connect, Flume</p> <p>선택 이유:</p> <ul style="list-style-type: none"> ●Apache Nifi는 데이터 플로우를 시각적으로 관리할 수 있어 다양한 데이터 소스에서 데이터를 쉽게 수집하고 통합할 수 있습니다. ●Kafka Connect는 다양한 소스로부터 데이터를 수집하여 실시간으로 Kafka로 전송할 수 있습니다. ●Flume은 대규모 로그 데이터를 효율적으로 수집하고 전달하는데 유용합니다. <p>장점:</p> <ul style="list-style-type: none"> ●유연성: 다양한 소스에서 데이터를 수집할 수 있습니다. ●확장성: 필요에 따라 쉽게 확장할 수 있습니다. ●시각화: Nifi를 통해 데이터 플로우를 시각 	

합니다.

2.실시간 및 배치 처리: 실시간 스트리밍 데이터와 배치 데이터를 모두 처리할 수 있어야 합니다.

3.확장성: 데이터 파이프라인은 대규모 데이터 처리와 증가하는 데이터 양에 대비해 확장 가능해야 합니다.

4.신뢰성: 데이터 손실 없이 안정적으로 데이터를 처리해야 합니다.

5.보안 및 규정 준수: 민감한 데이터를 보호하고 규정을 준수해야 합니다.

해결해야 할 문제

1.데이터 소스 연결: 다양한 데이터 소스에서 데이터를 수집하기 위해 필요한 기술과 도구를 선택해야 합니다.

2.데이터 정제 및 변환: 수집된 데이터를 정제하고 변환해야 합니다.

3.데이터 저장: 통합된 데이터를 저장할 적절한 데이터 웨어하우스를 선택해야 합니다.

4.데이터 처리: 실시간 및 배치 데이터를 처리할 수 있는 파이프라인을 구축해야 합니다.

5.보안 및 규정 준수

적으로 관리할 수 있습니다.

단점:

●설정 복잡성: 다양한 도구를 설정하고 관리하는 데 시간이 걸릴 수 있습니다.

●자원 사용량: 많은 데이터를 처리할 때 자원 사용량이 증가할 수 있습니다.

2. 데이터 정제 및 변환 선택한 솔루션:

Apache Spark, Apache Flink

선택 이유:

●Apache Spark는 대규모 데이터 처리를 위한 강력한 도구로, 다양한 데이터 소스에서 데이터를 읽고, 변환하고, 정제할 수 있습니다.

●Apache Flink는 실시간 스트리밍 데이터 처리를 위한 도구로, 복잡한 데이터 변환과 분석 작업을 실시간으로 수행할 수 있습니다.

장점:

●성능: 인메모리 컴퓨팅을 통해 빠른 데이터 처리가 가능합니다.

●유연성: 다양한 데이터 소스와 통합하여 데이터를 처리할 수 있습니다.

●확장성: 클러스터 환경에서 쉽게 확장할 수 있습니다.

단점:

●복잡성: 설정과 사용법이 복잡할 수 있습니다.

●자원 사용량: 대규모

수: 데이터 보안과 규정 준수를 보장해야 합니다.

기술 스택

●데이터 수집:

Apache Nifi, Kafka Connect, Flume

●데이터 정제 및

변환: Apache Spark, Apache Flink

●데이터 저장:

Amazon Redshift, Google BigQuery, Snowflake

●데이터 처리:

Apache Spark, Apache Flink, Apache Beam

●보안 및 규정 준수: AWS IAM, AWS KMS, Azure Active Directory

●모니터링 및 알림: Prometheus, Grafana, AWS CloudWatch

데이터를 처리할 때 많은 자원이 필요할 수 있습니다.

3. 데이터 저장 선택한

솔루션: Amazon Redshift, Google BigQuery, Snowflake

선택 이유:

●Amazon Redshift는 대규모 데이터 웨어하우스로, 분석 쿼리를 빠르게 처리할 수 있습니다.

●Google BigQuery는 서버리스 데이터 웨어하우스로, 빠른 쿼리 성능과 유연한 요금제를 제공합니다.

●Snowflake는 클라우드 기반 데이터 웨어하우스로, 높은 확장성과 성능을 제공합니다.

장점:

●확장성: 대규모 데이터를 저장할 수 있습니다.

●성능: 고성능 데이터 처리와 쿼리가 가능합니다.

●유연성: 다양한 분석 도구와 쉽게 통합할 수 있습니다.

단점:

●비용: 대규모 데이터를 장기간 저장할 경우 비용이 증가할 수 있습니다.

●의존성: 클라우드 서비스에 대한 의존성이 있습니다.

4. 데이터 처리 선택한

솔루션: Apache Spark, Apache Flink, Apache Beam

선택 이유:

●Apache Spark는 배치 및 스트리밍 데이터를 모두 처리할 수 있는 강력한 데이터 처리 엔진입니다.

●Apache Flink는 실시간 스트리밍 데이터 처리를 위한 도구로, 높은 처리량과 낮은 지연 시간을 제공합니다.

●Apache Beam은 다양한 실행 엔진을 추상화하여 동일한 파이프라인 코드를 여러 엔진에서 실행할 수 있게 합니다.

장점:

●유연성: 배치 및 실시간 데이터를 모두 처리할 수 있습니다.

●확장성: 클러스터 환경에서 쉽게 확장할 수 있습니다.

●통합성: 여러 실행 엔진과 통합할 수 있습니다.

단점:

●복잡성: 설정과 운영이 복잡할 수 있습니다.

●학습 곡선: 사용법을 익히는 데 시간이 걸릴 수 있습니다.

5. 보안 및 규정 준수
선택한 솔루션: AWS IAM, AWS KMS, Azure Active Directory

선택 이유:

●AWS IAM은 세분화된 권한 관리를 통해 데이터 접근을 제어할 수 있습니다.

●AWS KMS는 데이터 암호화를 통해 민감

한 데이터를 보호할 수 있습니다.

●Azure Active Directory는 사용자 인증 및 권한 관리를 위한 강력한 도구입니다.

장점:

●보안성: 민감한 데이터를 보호할 수 있습니다.

●규정 준수: 다양한 보안 규정을 준수할 수 있습니다.

●관리 용이성: 권한 관리와 데이터 보호를 중앙에서 관리할 수 있습니다.

단점:

●복잡성: 설정과 관리가 복잡할 수 있습니다.

●비용: 보안 솔루션을 사용하는 데 비용이 발생할 수 있습니다.

6. 모니터링 및 알림
선택한 솔루션:

Prometheus,
Grafana, AWS
CloudWatch

선택 이유:

●Prometheus는 강력한 모니터링과 경고 기능을 제공하며, Grafana와 함께 사용하여 데이터를 시각화할 수 있습니다.

●AWS CloudWatch는 AWS 리소스를 모니터링하고 로그를 수집할 수 있는 관리형 서비스입니다.

장점:

●실시간 모니터링: 실시간으로 시스템 상태를 모니터링할 수 있습니다.

			<p>니다.</p> <ul style="list-style-type: none"> ●시각화: Grafana를 통해 데이터를 시각화하여 쉽게 이해할 수 있습니다. ●통합성: 다양한 서비스와 쉽게 통합할 수 있습니다. <p>단점:</p> <ul style="list-style-type: none"> ●설정 복잡성: 설정과 운영이 복잡할 수 있습니다. ●학습 곡선: 사용법을 익히는 데 시간이 걸릴 수 있습니다. 	
김소라	상황제시	<p>OTT 회사에서 실시간 스트리밍 데이터 파이프라인을 구축하여 실시간 추천 시스템을 구축하려고 합니다. 사용자 행동 데이터를 실시간으로 수집, 처리, 저장, 분석하여 사용자에게 맞춤형 콘텐츠를 추천하는 것을 목표로 합니다.</p> <p>회사의 요구사항은 아래와 같습니다.</p> <ol style="list-style-type: none"> 1. 실시간 데이터 수집: 사용자 행동 데이터를 실시간으로 수집 (예: 재생 시작/중지, 시청 완료, 평가, 검색 기록). 2. 데이터 처리 및 변환: 수집된 데이터를 실시간으로 정제, 변환 및 사용자 프로필 업데이트. 3. 데이터 저장: 실 		

		<p>시간 처리된 데이터를 안정적이고 확장 가능한 저장소에 저장.</p> <p>4. 추천 알고리즘 적용: 실시간으로 수집된 데이터를 바탕으로 추천 알고리즘 실행.</p> <p>5. 실시간 추천 제공: 사용자 인터페이스에 실시간으로 추천 결과 제공.</p> <p>6. 확장성: 시스템이 수백만 명의 동시 사용자를 처리할 수 있도록 확장 가능해야 함.</p> <p>7. 고가용성: 시스템이 높은 가용성을 유지하고 장애 시 빠르게 복구될 수 있어야 함.</p> <p>8. 보안: 데이터 전송 및 저장 과정에서 데이터 보안을 유지해야 함.</p>		
김소라	12	<p>이 회사에서 실시간 추천시스템을 구축할 때, 데이터 파이프라인을 흐름대로 시나리오를 기술 하시오.</p> <p>* 기술을 고려하지 않고, 비즈니스 요구사항에 맞춰 파이프라인 기술해주세요.</p>	<p>##1. 데이터 수집 사용자 행동 이벤트 수집</p> <p>1. 사용자가 넷플릭스에서 동영상을 재생하거나 검색할 때마다 해당 이벤트가 로그로 기록됩니다.</p> <p>2. Logstash는 이 로그 데이터를 실시간으로 수집하여 Apache Kafka로 전송합니다.</p> <p>##2. 데이터 처리 실시간 데이터 정제 및 변환</p> <p>1. Apache Kafka는 수집된 이벤트 데이터를 토픽(topic)에 저장</p>	

합니다.

2. Apache Flink 또는 Apache Spark Streaming이 Kafka에서 데이터를 읽어와 실시간으로 처리합니다.

- 이벤트 데이터를 정제하여 필요한 필드만 추출하고, 사용자별 시청 기록 및 선호도 데이터를 업데이트합니다.

3. 데이터 저장
정제된 데이터 저장소
1. 정제된 데이터는 실시간으로 Amazon S3에 저장되어 데이터 레이크를 구성합니다.

2. 동시에, 주요 분석을 위한 데이터는 Apache Cassandra에 저장됩니다.

3. 사용자 프로필 및 추천 데이터는 빠른 액세스를 위해 Redis에 저장됩니다.

4. 추천 알고리즘 적용

추천 알고리즘 실행

1. Apache Flink 또는 Apache Spark Streaming에서 실시간으로 수집된 데이터를 기반으로 추천 알고리즘을 실행합니다.

- Apache Mahout을 사용하여 사용자 선호도와 시청 기록을 기반으로 추천 목록을 생성합니다.

- TensorFlow Serving을 통해 머신러닝 모델을 실시간으로 서빙하고, 더 정교한 추천을 제공합니다.

5. 실시간 추천 제공

추천 결과 제공

1. 사용자가 넷플릭스 인터페이스에 접속하면 Redis에서 실시간으로 업데이트된 추천 목록을 가져옵니다.
2. 추천 결과는 사용자에게 실시간으로 제공됩니다.

6. 실시간 분석 및 모니터링

시스템 모니터링 및 분석

1. Elastic Stack을 사용하여 실시간 로그 데이터를 수집하고 분석합니다.

- Logstash가 Kafka에서 데이터를 읽어와 Elasticsearch에 저장합니다.

- Kibana를 통해 실시간 대시보드에서 사용자 행동 데이터를 시각화합니다.

2. Grafana를 사용하여 시스템 상태 및 성능을 모니터링합니다.

- 예를 들어, 서버의 CPU 사용량, 메모리 사용량, 네트워크 트래픽 등을 모니터링하여 실시간으로 시각화합니다.

7. 보안 및 인증

데이터 보안 관리

1. Apache Ranger를 사용하여 데이터 접근 제어 정책을 관리합니다.

2. Kerberos를 통해 각 서비스 간의 인증을 수행하여 데이터 전송의 보안을 강화합니다.

결론

이 실시간 스트리밍 데이터 파이프라인을 통해 넷플릭스는 사용자 행동 데이터를 실시간으로 수집하고 처리하여 사용자에게 맞춤형 콘텐츠를 실시간으로 추천할 수 있습니다. 이 기술 스택을 통해 OTT와 같은 회사는 실시간 추천 시스템을 구축하여 사용자 경험을 향상시키고, 데이터 처리 및 분석의 효율성을 높일 수 있습니다. 각 기술의 장단점을 고려하여 적절히 조합함으로써 시스템의 확장성, 고가용성, 보안을 모두 충족할 수 있습니다.

- 사용자 경험 향상: 사용자에게 개인화된 추천 콘텐츠를 실시간으로 제공하여 만족도를 높입니다.
- 효율적인 데이터 처리: 실시간 데이터 수집 및 처리로 데이터 활용의 효율성을 극대화합니다.
- 확장성: 대규모 데이터와 동시 사용자를 처리할 수 있도록 시스템을 확장 가능하게 만듭니다.
- 고가용성: 시스템 장애 시에도 빠르게 복구할 수 있는 고가용성을 보장합니다.
- 보안: 데이터 전송 및 저장 과정에서 강력한 보안을 유지하여 사용

			자 데이터를 보호합니다.	
김소라	13	문제 1에 맞춰, 필요 기술들을 도입하여, 시스템을 구축하고, 해당 시스템 구축 효과에 대해서도 기술하세요.		
임성민	14	<p>데이터 엔지니어링과 관련된 기술사 형태의 샘플 문제는 여러 가지 주제를 포함할 수 있습니다. 아래는 데이터 엔지니어링의 핵심 개념과 기술을 다루는 몇 가지 예시 문제입니다.</p> <p>문제 1: 데이터 웨어하우스 설계 회사는 여러 소스 시스템으로부터 데이터를 수집하여 분석하고자 합니다. 데이터 웨어하우스를 설계하려고 할 때, 다음 질문에 답하세요.</p> <p>데이터 모델링: 데이터 웨어하우스를 설계하기 위해 어떤 데이터 모델링 기법을 사용할 것인지 설명하고, 그 이유를 논의하세요.</p> <p>ETL 프로세스: ETL(Extract, Transform, Load) 프로세스를 설계하는 단계와 주요 고려 사항을</p>	<p>문제 1: 데이터 웨어하우스 설계 데이터 모델링:</p> <p>사용할 기법: 스타 스키마(Star Schema)와 스노우플레이크 스키마(Snowflake Schema) 중 스타 스키마를 사용합니다. 이유: 스타 스키마는 단순하고 이해하기 쉬워 데이터 분석과 쿼리 성능이 뛰어납니다. 사실 테이블(fact table)과 차원 테이블(dimension table)로 구성되어 있어 데이터를 효율적으로 집계할 수 있습니다.</p> <p>ETL 프로세스:</p> <p>단계:</p> <p>추출(Extract): 소스 시스템으로부터 데이터를 추출합니다. 변환(Transform): 데이터를 일관성 있게 변환하고 클렌징합니다. 적재(Load): 변환된 데이터를 데이터 웨어하우스에 적재합니다.</p> <p>고려 사항:</p> <p>데이터 품질: 데이터 정확성과 일관성을 유지해야 합니다.</p>	

설명하세요.

성능 최적화: 데이터 웨어하우스의 성능을 최적화하기 위해 사용할 수 있는 방법들을 제시하세요.

문제 2: 빅데이터 처리

회사는 소셜 미디어 플랫폼으로부터 실시간 데이터를 수집하고 분석하여 고객 감정을 파악하려고 합니다. 이를 위해 빅데이터 처리 시스템을 설계해야 합니다.

데이터 수집: 실시간 데이터 스트리밍을 위한 데이터 수집 기술을 설명하고, 해당 기술의 장단점을 논의하세요.

데이터 저장: 수집된 데이터를 저장하기 위한 NoSQL 데이터베이스를 선택하고, 선택한 데이터베이스의 장점과 단점을 설명하세요.

데이터 처리: 실시간 데이터 처리를 위한 분산 처리 프레임워크를 선택하고, 해당 프레임워크의 주요 기능과 사용 사례를 설명하세요.

문제 3: 데이터 통합 및 품질 관리
여러 부서에서 서로 다른 형식의 데

성능: 대용량 데이터를 빠르게 처리할 수 있는 효율적인 프로세스를 설계해야 합니다.

오류 처리: 오류 발생 시 대처할 수 있는 메커니즘을 마련해야 합니다.

성능 최적화:

인덱싱: 적절한 인덱스를 생성하여 쿼리 성능을 향상시킵니다.

파티셔닝: 데이터를 파티션으로 분할하여 관리와 쿼리 성능을 개선합니다.

물리적 설계: 정규화와 비정규화를 적절히 사용하여 데이터 모델을 최적화합니다.

캐싱: 자주 조회되는 데이터를 캐시에 저장하여 쿼리 응답 시간을 줄입니다.

문제 2: 빅데이터 처리 데이터 수집:

기술: Apache Kafka

장단점:

장점: 높은 처리량, 확장성, 내결함성이 뛰어납니다.

단점: 초기 설정과 운영이 복잡하며, 학습 곡선이 가파릅니다.

데이터 저장:

NoSQL 데이터베이스:

Apache Cassandra

장점: 높은 쓰기 처리 성능, 분산형 구조로 확장성이 좋습니다.

단점: 복잡한 쿼리 지원이 부족하고, 데이터

이터를 사용하는 회사에서, 데이터 통합과 품질 관리 시스템을 구축하려고 합니다.

데이터 통합: 서로 다른 형식의 데이터를 통합하기 위한 방법론을 설명하고, 각 방법론의 장단점을 논의하세요.

데이터 품질 관리: 데이터 품질을 관리하기 위한 주요 지표와 이를 개선하기 위한 전략을 제시하세요.

메타데이터 관리: 메타데이터 관리의 중요성을 설명하고, 효과적인 메타데이터 관리 시스템을 설계하는 방법을 논의하세요.
문제 4: 클라우드 기반 데이터 엔지니어링
회사는 클라우드 환경에서 데이터 엔지니어링 파이프라인을 구축하려고 합니다.

클라우드 서비스: 데이터 엔지니어링을 위해 사용할 수 있는 주요 클라우드 서비스들을 비교하고, 각 서비스의 장단점을 설명하세요.

데이터 보안: 클라우드 환경에서 데이터를 안전하게

모델링이 어렵습니다.
데이터 처리:

프레임워크: Apache Spark

주요 기능: 대용량 데이터 처리, 실시간 스트리밍 분석, 다양한 언어 지원(Python, Scala, Java)
사용 사례: 로그 분석, 실시간 데이터 스트리밍 처리, 기계 학습 모델 훈련

문제 3: 데이터 통합 및 품질 관리
데이터 통합:

방법론: 데이터 레이크(Data Lake)

장점: 다양한 데이터 형식을 지원하며, 원본 데이터를 그대로 저장할 수 있습니다.

단점: 데이터 품질 관리가 어렵고, 잘못된 데이터가 혼재할 수 있습니다.

데이터 품질 관리:

주요 지표: 정확성(Accuracy), 일관성(Consistency), 완전성(Completeness), 적시성(Timeliness)

개선 전략:

데이터 프로파일링: 데이터의 특성을 파악하고, 문제점을 발견합니다.

데이터 클렌징: 오류 데이터를 수정하거나 제거합니다.

데이터 거버넌스: 데이터 관리 정책과 절차를 수립하여 품질을 유지

보호하기 위한 보안 전략을 제시하세요.

비용 관리: 클라우드 기반 데이터 엔지니어링 파이프라인의 비용을 효율적으로 관리하기 위한 방안을 논의하세요.

문제 5: 데이터 분석과 시각화
회사는 데이터 엔지니어링 파이프라인을 통해 수집된 데이터를 분석하고 시각화하여 경영진에게 보고하려고 합니다.

데이터 분석: 데이터 분석을 위한 주요 기법과 도구를 설명하고, 각각의 사용 사례를 논의하세요.

데이터 시각화: 데이터 시각화를 위해 사용할 수 있는 도구와 기술을 비교하고, 각각의 장단점을 설명하세요.

보고서 작성: 경영진에게 효과적으로 데이터를 전달하기 위한 보고서 작성의 원칙과 전략을 제시하세요.

이 샘플 문제들은 데이터 엔지니어링의 다양한 측면을 다루며, 실제 상황에서 발생할 수 있는 문제들을 해결하기 위한 전략과

합니다.

메타데이터 관리:

중요성: 메타데이터 관리의 데이터의 출처, 구조, 의미를 이해하고, 데이터 사용성을 높이는 데 중요합니다.

설계 방법:

중앙 메타데이터 저장소 구축: 모든 메타데이터를 중앙에서 관리합니다.

자동화 도구 사용: 메타데이터 수집과 갱신을 자동화하여 일관성을 유지합니다.

사용자 접근성 향상: 사용자가 메타데이터에 쉽게 접근할 수 있도록 인터페이스를 제공합니다.

문제 4: 클라우드 기반 데이터 엔지니어링 클라우드 서비스:

비교: AWS Redshift, Google BigQuery, Azure Synapse Analytics

장단점:

AWS Redshift: 높은 성능과 확장성, 다양한 통합 도구 지원. 비용이 상대적으로 높음.

Google BigQuery: 서버리스 아키텍처로 관리가 용이, 빠른 쿼리 성능. 복잡한 쿼리 비용이 높음.

Azure Synapse Analytics: 데이터 통합 및 분석 기능 통합, 하이브리드 데이터 처리. 초기 설정이 복잡할 수 있음.

기술을 이해하고
적용하는 데 도움
을 줄 것입니다.

데이터 보안:

전략:

데이터 암호화: 저장
데이터와 전송 데이터
모두 암호화합니다.

접근 제어: 역할 기반
접근 제어(RBAC)와
다단계 인증(MFA)을
사용합니다.

네트워크 보안: 가상
사설망(VPN)과 방화
벽을 사용하여 네트워
크를 보호합니다.

비용 관리:

방안:

비용 예측 도구 사용:
클라우드 제공사의 비
용 관리 도구를 사용하
여 비용을 예측하고 추
적합니다.

자동 스케일링: 필요할
때만 리소스를 사용하
도록 자동 스케일링을
설정합니다.

예약 인스턴스 사용:
장기적으로 사용할 리
소스는 예약 인스턴스
를 사용하여 비용을 절
감합니다.

문제 5: 데이터 분석과
시각화

데이터 분석:

기법:

회귀 분석: 변수 간의
관계를 분석하여 예측
모델을 만듭니다.

군집 분석: 데이터 포
인트를 유사한 그룹으
로 묶어 패턴을 발견합
니다.

시계열 분석: 시간에
따른 데이터 변화를 분
석하여 트렌드를 예측

합니다.

도구:

Python (pandas,
scikit-learn)

R

SQL

데이터 시각화:

도구:

Tableau: 사용하기 쉽
고, 다양한 데이터 소
스와 연결 가능. 비용
이 높음.

Power BI: Microsoft
생태계와의 통합이 우
수. 사용 시 학습 필요.

Matplotlib/Seaborn
(Python): 커스터마이
징이 용이. 시각적 디
자인이 제한적일 수 있
음.

장단점:

Tableau: 직관적인 인
터페이스, 고급 시각화
가능. 비용이 높음.

Power BI: 뛰어난 통
합 기능, 사용 편의성.
복잡한 분석에 한계.

Matplotlib/Seaborn:
코드 기반으로 유연성
높음. 시각화 디자인이
기본적.

보고서 작성:

원칙:

명확성과 간결성: 정보
를 명확하고 간결하게
전달합니다.

시각적 효과: 적절한
시각화를 사용하여 데
이터의 의미를 쉽게 이
해할 수 있도록 합니
다.

목적에 맞는 내용: 경
영진이 필요로 하는 정
보를 중심으로 구성함

			<p>니다.</p> <p>전략:</p> <p>스토리텔링 기법 활용: 데이터를 이야기 형태로 구성하여 이해를 돕습니다.</p> <p>주요 지표 강조: 중요한 지표와 인사이트를 강조하여 주의를 끄니다.</p> <p>피드백 반영: 경영진의 피드백을 반영하여 보고서를 개선합니다.</p>	
임성민	15	<p>K Software가 전 세계 판매를 목적으로 클라우드 환경에 게임 판매 사이트를 구축하려고 한다면,</p> <p>다음과 같은 데이터 엔지니어링 솔루션을 설계할 수 있습니다.</p> <p>이 솔루션은 사용자 성향 및 SNS 분석을 통해 게임 추천 기능을 제공하고, 구매 유도를 목표로 합니다.</p>	<p>문제 1: 클라우드 환경에서의 데이터 엔지니어링 솔루션 설계</p> <p>데이터 모델링</p> <p>사용할 데이터 모델링 기법:</p> <p>**스타 스키마(Star Schema)**를 사용합니다.</p> <p>이유: 스타 스키마는 데이터 웨어하우스 설계에 적합하며, 간단하고 효율적으로 데이터를 집계할 수 있습니다. 특히, 게임 판매 데이터와 사용자 활동 데이터를 분석하는 데 적합합니다.</p> <p>ETL 프로세스</p> <p>ETL 프로세스 설계:</p> <p>추출(Extract): 사용자 활동, 게임 다운로드, 구매 내역, SNS 데이터 등 다양한 소스 시스템에서 데이터를 추출합니다.</p> <p>변환(Transform): 데이터 클렌징, 일관성 유지, 사용자 성향 분석을 위한 데이터 변환을 수행합니다.</p> <p>적재(Load): 변환된</p>	

데이터를 클라우드 데이터 웨어하우스 (AWS Redshift, Google BigQuery 등)에 적재합니다.

고려 사항:

데이터의 일관성 및 정확성 확보

실시간 데이터 처리 요구 사항 반영

확장성과 유연성을 고려한 설계

성능 최적화

데이터 웨어하우스 성능 최적화:

인덱싱: 자주 조회되는 컬럼에 대해 인덱스를 생성하여 쿼리 성능을 향상시킵니다.

파티셔닝: 데이터를 날짜별, 사용자별 등으로 파티셔닝하여 쿼리 성능을 최적화합니다.

캐싱: 자주 조회되는 데이터를 캐시에 저장하여 응답 시간을 단축시킵니다.

물리적 설계: 비정규화된 테이블을 사용하여 읽기 성능을 향상시킵니다.

문제 2: 실시간 빅데이터 처리 시스템 설계

데이터 수집

실시간 데이터 수집 기술:

Apache Kafka를 사용합니다.

장단점:

장점: 높은 처리량, 확장성, 내결함성 제공

단점: 초기 설정과 운영의 복잡성, 학습 곡선이 가파름

데이터 저장

NoSQL 데이터베이스
선택:

Apache Cassandra
를 사용합니다.

장점: 높은 쓰기 성능,
분산 구조로 높은 확장
성

단점: 복잡한 쿼리 지
원 부족, 데이터 모델
링이 어렵습니다.

데이터 처리

실시간 데이터 처리 프
레이밍워크:

Apache Spark를 사
용합니다.

주요 기능: 대용량 데
이터 처리, 실시간 스트리밍 분석, 다양한
언어 지원(Python,
Scala, Java)

사용 사례: 실시간 사
용자 활동 분석, 실
시간 게임 추천 시스템

문제 3: 데이터 통합
및 품질 관리

데이터 통합

데이터 통합 방법론:

****데이터 레이크**

(Data Lake)**를 사
용합니다.

장점: 다양한 데이터
형식 지원, 원본 데이
터를 그대로 저장 가능

단점: 데이터 품질 관
리가 어려움, 잘못된
데이터 혼재 가능

데이터 품질 관리

데이터 품질 관리 지표
및 전략:

주요 지표: 정확성

(Accuracy), 일관성

(Consistency), 완전

성(Completeness),

적시성(Timeliness)

개선 전략:

데이터 프로파일링: 데이터의 특성을 파악하고 문제점 발견
 데이터 클렌징: 오류 데이터를 수정하거나 제거
 데이터 거버넌스: 데이터 관리 정책과 절차 수립 및 실행
 메타데이터 관리
 메타데이터 관리 시스템 설계:
 중요성: 데이터의 출처, 구조, 의미를 이해하고, 데이터 사용성을 높이는 데 중요
 설계 방법:
 중앙 메타데이터 저장소 구축: 모든 메타데이터를 중앙에서 관리
 자동화 도구 사용: 메타데이터 수집과 갱신을 자동화하여 일관성 유지
 사용자 접근성 향상: 사용자가 메타데이터에 쉽게 접근할 수 있도록 인터페이스 제공

문제 4: 클라우드 기반 데이터 엔지니어링
 클라우드 서비스
 클라우드 서비스 비교 및 선택:
 AWS Redshift,
 Google BigQuery,
 Azure Synapse Analytics 비교
 장단점:
 AWS Redshift: 높은 성능, 다양한 통합 도구 지원, 비용이 상대적으로 높음
 Google BigQuery: 서버리스 아키텍처로 관리 용이, 빠른 쿼리

성능, 복잡한 쿼리 비용 높음

Azure Synapse

Analytics: 데이터 통합 및 분석 기능 통합, 하이브리드 데이터 처리, 초기 설정 복잡
데이터 보안

클라우드 데이터 보안 전략:

데이터 암호화: 저장 데이터와 전송 데이터를 모두 암호화

접근 제어: 역할 기반 접근 제어(RBAC) 및 다단계 인증(MFA) 사용

네트워크 보안: 가상 사설망(VPN) 및 방화벽 사용

비용 관리

클라우드 비용 관리 방안:

비용 예측 도구 사용: 클라우드 제공사의 비용 관리 도구 사용
자동 스케일링: 필요할 때만 리소스를 사용하여 자동 스케일링 설정

예약 인스턴스 사용: 장기적으로 사용할 리소스는 예약 인스턴스 사용

문제 5: 데이터 분석과 시각화

데이터 분석

데이터 분석 기법과 도구:

기법:

회귀 분석: 변수 간 관계 분석 및 예측 모델 생성

군집 분석: 데이터 포인트를 유사한 그룹으로

로 묶어 패턴 발견
시계열 분석: 시간에
따른 데이터 변화 분석
및 트렌드 예측
도구:
Python (pandas,
scikit-learn)
R
SQL
데이터 시각화
데이터 시각화 도구 및
기술:
도구:
Tableau: 사용하기 쉽
고 다양한 데이터 소스
와 연결 가능, 비용이
높음
Power BI: Microsoft
생태계와 통합 우수,
사용 시 학습 필요
Matplotlib/Seaborn
(Python): 커스터마이
징 용이, 시각적 디자
인 제한적
장단점:
Tableau: 직관적 인터
페이스, 고급 시각화
가능, 비용 높음
Power BI: 뛰어난 통
합 기능, 사용 편의성,
복잡한 분석에 한계
Matplotlib/Seaborn:
코드 기반으로 유연성
높음, 시각화 디자인
기본적
보고서 작성
보고서 작성 원칙 및
전략:
원칙:
명확성과 간결성: 정보
를 명확하고 간결하게
전달
시각적 효과: 적절한
시각화 사용으로 데이
터의 의미 쉽게 이해
가능

			<p>목적에 맞는 내용: 경영진이 필요로 하는 정보 중심으로 구성</p> <p>전략:</p> <p>스토리텔링 기법 활용: 데이터를 이야기 형태로 구성하여 이해 도움</p> <p>주요 지표 강조: 중요한 지표와 인사이트 강조하여 주의 집중</p> <p>피드백 반영: 경영진의 피드백 반영하여 보고서 개선</p> <p>위의 솔루션을 통해 K Software는 전 세계적으로 확장 가능한 클라우드 기반 게임 판매 사이트를 구축하고, 실시간 데이터 분석 및 사용자 성향 분석을 통해 맞춤형 게임 추천 시스템을 구현할 수 있습니다.</p>	
최인식	상황제시	<p>문제 제목 : 통신 부정이용 차단 시스템 구축</p> <p>[고객 상황]</p> <p>고객 상황 : 부정이용차단은 휴대전화 명의도용 및 자금 융통 목적의 단말기 폰깡과 보이스 피싱등으로 부터 고객 보호 목적으로 시행하고 있으며 부정이용자들의 통신사용패턴이 다양하게 변화되어가는 수법을 차단하기 위해서 노력하고 있습니다.</p> <p>Pain Point : 2개의 시스템에서 데</p>		

이터를 수작업으로
추출하여 100개
이상의 차단패턴을
수작업으로 분류하
고 있어서 소규모
팀에서 업무가 과
중되어있는 상황

업무 요건 :

- 기존 온프레미스
기반의 Legacy
시스템의 데이터를
통합할 수 있는 데
이터마트 구성
- 신규 독자적 서
버 구축 필요
- Rule 기반의 패
턴 감지를 머신러
닝을 통한 패턴 예
측 및 이상징후 탐
지 시스템 개발
- 기존 Legacy 시
스템 중 A 시스템
은 실시간 연동 필
요
- WEB 상에서 패
턴결과에 대한 데
이터 관리가 가능
하도록 WebApp
개발

성공 지표 :

- 부정이용 대상
차단율
- 고객 수납율

[업무 시나리오 1]

1. 실시간 및 배치
기반의 수집 처리 :
데이터성격에 따른
실시간 및 배치로
수집
2. 수집된 데이터
에 대한 통합 및 적
재 처리
3. 적재된 데이터

		<p>소스로 Rule 기반의 패턴분류 및 모니터링 시스템 구축</p> <p>4. 사업부서는 구축된 모니터링 시스템을 통해서 결제 차단</p> <p>5. 차단된 내용에 대해서 과거 1년치 까지 관리</p> <p>6. 사업부서 이외에 다른 사용자가 접근할 수 없도록 보안시스템 구축</p> <p>7. Rule 기반과 병행할 수 있는 AI 머신러닝으로 구축</p>		
최인식	16	<p>업무시나리오 1번에서 Legacy 시스템에서 실시간 및 배치로 수집 및 대시보드까지 Data Pipe Line을 설계하세요.</p>		
최인식	17	<p>FDS 시스템을 Container 기반으로 설계하세요.</p>		
김재환	18	<p>문제1: 데이터 품질 관리와 비즈니스 인텔리전스를 위한 데이터 아키텍처 설계</p> <p>[고객 상황] 미디어 스트리밍 서비스</p> <p>'PlayStream'은 사용자 활동 데이터와 콘텐츠 소비 데이터를 활용하여 데이터 기반 비즈니스 인텔리전스 (BI) 시스템을 구축하고자 합니다.</p>		

업무 요건:

- 실시간 사용자 활동 데이터와 콘텐츠 소비 데이터를 수집
- 대규모 데이터 저장 및 관리
- 데이터 품질 문제 해결 및 데이터 정제
- 사용자 행동 패턴, 선호 콘텐츠 분석
- 보고서 생성 및 비즈니스 인사이트 도출

질문:

- 1.실시간 사용자 활동 데이터와 콘텐츠 소비 데이터를 수집하기 위한 데이터 파이프라인을 설계하세요. 두 가지 안을 제시하고 각 선택의 이유를 설명하세요.
- 2.데이터 품질 관리 및 정제를 위해 필수적인 단계와 기술을 제시하세요.
- 3.사용자 행동 패턴과 선호 콘텐츠 분석을 위한 데이터 모델과 분석 방식을 설계하세요.
- 4.비즈니스 인사이트를 도출하기 위한 데이터 시각화 도구와 방법을 제안하세요.
- 5.해당 아키텍처가 데이터 보안을 강화하기 위한 방법을 설명하세요.

김재환	19	<p>문제2: IoT 데이터 분석 및 예방적 유지 보수를 위한 데이터 아키텍처 설계</p> <p>[고객 상황] 산업 장비 제조업체 'Industech'는 IoT 장비로부터 발생하는 데이터를 활용하여 장비의 예방적 유지 보수를 실시하고자 합니다.</p> <p>업무 요건:</p> <ul style="list-style-type: none"> ●IoT 센서 데이터 실시간 수집과 처리 ●대규모 IoT 데이터 저장 ●데이터 정제 및 품질 관리 ●장비 이상 탐지 및 예측 모델 개발 ●예방적 유지 보수 스케줄링 ●실시간 알림 시스템 구축 <p>질문:</p> <p>1.IoT 장비로부터 발생하는 실시간 데이터를 수집하고 처리하기 위한 데이터 파이프라인을 설계하세요. 두 가지 안을 제시하고 각 선택의 이유를 설명하세요.</p> <p>2.대규모 IoT 데이터를 저장하고 관리하기 위한 데이터베이스 아키텍처를 설계하세요.</p> <p>3.데이터 정제 및 품질 관리를 위해</p>		
-----	----	--	--	--

		<p>필요한 단계와 자동화 기술을 제시하세요.</p> <p>4.장비 이상 탐지와 예측 모델을 위한 머신러닝 워크플로우를 설계하세요.</p> <p>5.실시간 알림 시스템을 구축하기 위한 아키텍처와 기술을 제안하세요.</p>		
이유진	상황제시	<p>문제 제목 : 실손보험료 자동 청구 시스템 구축</p> <p>상황 설명</p> <p>고객의 동의 하에 국민건강보험공단의 개인별 의료정보를 보험기관에 연계하여 급여/비급여 의료비에 대한 실손보험 청구료를 계산하여 자동 지급될 수 있도록 시스템 구축</p> <ul style="list-style-type: none"> - 각 보험사에서 요구하는 보험료 산정을 위한 의료정보 표준모델 구축 - 고객개인정보 연동에 따른 개인정보 암호화 시스템 구축 - 개인정보 전송에 따른 전송이력 관리 - 의료정보 전송기 		

		록 관리를 위한 플랫폼 구축		
이유진	20	개인의 의료정보에 대한 데이터 표준 모델을 구축할 때 고려해야 할 사항을 기술하십시오.		
이유진	21	개인의 의료기록을 수집하여 각 보험사로 제공하기 위해 사용해야 할 데이터 파이프라인을 설계하고 이를 관리할 수 있는 모니터링 방안을 제시하십시오.		

이름	No.	질문	답안	주요 키워드
박성근	1	해외 지사를 두고 있는 글로벌 기업에서 각 지사에서 발생하는 데이터를 통합하여 데이터레이크를 구성하려고 할 때 IT 관점에서 고려해야 되는 사항은 무엇인가요?	<p>1.데이터 거버넌스 (Data Governance):</p> <p>○정책 및 규제 준수: 각 국가 및 지역의 데이터 보호 법규 (예: GDPR, CCPA)를 준수하기 위한 정책을 수립하고 준수해야 합니다.</p> <p>○데이터 품질 관리: 일관된 데이터 품질을 유지하기 위해 데이터 정제, 데이터 중복 제거 및 데이터 표준화를 실시해야 합니다.</p> <p>2.데이터 보안 (Data Security):</p> <p>○암호화: 데이터 저장 시와 전송 시의 암호화를 적용하여 데이터 보안을 강화해야 합니다.</p> <p>○액세스 제어: 민감 데이터에 대한 접근 권한을 제어하고, 역할 기반 접근 제어 (RBAC)를 구현해야 합니다.</p>	데이터레이크 구성방안

○침입 탐지 및 대응:
실시간으로 데이터 접근을 모니터링하고, 잠재적인 침입 시도를 탐지하여 대응할 수 있는 시스템을 갖추어야 합니다.

3.데이터 인프라 (Data Infrastructure):

○클라우드 기반 데이터 레이크: 글로벌 스케일의 데이터 저장 및 처리를 위해 클라우드 기반 솔루션 (예: AWS S3, Azure Data Lake, Google Cloud Storage)을 고려해야 합니다.

○데이터 전송: 지사에서 중앙 데이터 레이크로 데이터를 안전하고 효율적으로 전송할 수 있는 네트워크 인프라를 구축해야 합니다.

○지연 최소화: 데이터 전송 시의 지연을 최소화하고, 전송 속도를 최적화할 수 있는 방법을 찾아야 합니다.

4.데이터 통합 (Data Integration):

○ETL 프로세스: 데이터를 추출(Extract), 변환(Transform), 적재(Load)하는 ETL 파이프라인을 설계하고 자동화해야 합니다.

○실시간 데이터 스트리밍: 실시간 데이터를 처리하기 위한 스트리밍 플랫폼 (예: Apache Kafka)을 구축하여 실시간 데이터 통합을 지원해야 합니다.

5.데이터 카탈로그 및
메타데이터 관리

(Data Catalog and
Metadata
Management):

○데이터 카탈로그: 모
든 데이터를 쉽게 검색
하고 활용할 수 있도록
데이터 카탈로그를 구
축하여 메타데이터를
관리해야 합니다.

○데이터 계보 (Data
Lineage): 데이터의
출처, 이동 경로 및 변
환 내역을 추적할 수
있는 시스템을 도입하
여 데이터 계보를 관리
해야 합니다.

6.성능 및 확장성
(Performance and
Scalability):

○확장성: 데이터 양이
증가함에 따라 쉽게 확
장할 수 있는 인프라를
설계해야 합니다.

○성능 최적화: 데이터
처리 및 쿼리 성능을
최적화하기 위해 적절
한 인덱싱, 파티셔닝
및 캐싱 전략을 적용해
야 합니다.

7.백업 및 복구
(Backup and
Recovery):

○정기 백업: 데이터의
정기적인 백업을 통해
데이터 손실에 대비해
야 합니다.

○재해 복구: 재해 발
생 시 신속하게 데이터
복구가 가능하도록 재
해 복구 계획을 수립하
고 테스트해야 합니다.

8.데이터 분석 및 활용
(Data Analytics and
Utilization):

			<p>○분석 플랫폼 통합: 데이터 레이크와 분석 플랫폼 (예: Apache Spark, Presto, Apache Flink)을 통합하여 데이터 분석을 지원해야 합니다.</p> <p>○머신러닝 및 AI: 데이터 레이크 상에서 머신러닝 및 AI 모델을 구축하고 운영할 수 있는 환경을 마련해야 합니다.</p>	
박성근	2	빅데이터 실시간 처리 데이터 아키텍처 수립 시 고려해야 되는 사항은 무엇인가요?	<p>1.데이터 수집 (Data Ingestion):</p> <p>○다양한 데이터 소스 지원: 여러 소스(예: IoT 디바이스, 웹 로그, 소셜 미디어 등)에서 데이터를 실시간으로 수집할 수 있는 인프라를 구축해야 합니다.</p> <p>○스트리밍 플랫폼: Apache Kafka, Apache Pulsar, Amazon Kinesis와 같은 스트리밍 플랫폼을 사용하여 높은 처리량과 저지연 데이터 수집을 보장해야 합니다.</p> <p>2.데이터 처리 (Data Processing):</p> <p>○스트림 처리 엔진: Apache Flink, Apache Spark Streaming, Apache Storm과 같은 스트림 처리 엔진을 사용하여 실시간 데이터 처리를 구현해야 합니다.</p> <p>○저지연 처리: 데이터를 신속하게 처리하기 위해 저지연 처리 프레임워크와 기술을 선택해야 합니다.</p>	실시간 처리

○상태 관리: 실시간 데이터 처리 중 상태를 유지하고 관리할 수 있는 방법을 고려해야 합니다.

3.데이터 저장 (Data Storage):

○빠른 읽기/쓰기를 지원하는 저장소:

HBase, Cassandra, Redis와 같은 NoSQL 데이터베이스를 사용하여 실시간 데이터를 저장하고 검색할 수 있어야 합니다.

○시간 기반 데이터 저장소: InfluxDB, TimescaleDB와 같은 시계열 데이터베이스를 사용하여 시간 기반 데이터를 효율적으로 저장하고 관리할 수 있습니다.

4.데이터 분석 및 시각화 (Data Analytics and Visualization):

○실시간 분석 도구: 실시간 분석을 지원하는 도구 (예: Apache Druid, ClickHouse)를 사용하여 실시간 쿼리와 분석을 수행해야 합니다.

○시각화 도구: Grafana, Kibana와 같은 시각화 도구를 사용하여 실시간 데이터 시각화를 제공해야 합니다.

5.확장성 (Scalability):

○수평적 확장: 시스템이 증가하는 데이터 양을 처리할 수 있도록 수평적으로 확장 가능한 아키텍처를 설계해

야 합니다.

○자동 확장:

Kubernetes와 같은 오케스트레이션 도구를 사용하여 필요에 따라 리소스를 자동으로 확장할 수 있도록 해야 합니다.

6.신뢰성 및 가용성 (Reliability and Availability):

○데이터 손실 방지:

데이터 손실을 방지하기 위해 데이터 복제 및 백업 전략을 구현해야 합니다.

○고가용성: 장애 발생 시에도 시스템이 지속적으로 작동할 수 있도록 고가용성 아키텍처를 설계해야 합니다.

7.보안 (Security):

○데이터 암호화: 데이터 전송 시와 저장 시 모두 암호화를 적용하여 데이터 보안을 강화해야 합니다.

○접근 제어: 역할 기반 접근 제어 (RBAC)를 사용하여 민감 데이터에 대한 접근을 관리해야 합니다.

○감사 및 모니터링: 실시간 데이터 접근 및 변경 내역을 모니터링하고 감사 로그를 유지해야 합니다.

8.유지보수 및 운영 (Maintenance and Operations):

○모니터링 및 경고 시스템: Prometheus, ELK Stack과 같은 모니터링 및 로깅 도구를 사용하여 시스템 성능을 모니터링하고 문제

			<p>발생 시 경고를 받을 수 있어야 합니다.</p> <p>○자동화된 배포: CI/CD 파이프라인을 구축하여 코드 변경 사항을 자동으로 배포하고, 지속적인 통합 및 배포를 가능하게 해야 합니다.</p> <p>9.지연 허용 범위 (Latency Tolerance):</p> <p>○저지연 처리 필요성: 애플리케이션 요구사항에 따라 허용 가능한 지연 범위를 설정하고, 이를 충족하기 위한 기술을 선택해야 합니다.</p> <p>○배치 처리와의 통합: 실시간 처리와 배치 처리를 결합하여 하이브리드 데이터 처리 아키텍처를 구현할 수 있습니다.</p>	
김미식	3	<p>한국에 본사를 둔 글로벌 기업입니다. 전 세계 지사에서 발생하는 실적에 대해 실시간으로 시각화 분석하고 당월 월실적에 대한 예측도 하고 싶습니다.</p> <p>실적은 로그성으로 발생하고 지사마다 각국 환경에 따라 포맷이 다를 수 있습니다.</p> <p>적당한 데이터플랫폼 구축에 대해 제안해보셔요</p>		<p>데이터파이프라인 설계 실시간 분석, 예측, 데이터전처리</p>
김미식	4	<p>T사는 사람들에게 특정장소 방문미션 성공 시 포인트를 제공합니다. 특정</p>		<p>확장성, 실시간/배치, 메타데이터, 보안, 데이터분류 (hot/Cold)</p>

		<p>장소는 날마다 달라질 수 있고 시간대에 따라 사용자 접속양 차이가 클 수 있습니다. 개인 정보는 철저히 보호되어야하고 1년치 데이터를 분석할 수 있는 데이터 플랫폼을 구축하고자합니다.(장소별로 실시간&일마감 분석 필요) 코로나 등 특수상황 발생 시 최대 5년치 데이터에 대한 분석이 필요할 수 있어 최장 보관기간은 5년으로 하고싶고 비용 최적화 필요합니다. 방문장소가 다양하게 변하고 있고 지역마다 장소명이 같을 수도 있어 장소별 분석을 위해 방문장소 정보를 분석가들에게 어떻게 제공할 건지에 대한 방안도 필요합니다</p>		
정경주	5	<p>온프레미스 데이터 웨어하우스 시스템을 클라우드로 전환 시 가장 유념해야 할 고려 사항을 제시해 주세요.</p>	<p>1. 비용 및 ROI 클라우드 기반 빅데이터 플랫폼은 초기 투자 비용이 적게 들 수 있지만, 장기적인 운영 비용은 사용량 및 데이터 규모에 따라 달라질 수 있습니다. 예상되는 비용을 정확하게 계산하고, 클라우드 서비스 제공업체의 가격 모델을 꼼꼼하게 비교 분석해야 합니다. 또한, 데이터 마이그레이션, 시스템 관리, 보</p>	클라우드 전환 방안

안 강화 등 클라우드 전환으로 인해 발생하는 추가 비용도 고려해야 합니다.

투자 수익률(ROI)을 계산하여 클라우드 전환의 경제적 이득을 명확하게 제시해야 합니다.

2. 성능 및 확장성

클라우드 기반 빅데이터 플랫폼은 온프레미스 시스템보다 뛰어난 확장성을 제공하며, 워크로드 증가에 따라 용량을 빠르게 확장할 수 있습니다.

하지만, 네트워크 대역폭 및 레이턴시가 성능에 영향을 미칠 수 있으므로, 클라우드 서비스 제공업체의 네트워크 인프라를 평가해야 합니다.

또한, 데이터 웨어하우스 시스템의 처리 능력과 저장 용량 요구 사항을 정확하게 예측하고, 이에 맞는 클라우드 리소스를 선택해야 합니다.

3. 보안 및 규정 준수

클라우드 기반 빅데이터 플랫폼은 민감한 데이터를 저장하기 때문에, 강력한 보안 및 규정 준수 요구 사항을 충족해야 합니다.

데이터 암호화, 액세스 제어, 데이터 손실 방지 등 다양한 보안 기능을 제공하는 클라우드 서비스 제공업체를 선택해야 합니다.

또한, 개인정보보호법 (GDPR), 산업별 규제 등 관련 규정을 준수해야 합니다.

4. 데이터 마이그레이션 및 통합
온프레미스 데이터를 클라우드로 마이그레이션하는 과정은 복잡하고 시간이 많이 걸릴 수 있습니다.

데이터 마이그레이션 전략을 수립하고, 데이터 손실 및 오류를 방지하기 위한 프로세스를 정의해야 합니다. 또한, 기존 온프레미스 시스템과 클라우드 기반 빅데이터 플랫폼을 통합하고, 데이터 흐름을 원활하게 연결해야 합니다.

5. 시스템 관리 및 운영
클라우드 기반 빅데이터 플랫폼은 온프레미스 시스템보다 관리 및 운영이 간편할 수 있습니다.

하지만, 클라우드 서비스 제공업체의 서비스 수준 계약(SLA)을 확인하고, 시스템 가용성 및 성능을 모니터링해야 합니다.

또한, 클라우드 환경에서 발생하는 보안 위협 및 새로운 기술 트렌드를 지속적으로 파악하고 대응해야 합니다.

6. 인력 및 기술
클라우드 기반 빅데이터 플랫폼을 관리 및

			<p>운영するには, 클라우드 컴퓨팅 및 관련 기술에 대한 전문 지식을 갖춘 인력이 필요합니다.</p> <p>직원 교육 및 개발을 통해 필요한 기술 능력을 갖추도록 지원해야 합니다.</p> <p>또한, 클라우드 서비스 제공업체의 지원 서비스를 활용하여 전문적인 도움을 받을 수도 있습니다.</p> <p>7. 변경 관리</p> <p>클라우드 기반 빅데이터 플랫폼으로 전환하는 것은 조직의 IT 인프라 및 운영 방식에 큰 변화를 가져올 수 있습니다.</p> <p>직원들에게 변화에 대한 교육을 제공하고, 새로운 시스템 사용에 대한 지원을 제공해야 합니다</p>	
정경주	6	<p>다양한 데이터 플랫폼 중에서 고객에게 적합한 플랫폼을 선정하기 위해 중요하게 살펴야 할 항목을 제시해 주세요.</p>	<p>1. 요구 사항:</p> <p>기능: 플랫폼이 사용자의 모든 기능 요구 사항을 충족하는지 확인해야 합니다.</p> <p>성능: 플랫폼이 사용자의 처리량, 데이터 규모, 성능 요구 사항을 처리할 수 있는지 확인해야 합니다.</p> <p>확장성: 플랫폼이 사용자의 데이터 및 사용자 트래픽 증가에 따라 확장될 수 있는지 확인해야 합니다.</p> <p>보안: 플랫폼이 사용자 데이터를 보호하고 액세스 제어, 암호화, 백</p>	<p>데이터플랫폼 선정 조건</p>

업 및 재해 복구와 같은 강력한 보안 기능을 제공하는지 확인해야 합니다.

규정 준수: 플랫폼이 사용자의 산업 및 규제 요구 사항을 준수하는지 확인해야 합니다.

2. 비용:

초기 투자 비용: 플랫폼 라이선스, 하드웨어, 소프트웨어 및 기타 초기 비용을 고려해야 합니다.

지속적인 운영 비용: 플랫폼 사용료, 인프라 비용, 유지 관리 비용 및 기타 지속적인 비용을 고려해야 합니다.

가격 모델: 플랫폼 제공업체가 제공하는 다양한 가격 모델 (예: 사용량 기반, 구독 기반)을 비교해야 합니다.

3. 사용 편의성:

사용자 인터페이스: 플랫폼의 사용자 인터페이스가 사용하기 쉽고 직관적인지 확인해야 합니다.

문서 및 교육 자료: 플랫폼 사용을 배우고 문제를 해결하는 데 도움이 되는 충분한 문서 및 교육 자료가 있는지 확인해야 합니다.

고객 지원: 플랫폼 제공업체가 문제 해결 및 지원을 위해 우수한 고객 지원을 제공하는지 확인해야 합니다.

4. 기술 스택:

호환성: 플랫폼이 사용

자의 기존 기술 스택 및 시스템과 호환되는지 확인해야 합니다.
개발 도구 및 언어: 플랫폼이 사용자의 선호하는 개발 도구 및 언어를 지원하는지 확인해야 합니다.
통합: 플랫폼이 사용자의 기존 애플리케이션 및 시스템과 통합될 수 있는지 확인해야 합니다.

5. 공급업체:
신뢰성: 플랫폼 제공업체가 신뢰할 수 있고 재무적으로 안정적인지 확인해야 합니다.
실적: 플랫폼 제공업체가 사용자와 유사한 업계에서 성공적인 실적을 가지고 있는지 확인해야 합니다.
고객 만족도: 플랫폼 제공업체의 고객 만족도 및 리뷰를 확인해야 합니다.

6. 기타 고려 사항:
장기적인 로드맵: 플랫폼 제공업체가 미래에 어떤 계획을 가지고 있는지 확인하고, 플랫폼이 사용자의 장기적인 요구 사항을 충족할 수 있는지 확인해야 합니다.
커뮤니티: 플랫폼 사용자 커뮤니티가 활발하고 도움이 되는지 확인해야 합니다.

오픈 소스: 플랫폼이 오픈 소스인 경우, 사용자가 코드를 검토하

			고 수정할 수 있는지 확인해야 합니다.	
윤준구	상황제시	<p>[고객 상황] 고객 상황 : 글로벌 패션 브랜드 '스 타일 잇'은 온라인 쇼핑몰 운영 데이터를 활용하여 고객 맞춤형 마케팅 전략을 수립하고자 한다.</p> <p>Pain Point : 막대한 구매 데이터와 고객 행동 데이터를 효율적으로 분석하고 활용하기 어려움</p> <p>업무 요건 :</p> <ul style="list-style-type: none"> - 다양한 데이터 소스(웹로그, 거래 데이터, CRM 등)를 통합하고 처리할 수 있는 데이터 파이프라인 구축 - 고객 구매 패턴, 인구 통계, 선호도 분석을 위한 머신러닝 모델 개발 - 실시간 고객 추천 및 맞춤형 프로모션 자동화 시스템 구축 - 데이터 기반 마케팅 캠페인 효과 측정 및 최적화 성공 지표 : - 구매 전환율 증대 - 고객 유지율 향상 - 마케팅 투자 수익률 (ROI) 최대화 	<p>[업무 시나리오 1]</p> <ol style="list-style-type: none"> 1. 다양한 데이터 소스 통합: 웹로그, 거래 데이터, CRM 등 다양한 데이터 소스를 통합하고 처리할 수 있는 데이터 파이프라인 구축 2. 실시간 데이터 처리: 구매, 검색, 클릭 등 사용자 행동 데이터를 실시간으로 처리 및 분석 3. 데이터 저장 및 관리: 대규모 데이터 저장 및 관리를 위한 분산 데이터베이스 구축 4. 데이터 품질 관리: 데이터 누락, 오류, 불일치 등 데이터 품질 문제 해결 및 데이터 정제 5. 고객 분석: 고객 구매 패턴, 인구 통계, 선호도 등을 분석하기 위한 머신러닝 모델 개발 6. 실시간 추천: 고객 맞춤형 상품 추천을 위한 실시간 추천 시스템 구축 7. 마케팅 자동화: 고객 세분화, 타겟팅, 캠페인 자동화 등 마케팅 자동화 시스템 구축 8. A/B 테스트: 다양한 마케팅 전략의 효과를 비교 분석하기 위한 A/B 테스트 플랫폼 구축 9. 데이터 시각화: 분석 결과를 시각화하고 이해하기 쉽게 제시하기 위한 데이터 시각화 도구 활용 	

			10.보안 및 개인정보 보호: 고객 데이터 보안 및 개인정보 보호를 위한 강력한 보안 시스템 구축	
윤준구	7	업무 시나리오 1에서 ①②에 대해 수집 파이프라인을 2가지 안으로 설계하세요. 각 각의 안에 대해서 선정한 사유를 기술하세요.		수집파이프라인 (실시간, 저장/관리)
윤준구	8	문제 1에서 설계된 수집 레이어 기반으로 ③④을 기준으로 data warehouse/data lake 레이어를 설계하세요		DW, DataLake 설계방안
조항륜	9	<p>사례 문제 1번 : 온라인 소매업체 데이터 파이프라인 구축</p> <p>시나리오</p> <p>당신은 큰 온라인 소매업체에서 데이터 엔지니어로 일하고 있습니다. 이 소매업체는 다양한 소스(웹사이트, 모바일 앱, 물리적 상점 등)에서 데이터를 수집하고 있으며, 이 데이터를 활용하여 실시간 추천 시스템을 운영하고자 합니다. 현재 데이터는 여러 개의 서로 다른 데이터베이스와 파일 시스템에 분산되어 있습니다. 또한, 기존의 데이터 파이</p>	<p>단계별 접근 방법</p> <p>1.데이터 수집 및 통합: Kafka Connect를 사용하여 다양한 데이터 소스에서 데이터를 수집하고, 이 데이터를 Kafka 토픽에 게시합니다.</p> <p>2.데이터 정제 및 변환: Apache Spark를 사용하여 수집된 데이터를 정제하고 변환합니다.</p> <p>3.실시간 데이터 스트리밍: Apache Kafka를 사용하여 실시간 데이터 스트리밍을 구현합니다.</p> <p>4.데이터 저장: Snowflake를 데이터 웨어하우스로 선택하고, 정제된 데이터를 저장합니다.</p> <p>5.모니터링 및 알림: Prometheus와</p>	

프라인은 느리고 비효율적이어서 실시간 처리가 어렵습니다.

요구 사항

1.데이터 통합: 모든 소스에서 데이터를 수집하여 중앙 데이터 웨어하우스로 통합해야 합니다.

2.실시간 처리: 실시간 추천 시스템을 운영하기 위해 실시간 데이터 스트리밍을 구현해야 합니다.

3.확장성: 데이터 파이프라인은 트래픽 증가에 대비해 확장 가능해야 합니다.

4.신뢰성: 데이터 손실 없이 안정적으로 데이터를 처리해야 합니다.

해결해야 할 문제
1.데이터 소스 연결: 여러 소스에서 데이터를 수집하기 위해 필요한 기술과 도구를 선택해야 합니다.

2.데이터 정제: 수집된 데이터를 분석 가능하게 정제해야 합니다.

3.데이터 스트리밍: 실시간 처리를 위한 스트리밍 솔루션을 설계하고 구현해야 합니다.

4.데이터 저장: 통합된 데이터를 저

Grafana를 사용하여 데이터 파이프라인의 상태를 모니터링하고, 문제가 발생하면 알림을 받습니다.

계별 해결 솔루션의 선택 이유와 장단점

1. 데이터 수집 및 통합
선택한 솔루션: Kafka Connect

선택 이유:

●Kafka Connect는 다양한 소스에서 데이터를 쉽게 수집하고, Kafka로 데이터를 전송할 수 있는 커넥터를 제공합니다.

●다양한 커넥터를 통해 쉽게 확장할 수 있으며, 분산 처리와 스케일링이 용이합니다.

장점:

●확장성: 필요에 따라 새로운 커넥터를 추가하여 쉽게 확장할 수 있습니다.

●호환성: 다양한 데이터 소스(MySQL, PostgreSQL, Elasticsearch 등)와 호환됩니다.

●실시간 처리: 실시간으로 데이터를 수집하고 처리할 수 있습니다.

단점:

●복잡성: 다양한 커넥터를 설정하고 관리하는 데 시간이 걸릴 수 있습니다.

●자원 사용량: 많은 데이터를 처리할 때 자원 사용량이 증가할 수 있습니다.

2. 데이터 정제 및 변환

장할 적절한 데이터 웨어하우스를 선택하고 설계해야 합니다.

5.데이터 분석 및 활용: 데이터 과학 팀이 실시간으로 데이터를 분석하고 활용할 수 있도록 지원해야 합니다.

고려할 기술 스택

●데이터 수집:

Apache Nifi, Logstash, Kafka Connect

●데이터 정제:

Apache Spark, Apache Flink

●데이터 스트리밍:

Apache Kafka, Amazon Kinesis

●데이터 저장:

Amazon Redshift, Google BigQuery, Snowflake

●모니터링 및 알림: Prometheus, Grafana

선택한 솔루션:

Apache Spark

선택 이유:

●Apache Spark는 대규모 데이터 처리를 위한 강력한 도구로, 다양한 데이터 소스에서 데이터를 읽고, 변환하고, 정제할 수 있습니다.

●배치 처리와 스트리밍 처리를 모두 지원하여 유연한 데이터 파이프라인을 구축할 수 있습니다.

장점:

●성능: 인메모리 컴퓨팅을 통해 빠른 데이터 처리가 가능합니다.

●유연성: 다양한 데이터 소스와 통합하여 데이터를 처리할 수 있습니다.

●확장성: 클러스터 환경에서 쉽게 확장할 수 있습니다.

단점:

●복잡성: 설정과 사용법이 복잡할 수 있습니다.

●자원 사용량: 대규모 데이터를 처리할 때 많은 자원이 필요할 수 있습니다.

3. 실시간 데이터 스트리밍

선택한 솔루션:

Apache Kafka와

Apache Flink

선택 이유:

●Apache Kafka는 높은 처리량과 낮은 지연 시간을 제공하며, 실시간 데이터 스트리밍에 적합합니다.

●Apache Flink는 강

력한 실시간 데이터 처리 엔진으로, 복잡한 데이터 변환과 분석 작업을 실시간으로 수행할 수 있습니다.

장점:

- 신뢰성: 데이터 손실 없이 높은 신뢰성으로 데이터를 전송합니다.

- 확장성: 대규모 데이터 스트리밍을 처리할 수 있습니다.

- 유연성: 다양한 스트리밍 애플리케이션을 쉽게 구축할 수 있습니다.

단점:

- 복잡성: 설정과 운영이 복잡할 수 있습니다.

- 학습 곡선: 사용법을 익히는 데 시간이 걸릴 수 있습니다.

4. 데이터 저장

선택한 솔루션:

Snowflake

선택 이유:

- Snowflake는 클라우드 기반의 데이터 웨어하우스로, 높은 확장성과 성능을 제공합니다.

- 다양한 데이터 소스와 통합이 용이하며, 복잡한 데이터 쿼리를 효율적으로 처리할 수 있습니다.

장점:

- 확장성: 클라우드 기반으로 필요에 따라 자원을 쉽게 확장할 수 있습니다.

- 성능: 고성능 데이터 처리와 쿼리가 가능합니다.

- 관리 용이성: 자동화

된 관리 기능을 제공하여 운영 부담이 적습니다.

단점:

- 비용: 대규모 데이터를 처리할 때 비용이 증가할 수 있습니다.

- 의존성: 클라우드 서비스에 대한 의존성이 있습니다.

5. 모니터링 및 알림
선택한 솔루션:

Prometheus와
Grafana

선택 이유:

- Prometheus는 강력한 모니터링과 경고 기능을 제공하며, Grafana와 함께 사용하여 데이터를 시각화할 수 있습니다.

- 다양한 데이터 소스와 쉽게 통합할 수 있으며, 실시간 모니터링이 가능합니다.

장점:

- 실시간 모니터링: 실시간으로 시스템 상태를 모니터링할 수 있습니다.

- 시각화: Grafana를 통해 데이터를 시각화하여 쉽게 이해할 수 있습니다.

- 확장성: 다양한 메트릭을 추가하여 쉽게 확장할 수 있습니다.

단점:

- 설정 복잡성: 설정과 운영이 복잡할 수 있습니다.

- 학습 곡선: 사용법을 익히는 데 시간이 걸릴 수 있습니다.

조항문	10	<p>사례 문제 2번: 금융 기관의 데이터 레이크 구축 문제 정의</p> <p>금융 기관은 다양한 소스(거래 시스템, 고객 관계 관리 시스템, 외부 데이터 소스 등)에서 데이터를 수집하여 통합된 데이터 레이크를 구축하려고 합니다. 데이터는 다양한 형식(구조화, 반구조화, 비구조화)으로 존재하며, 실시간 분석과 배치 분석을 모두 지원해야 합니다. 현재 시스템은 여러 데이터 사일로(silo)로 구성되어 있어 데이터 접근성과 분석이 어렵습니다.</p> <p>요구 사항</p> <ol style="list-style-type: none"> 1. 데이터 통합: 다양한 소스에서 데이터를 통합하여 데이터 레이크에 저장해야 합니다. 2. 데이터 정제 및 변환: 수집된 데이터를 정제하고 변환하여 분석 가능하게 만들어야 합니다. 3. 실시간 및 배치 처리: 실시간 스트리밍 데이터와 배치 데이터를 모두 처리할 수 있어야 합니다. 4. 보안 및 규정 준수: 민감한 금융 데이터 	<p>해결 방법</p> <ol style="list-style-type: none"> 1. 데이터 수집 및 통합 선택한 솔루션: Apache Nifi, Kafka Connect, AWS Glue 선택 이유: <ul style="list-style-type: none"> ● Apache Nifi는 다양한 소스에서 데이터를 수집하고, 실시간으로 데이터 흐름을 관리할 수 있습니다. ● Kafka Connect는 다양한 소스로부터 데이터를 쉽게 수집하고, Kafka로 데이터를 전송할 수 있는 커넥터를 제공합니다. ● AWS Glue는 서버리스 데이터 통합 서비스로, 다양한 데이터 소스를 연결하고 ETL 작업을 수행할 수 있습니다. <p>장점:</p> <ul style="list-style-type: none"> ● 유연성: 다양한 소스에서 데이터를 수집할 수 있습니다. ● 확장성: 필요에 따라 쉽게 확장할 수 있습니다. ● 자동화: ETL 작업을 자동화하여 효율적으로 관리할 수 있습니다. <p>단점:</p> <ul style="list-style-type: none"> ● 설정 복잡성: 다양한 도구를 설정하고 관리하는 데 시간이 걸릴 수 있습니다. ● 비용: AWS Glue와 같은 클라우드 서비스를 사용할 경우 비용이 발생할 수 있습니다. <ol style="list-style-type: none"> 2. 데이터 정제 및 변환 선택한 솔루션: Apache Spark,
-----	----	---	--

이터를 보호하고
규정을 준수해야
합니다.

5.확장성 및 신뢰
성: 대규모 데이터
를 처리할 수 있도
록 확장 가능하고
신뢰성이 있어야
합니다.

해결해야 할 문제

1.데이터 소스 연
결: 다양한 데이터
소스에서 데이터를
수집하기 위해 필
요한 기술과 도구
를 선택해야 합니
다.

2.데이터 정제 및
변환: 수집된 데이
터를 정제하고 변
환해야 합니다.

3.데이터 저장: 데
이터 레이크에 데
이터를 저장할 적
절한 스토리지 솔
루션을 선택해야
합니다.

4.데이터 처리: 실
시간 및 배치 데이
터를 처리할 수 있
는 파이프라인을
구축해야 합니다.

5.보안 및 규정 준
수: 데이터 보안과
규정 준수를 보장
해야 합니다.

기술 스택

●데이터 수집:
Apache Nifi,
Kafka Connect,
AWS Glue

●데이터 정제 및
변환: Apache
Spark, AWS
Glue

AWS Glue

선택 이유:

●Apache Spark는
대규모 데이터 처리를
위한 강력한 도구로,
다양한 데이터 소스에
서 데이터를 읽고, 변
환하고, 정제할 수 있
습니다.

●AWS Glue는 서버
리스 ETL 서비스로,
데이터 정제와 변환을
자동화할 수 있습니다.

장점:

●성능: 인메모리 컴퓨
팅을 통해 빠른 데이터
처리가 가능합니다.

●유연성: 다양한 데이
터 소스와 통합하여 데
이터를 처리할 수 있습
니다.

●자동화: ETL 작업을
자동화하여 효율적으
로 관리할 수 있습니
다.

단점:

●복잡성: 설정과 사용
법이 복잡할 수 있습니
다.

●자원 사용량: 대규모
데이터를 처리할 때 많
은 자원이 필요할 수
있습니다.

3. 데이터 저장 선택한
솔루션: AWS S3,
Hadoop HDFS,
Google Cloud
Storage

선택 이유:

●AWS S3는 확장성
이 뛰어난 클라우드 스토리지
솔루션으로, 대
규모 데이터를 저장할
수 있습니다.

●Hadoop HDFS는
분산 파일 시스템으로,

●데이터 저장:
AWS S3,
Hadoop HDFS,
Google Cloud
Storage
●데이터 처리:
Apache Spark,
Apache Flink,
AWS Lambda
●보안 및 규정 준
수: AWS IAM,
AWS KMS,
Azure Active
Directory
●모니터링 및 알
림: Prometheus,
Grafana, AWS
CloudWatch

대규모 데이터 저장에
적합합니다.
●Google Cloud
Storage는 클라우드
기반의 스토리지 솔루
션으로, 높은 확장성과
성능을 제공합니다.
장점:
●확장성: 대규모 데이
터를 저장할 수 있습니
다.
●비용 효율성: 사용한
만큼 비용을 지불하는
유연한 요금제를 제공
합니다.
●통합성: 다양한 분석
도구와 쉽게 통합할 수
있습니다.
단점:
●의존성: 클라우드 서
비스에 대한 의존성이
있습니다.
●비용: 대규모 데이
터를 장기간 저장할 경우
비용이 증가할 수 있습
니다.
4. 데이터 처리 선택한
솔루션: Apache
Spark, Apache
Flink, AWS Lambda
선택 이유:
●Apache Spark는
배치 및 스트리밍 데이
터를 모두 처리할 수
있는 강력한 데이터 처
리 엔진입니다.
●Apache Flink는 실
시간 스트리밍 데이터
처리를 위한 도구로,
높은 처리량과 낮은 지
연 시간을 제공합니다.
●AWS Lambda는
서버리스 컴퓨팅 서비
스로, 이벤트 기반 데
이터 처리를 자동화할
수 있습니다.

장점:

- 유연성: 배치 및 실시간 데이터를 모두 처리할 수 있습니다.

- 확장성: 클러스터 환경에서 쉽게 확장할 수 있습니다.

- 자동화: 이벤트 기반으로 데이터 처리를 자동화할 수 있습니다.

단점:

- 복잡성: 설정과 운영이 복잡할 수 있습니다.

- 학습 곡선: 사용법을 익히는 데 시간이 걸릴 수 있습니다.

5. 보안 및 규정 준수

선택한 솔루션: AWS

IAM, AWS KMS,

Azure Active

Directory

선택 이유:

- AWS IAM은 세분화된 권한 관리를 통해 데이터 접근을 제어할 수 있습니다.

- AWS KMS는 데이터 암호화를 통해 민감한 데이터를 보호할 수 있습니다.

- Azure Active Directory는 사용자 인증 및 권한 관리를 위한 강력한 도구입니다.

장점:

- 보안성: 민감한 데이터를 보호할 수 있습니다.

- 규정 준수: 다양한 보안 규정을 준수할 수 있습니다.

- 관리 용이성: 권한 관리와 데이터 보호를 중앙에서 관리할 수 있

습니다.

단점:

- 복잡성: 설정과 관리가 복잡할 수 있습니다.

- 비용: 보안 솔루션을 사용하는 데 비용이 발생할 수 있습니다.

6. 모니터링 및 알림
선택한 솔루션:

Prometheus,
Grafana, AWS
CloudWatch

선택 이유:

- Prometheus는 강력한 모니터링과 경고 기능을 제공하며, Grafana와 함께 사용하여 데이터를 시각화할 수 있습니다.

- AWS CloudWatch는 AWS 리소스를 모니터링하고 로그를 수집할 수 있는 관리형 서비스입니다.

장점:

- 실시간 모니터링: 실시간으로 시스템 상태를 모니터링할 수 있습니다.

- 시각화: Grafana를 통해 데이터를 시각화하여 쉽게 이해할 수 있습니다.

- 통합성: 다양한 서비스와 쉽게 통합할 수 있습니다.

단점:

- 설정 복잡성: 설정과 운영이 복잡할 수 있습니다.

- 학습 곡선: 사용법을 익히는 데 시간이 걸릴 수 있습니다.

이와 같은 이유로 각 단계별 솔루션을 선택

			하고, 해당 솔루션의 장단점을 고려하여 데이터 레이크를 설계하고 구현합니다.	
조항문	11	<p>사례 문제 3번: 통신사의 데이터 파이프라인 구축 문제 정의</p> <p>통신사는 다양한 소스(콜 로그, 네트워크 트래픽, 고객 데이터 등)에서 데이터를 수집하여 통합된 데이터 파이프라인을 구축하려고 합니다. 데이터는 실시간 분석과 배치 분석을 모두 지원해야 하며, 네트워크 운영, 고객 행동 분석, 마케팅 캠페인 최적화 등에 활용될 것입니다. 현재 시스템은 여러 데이터 사일로(silo)로 구성되어 있어 데이터 접근성과 분석이 어렵습니다.</p> <p>요구 사항</p> <p>1.데이터 통합: 다양한 소스에서 데이터를 통합하여 중앙 데이터 웨어하우스에 저장해야 합니다.</p> <p>2.실시간 및 배치 처리: 실시간 스트리밍 데이터와 배치 데이터를 모두 처리할 수 있어야 합니다.</p> <p>3.확장성: 데이터 파이프라인은 대규모</p>	<p>해결 방법</p> <p>1. 데이터 수집 및 통합 선택한 솔루션: Apache Nifi, Kafka Connect, Flume</p> <p>선택 이유:</p> <ul style="list-style-type: none"> ●Apache Nifi는 데이터 플로우를 시각적으로 관리할 수 있어 다양한 데이터 소스에서 데이터를 쉽게 수집하고 통합할 수 있습니다. ●Kafka Connect는 다양한 소스로부터 데이터를 수집하여 실시간으로 Kafka로 전송할 수 있습니다. ●Flume은 대규모 로그 데이터를 효율적으로 수집하고 전달하는데 유용합니다. <p>장점:</p> <ul style="list-style-type: none"> ●유연성: 다양한 소스에서 데이터를 수집할 수 있습니다. ●확장성: 필요에 따라 쉽게 확장할 수 있습니다. ●시각화: Nifi를 통해 데이터 플로우를 시각적으로 관리할 수 있습니다. <p>단점:</p> <ul style="list-style-type: none"> ●설정 복잡성: 다양한 도구를 설정하고 관리하는 데 시간이 걸릴 수 있습니다. ●자원 사용량: 많은 데이터를 처리할 때 자 	

모 데이터 처리와 증가하는 데이터 양에 대비해 확장 가능해야 합니다.

4.신뢰성: 데이터 손실 없이 안정적으로 데이터를 처리해야 합니다.

5.보안 및 규정 준수: 민감한 데이터를 보호하고 규정을 준수해야 합니다.

해결해야 할 문제

1.데이터 소스 연결: 다양한 데이터 소스에서 데이터를 수집하기 위해 필요한 기술과 도구를 선택해야 합니다.

2.데이터 정제 및 변환: 수집된 데이터를 정제하고 변환해야 합니다.

3.데이터 저장: 통합된 데이터를 저장할 적절한 데이터 웨어하우스를 선택해야 합니다.

4.데이터 처리: 실시간 및 배치 데이터를 처리할 수 있는 파이프라인을 구축해야 합니다.

5.보안 및 규정 준수: 데이터 보안과 규정 준수를 보장해야 합니다.

기술 스택

●데이터 수집: Apache Nifi, Kafka Connect, Flume

●데이터 정제 및

원 사용량이 증가할 수 있습니다.

2. 데이터 정제 및 변환 선택한 솔루션:

Apache Spark, Apache Flink

선택 이유:

●Apache Spark는 대규모 데이터 처리를 위한 강력한 도구로, 다양한 데이터 소스에서 데이터를 읽고, 변환하고, 정제할 수 있습니다.

●Apache Flink는 실시간 스트리밍 데이터 처리를 위한 도구로, 복잡한 데이터 변환과 분석 작업을 실시간으로 수행할 수 있습니다.

장점:

●성능: 인메모리 컴퓨팅을 통해 빠른 데이터 처리가 가능합니다.

●유연성: 다양한 데이터 소스와 통합하여 데이터를 처리할 수 있습니다.

●확장성: 클러스터 환경에서 쉽게 확장할 수 있습니다.

단점:

●복잡성: 설정과 사용법이 복잡할 수 있습니다.

●자원 사용량: 대규모 데이터를 처리할 때 많은 자원이 필요할 수 있습니다.

3. 데이터 저장 선택한

솔루션: Amazon Redshift, Google BigQuery, Snowflake

선택 이유:

변환: Apache Spark, Apache Flink

●데이터 저장:

Amazon Redshift, Google BigQuery, Snowflake

●데이터 처리:

Apache Spark, Apache Flink, Apache Beam

●보안 및 규정 준수: AWS IAM, AWS KMS, Azure Active Directory

●모니터링 및 알림: Prometheus, Grafana, AWS CloudWatch

●Amazon Redshift는 대규모 데이터 웨어하우스로, 분석 쿼리를 빠르게 처리할 수 있습니다.

●Google BigQuery는 서버리스 데이터 웨어하우스로, 빠른 쿼리 성능과 유연한 요금제를 제공합니다.

●Snowflake는 클라우드 기반 데이터 웨어하우스로, 높은 확장성과 성능을 제공합니다. 장점:

●확장성: 대규모 데이터를 저장할 수 있습니다.

●성능: 고성능 데이터 처리와 쿼리가 가능합니다.

●유연성: 다양한 분석 도구와 쉽게 통합할 수 있습니다.

단점:

●비용: 대규모 데이터를 장기간 저장할 경우 비용이 증가할 수 있습니다.

●의존성: 클라우드 서비스에 대한 의존성이 있습니다.

4. 데이터 처리 선택한

솔루션: Apache Spark, Apache Flink, Apache Beam

선택 이유:

●Apache Spark는 배치 및 스트리밍 데이터를 모두 처리할 수 있는 강력한 데이터 처리 엔진입니다.

●Apache Flink는 실시간 스트리밍 데이터 처리를 위한 도구로,

높은 처리량과 낮은 지연 시간을 제공합니다.

●Apache Beam은 다양한 실행 엔진을 추상화하여 동일한 파이프라인 코드를 여러 엔진에서 실행할 수 있게 합니다.

장점:

●유연성: 배치 및 실시간 데이터를 모두 처리할 수 있습니다.

●확장성: 클러스터 환경에서 쉽게 확장할 수 있습니다.

●통합성: 여러 실행 엔진과 통합할 수 있습니다.

단점:

●복잡성: 설정과 운영이 복잡할 수 있습니다.

●학습 곡선: 사용법을 익히는 데 시간이 걸릴 수 있습니다.

5. 보안 및 규정 준수
선택한 솔루션: AWS IAM, AWS KMS, Azure Active Directory

선택 이유:

●AWS IAM은 세분화된 권한 관리를 통해 데이터 접근을 제어할 수 있습니다.

●AWS KMS는 데이터 암호화를 통해 민감한 데이터를 보호할 수 있습니다.

●Azure Active Directory는 사용자 인증 및 권한 관리를 위한 강력한 도구입니다.

장점:

●보안성: 민감한 데이

터를 보호할 수 있습니다.

●규정 준수: 다양한 보안 규정을 준수할 수 있습니다.

●관리 용이성: 권한 관리와 데이터 보호를 중앙에서 관리할 수 있습니다.

단점:

●복잡성: 설정과 관리가 복잡할 수 있습니다.

●비용: 보안 솔루션을 사용하는 데 비용이 발생할 수 있습니다.

6. 모니터링 및 알림
선택한 솔루션:

Prometheus,
Grafana, AWS
CloudWatch

선택 이유:

●Prometheus는 강력한 모니터링과 경고 기능을 제공하며, Grafana와 함께 사용하여 데이터를 시각화할 수 있습니다.

●AWS CloudWatch는 AWS 리소스를 모니터링하고 로그를 수집할 수 있는 관리형 서비스입니다.

장점:

●실시간 모니터링: 실시간으로 시스템 상태를 모니터링할 수 있습니다.

●시각화: Grafana를 통해 데이터를 시각화하여 쉽게 이해할 수 있습니다.

●통합성: 다양한 서비스와 쉽게 통합할 수 있습니다.

단점:

			<ul style="list-style-type: none"> ● 설정 복잡성: 설정과 운영이 복잡할 수 있습니다. ● 학습 곡선: 사용법을 익히는 데 시간이 걸릴 수 있습니다. 	
김소라	상황제시	<p>OTT 회사에서 실시간 스트리밍 데이터 파이프라인을 구축하여 실시간 추천 시스템을 구축하려고 합니다. 사용자 행동 데이터를 실시간으로 수집, 처리, 저장, 분석하여 사용자에게 맞춤형 콘텐츠를 추천하는 것을 목표로 합니다.</p> <p>회사의 요구사항은 아래와 같습니다.</p> <ol style="list-style-type: none"> 1. 실시간 데이터 수집: 사용자 행동 데이터를 실시간으로 수집 (예: 재생 시작/중지, 시청 완료, 평가, 검색 기록). 2. 데이터 처리 및 변환: 수집된 데이터를 실시간으로 정제, 변환 및 사용자 프로필 업데이트. 3. 데이터 저장: 실시간 처리된 데이터를 안정적이고 확장 가능한 저장소에 저장. 4. 추천 알고리즘 적용: 실시간으로 수집된 데이터를 바탕으로 추천 알고리즘 실행. 		

		<p>5. 실시간 추천 제공: 사용자 인터페이스에 실시간으로 추천 결과 제공.</p> <p>6. 확장성: 시스템이 수백만 명의 동시 사용자를 처리할 수 있도록 확장 가능해야 함.</p> <p>7. 고가용성: 시스템이 높은 가용성을 유지하고 장애 시 빠르게 복구될 수 있어야 함.</p> <p>8. 보안: 데이터 전송 및 저장 과정에서 데이터 보안을 유지해야 함.</p>		
김소라	12	<p>이 회사에서 실시간 추천시스템을 구축할 때, 데이터 파이프라인을 흐름대로 시나리오를 기술 하시오.</p> <p>* 기술을 고려하지 않고, 비즈니스 요구사항에 맞춰 파이프라인 기술해주세요.</p>	<p>##1. 데이터 수집 사용자 행동 이벤트 수집</p> <p>1. 사용자가 넷플릭스에서 동영상을 재생하거나 검색할 때마다 해당 이벤트가 로그로 기록됩니다.</p> <p>2. Logstash는 이 로그 데이터를 실시간으로 수집하여 Apache Kafka로 전송합니다.</p> <p>##2. 데이터 처리 실시간 데이터 정제 및 변환</p> <p>1. Apache Kafka는 수집된 이벤트 데이터를 토픽(topic)에 저장합니다.</p> <p>2. Apache Flink 또는 Apache Spark Streaming이 Kafka에서 데이터를 읽어와 실시간으로 처리합니다.</p> <p>- 이벤트 데이터를 정제하여 필요한 필드만</p>	

추출하고, 사용자별 시청 기록 및 선호도 데이터를 업데이트합니다.

3. 데이터 저장

정제된 데이터 저장소

1. 정제된 데이터는 실시간으로 Amazon S3에 저장되어 데이터 레이크를 구성합니다.

2. 동시에, 주요 분석을 위한 데이터는 Apache Cassandra에 저장됩니다.

3. 사용자 프로필 및 추천 데이터는 빠른 액세스를 위해 Redis에 저장됩니다.

4. 추천 알고리즘 적용

추천 알고리즘 실행

1. Apache Flink 또는 Apache Spark Streaming에서 실시간으로 수집된 데이터를 기반으로 추천 알고리즘을 실행합니다.

- Apache Mahout을 사용하여 사용자 선호도와 시청 기록을 기반으로 추천 목록을 생성합니다.

- TensorFlow Serving을 통해 머신러닝 모델을 실시간으로 서빙하고, 더 정교한 추천을 제공합니다.

5. 실시간 추천 제공

추천 결과 제공

1. 사용자가 넷플릭스 인터페이스에 접속하면 Redis에서 실시간으로 업데이트된 추천 목록을 가져옵니다.

2. 추천 결과는 사용자

에게 실시간으로 제공
됩니다.

6. 실시간 분석 및 모니터링

시스템 모니터링 및 분
석

1. Elastic Stack을 사
용하여 실시간 로그 데
이터를 수집하고 분석
합니다.

- Logstash가 Kafka
에서 데이터를 읽어와
Elasticsearch에 저
장합니다.

- Kibana를 통해 실시
간 대시보드에서 사용
자 행동 데이터를 시각
화합니다.

2. Grafana를 사용하
여 시스템 상태 및 성
능을 모니터링합니다.

- 예를 들어, 서버의
CPU 사용량, 메모리
사용량, 네트워크 트래
픽 등을 모니터링하여
실시간으로 시각화합
니다.

7. 보안 및 인증 데이터 보안 관리

1. Apache Ranger를
사용하여 데이터 접근
제어 정책을 관리합니
다.

2. Kerberos를 통해
각 서비스 간의 인증을
수행하여 데이터 전송
의 보안을 강화합니다.

결론

이 실시간 스트리밍 데
이터 파이프라인을 통
해 넷플릭스는 사용자
행동 데이터를 실시간
으로 수집하고 처리하
여 사용자에게 맞춤형
콘텐츠를 실시간으로
추천할 수 있습니다.

			<p>이 기술 스택을 통해 OTT와 같은 회사는 실시간 추천 시스템을 구축하여 사용자 경험을 향상시키고, 데이터 처리 및 분석의 효율성을 높일 수 있습니다. 각 기술의 장단점을 고려하여 적절히 조합함으로써 시스템의 확장성, 고가용성, 보안을 모두 충족할 수 있습니다.</p> <ul style="list-style-type: none"> - 사용자 경험 향상: 사용자에게 개인화된 추천 콘텐츠를 실시간으로 제공하여 만족도를 높입니다. - 효율적인 데이터 처리: 실시간 데이터 수집 및 처리로 데이터 활용의 효율성을 극대화합니다. - 확장성: 대규모 데이터와 동시 사용자를 처리할 수 있도록 시스템을 확장 가능하게 만듭니다. - 고가용성: 시스템 장애 시에도 빠르게 복구할 수 있는 고가용성을 보장합니다. - 보안: 데이터 전송 및 저장 과정에서 강력한 보안을 유지하여 사용자 데이터를 보호합니다. 	
김소라	13	문제 1에 맞춰, 필요 기술들을 도입하여, 시스템을 구축하고, 해당 시스템 구축 효과에 대해서도 기술하세요.		

임성민	14	<p>데이터 엔지니어링과 관련된 기술사 형태의 샘플 문제는 여러 가지 주제를 포함할 수 있습니다. 아래는 데이터 엔지니어링의 핵심 개념과 기술을 다루는 몇 가지 예시 문제입니다.</p> <p>문제 1: 데이터 웨어하우스 설계</p> <p>회사는 여러 소스 시스템으로부터 데이터를 수집하여 분석하고자 합니다. 데이터 웨어하우스를 설계하려고 할 때, 다음 질문에 답하세요.</p> <p>데이터 모델링: 데이터 웨어하우스를 설계하기 위해 어떤 데이터 모델링 기법을 사용할 것인지 설명하고, 그 이유를 논의하세요.</p> <p>ETL 프로세스: ETL(Extract, Transform, Load) 프로세스를 설계하는 단계와 주요 고려 사항을 설명하세요.</p> <p>성능 최적화: 데이터 웨어하우스의 성능을 최적화하기 위해 사용할 수 있는 방법들을 제시하세요.</p> <p>문제 2: 빅데이터 처리</p> <p>회사는 소셜 미디어</p>	<p>문제 1: 데이터 웨어하우스 설계</p> <p>데이터 모델링:</p> <p>사용할 기법: 스타 스키마(Star Schema)와 스노우플레이크 스키마(Snowflake Schema) 중 스타 스키마를 사용합니다.</p> <p>이유: 스타 스키마는 단순하고 이해하기 쉬워 데이터 분석과 쿼리 성능이 뛰어납니다. 사실 테이블(fact table)과 차원 테이블(dimension table)로 구성되어 있어 데이터를 효율적으로 집계할 수 있습니다.</p> <p>ETL 프로세스:</p> <p>단계:</p> <p>추출(Extract): 소스 시스템으로부터 데이터를 추출합니다.</p> <p>변환(Transform): 데이터를 일관성 있게 변환하고 클렌징합니다.</p> <p>적재(Load): 변환된 데이터를 데이터 웨어하우스에 적재합니다.</p> <p>고려 사항:</p> <p>데이터 품질: 데이터 정확성과 일관성을 유지해야 합니다.</p> <p>성능: 대용량 데이터를 빠르게 처리할 수 있는 효율적인 프로세스를 설계해야 합니다.</p> <p>오류 처리: 오류 발생 시 대처할 수 있는 메커니즘을 마련해야 합니다.</p> <p>성능 최적화:</p>
-----	----	---	---

어 플랫폼으로부터 실시간 데이터를 수집하고 분석하여 고객 감정을 파악하려고 합니다. 이를 위해 빅데이터 처리 시스템을 설계해야 합니다.

데이터 수집: 실시간 데이터 스트리밍을 위한 데이터 수집 기술을 설명하고, 해당 기술의 장단점을 논의하세요.

데이터 저장: 수집된 데이터를 저장하기 위한 NoSQL 데이터베이스를 선택하고, 선택한 데이터베이스의 장점과 단점을 설명하세요.

데이터 처리: 실시간 데이터 처리를 위한 분산 처리 프레임워크를 선택하고, 해당 프레임워크의 주요 기능과 사용 사례를 설명하세요.

문제 3: 데이터 통합 및 품질 관리
여러 부서에서 서로 다른 형식의 데이터를 사용하는 회사에서, 데이터 통합과 품질 관리 시스템을 구축하려고 합니다.

데이터 통합: 서로 다른 형식의 데이터를 통합하기 위한 방법론을 설명

인덱싱: 적절한 인덱스를 생성하여 쿼리 성능을 향상시킵니다.

파티셔닝: 데이터를 파티션으로 분할하여 관리와 쿼리 성능을 개선합니다.

물리적 설계: 정규화와 비정규화를 적절히 사용하여 데이터 모델을 최적화합니다.

캐싱: 자주 조회되는 데이터를 캐시에 저장하여 쿼리 응답 시간을 줄입니다.

문제 2: 빅데이터 처리 데이터 수집:

기술: Apache Kafka
장단점:

장점: 높은 처리량, 확장성, 내결함성이 뛰어납니다.

단점: 초기 설정과 운영이 복잡하며, 학습 곡선이 가파릅니다.

데이터 저장:

NoSQL 데이터베이스: Apache Cassandra

장점: 높은 쓰기 처리 성능, 분산형 구조로 확장성이 좋습니다.

단점: 복잡한 쿼리 지원이 부족하고, 데이터 모델링이 어렵습니다.

데이터 처리:

프레임워크: Apache Spark

주요 기능: 대용량 데이터 처리, 실시간 스트리밍 분석, 다양한 언어 지원(Python, Scala, Java)

하고, 각 방법론의 장단점을 논의하세요.

데이터 품질 관리: 데이터 품질을 관리하기 위한 주요 지표와 이를 개선하기 위한 전략을 제시하세요.

메타데이터 관리: 메타데이터 관리의 중요성을 설명하고, 효과적인 메타데이터 관리 시스템을 설계하는 방법을 논의하세요.
문제 4: 클라우드 기반 데이터 엔지니어링

회사는 클라우드 환경에서 데이터 엔지니어링 파이프라인을 구축하려고 합니다.

클라우드 서비스: 데이터 엔지니어링을 위해 사용할 수 있는 주요 클라우드 서비스들을 비교하고, 각 서비스의 장단점을 설명하세요.

데이터 보안: 클라우드 환경에서 데이터를 안전하게 보호하기 위한 보안 전략을 제시하세요.

비용 관리: 클라우드 기반 데이터 엔지니어링 파이프라인의 비용을 효율적으로 관리하기 위한 방안을 논의하세요.

사용 사례: 로그 분석, 실시간 데이터 스트리밍 처리, 기계 학습 모델 훈련

문제 3: 데이터 통합 및 품질 관리
데이터 통합:

방법론: 데이터 레이크(Data Lake)

장점: 다양한 데이터 형식을 지원하며, 원본 데이터를 그대로 저장할 수 있습니다.

단점: 데이터 품질 관리가 어렵고, 잘못된 데이터가 혼재할 수 있습니다.

데이터 품질 관리:

주요 지표: 정확성(Accuracy), 일관성(Consistency), 완전성(Completeness), 적시성(Timeliness)

개선 전략:

데이터 프로파일링: 데이터의 특성을 파악하고, 문제점을 발견합니다.

데이터 클렌징: 오류 데이터를 수정하거나 제거합니다.

데이터 거버넌스: 데이터 관리 정책과 절차를 수립하여 품질을 유지합니다.

메타데이터 관리:

중요성: 메타데이터 관리는 데이터의 출처, 구조, 의미를 이해하고, 데이터 사용성을 높이는 데 중요합니다.

설계 방법:

중앙 메타데이터 저장

문제 5: 데이터 분석과 시각화
회사는 데이터 엔지니어링 파이프라인을 통해 수집된 데이터를 분석하고 시각화하여 경영진에게 보고하려고 합니다.

데이터 분석: 데이터 분석을 위한 주요 기법과 도구를 설명하고, 각각의 사용 사례를 논의하세요.

데이터 시각화: 데이터 시각화를 위해 사용할 수 있는 도구와 기술을 비교하고, 각각의 장단점을 설명하세요.

보고서 작성: 경영진에게 효과적으로 데이터를 전달하기 위한 보고서 작성의 원칙과 전략을 제시하세요.

이 샘플 문제들은 데이터 엔지니어링의 다양한 측면을 다루며, 실제 상황에서 발생할 수 있는 문제들을 해결하기 위한 전략과 기술을 이해하고 적용하는 데 도움을 줄 것입니다.

소 구축: 모든 메타데이터를 중앙에서 관리합니다.

자동화 도구 사용: 메타데이터 수집과 갱신을 자동화하여 일관성을 유지합니다.

사용자 접근성 향상: 사용자가 메타데이터에 쉽게 접근할 수 있도록 인터페이스를 제공합니다.

문제 4: 클라우드 기반 데이터 엔지니어링 클라우드 서비스:

비교: AWS Redshift, Google BigQuery, Azure Synapse Analytics

장단점:

AWS Redshift: 높은 성능과 확장성, 다양한 통합 도구 지원. 비용이 상대적으로 높음.

Google BigQuery: 서버리스 아키텍처로 관리가 용이, 빠른 쿼리 성능. 복잡한 쿼리 비용이 높음.

Azure Synapse Analytics: 데이터 통합 및 분석 기능 통합, 하이브리드 데이터 처리. 초기 설정이 복잡할 수 있음.

데이터 보안:

전략:

데이터 암호화: 저장 데이터와 전송 데이터 모두 암호화합니다.

접근 제어: 역할 기반 접근 제어(RBAC)와 다단계 인증(MFA)을 사용합니다.

네트워크 보안: 가상 사설망(VPN)과 방화벽을 사용하여 네트워크를 보호합니다.

비용 관리:

방안:

비용 예측 도구 사용:
클라우드 제공사의 비용 관리 도구를 사용하여 비용을 예측하고 추적합니다.

자동 스케일링: 필요할 때만 리소스를 사용하도록 자동 스케일링을 설정합니다.

예약 인스턴스 사용:
장기적으로 사용할 리소스는 예약 인스턴스를 사용하여 비용을 절감합니다.

문제 5: 데이터 분석과 시각화

데이터 분석:

기법:

회귀 분석: 변수 간의 관계를 분석하여 예측 모델을 만듭니다.

군집 분석: 데이터 포인트를 유사한 그룹으로 묶어 패턴을 발견합니다.

시계열 분석: 시간에 따른 데이터 변화를 분석하여 트렌드를 예측합니다.

도구:

Python (pandas, scikit-learn)

R

SQL

데이터 시각화:

도구:

Tableau: 사용하기 쉽

고, 다양한 데이터 소스와 연결 가능. 비용이 높음.

Power BI: Microsoft 생태계와의 통합이 우수. 사용 시 학습 필요.

Matplotlib/Seaborn (Python): 커스터마이징이 용이. 시각적 디자인이 제한적일 수 있음.

장단점:

Tableau: 직관적인 인터페이스, 고급 시각화 가능. 비용이 높음.

Power BI: 뛰어난 통합 기능, 사용 편의성. 복잡한 분석에 한계.

Matplotlib/Seaborn: 코드 기반으로 유연성 높음. 시각화 디자인이 기본적.

보고서 작성:

원칙:

명확성과 간결성: 정보를 명확하고 간결하게 전달합니다.

시각적 효과: 적절한 시각화를 사용하여 데이터의 의미를 쉽게 이해할 수 있도록 합니다.

목적에 맞는 내용: 경영진이 필요로 하는 정보를 중심으로 구성합니다.

전략:

스토리텔링 기법 활용: 데이터를 이야기 형태로 구성하여 이해를 돕습니다.

주요 지표 강조: 중요한 지표와 인사이트를 강조하여 주의를 끄니다.

			피드백 반영: 경영진의 피드백을 반영하여 보고서를 개선합니다.	
임성민	15	<p>K Software가 전 세계 판매를 목적으로 클라우드 환경에 게임 판매 사이트를 구축하려고 한다면, 다음과 같은 데이터 엔지니어링 솔루션을 설계할 수 있습니다.</p> <p>이 솔루션은 사용자 성향 및 SNS 분석을 통해 게임 추천 기능을 제공하고, 구매 유도를 목표로 합니다.</p>	<p>문제 1: 클라우드 환경에서의 데이터 엔지니어링 솔루션 설계</p> <p>데이터 모델링</p> <p>사용할 데이터 모델링 기법:</p> <p>**스타 스키마(Star Schema)**를 사용합니다.</p> <p>이유: 스타 스키마는 데이터 웨어하우스 설계에 적합하며, 간단하고 효율적으로 데이터를 집계할 수 있습니다. 특히, 게임 판매 데이터와 사용자 활동 데이터를 분석하는 데 적합합니다.</p> <p>ETL 프로세스</p> <p>ETL 프로세스 설계:</p> <p>추출(Extract): 사용자 활동, 게임 다운로드, 구매 내역, SNS 데이터 등 다양한 소스 시스템에서 데이터를 추출합니다.</p> <p>변환(Transform): 데이터 클렌징, 일관성 유지, 사용자 성향 분석을 위한 데이터 변환을 수행합니다.</p> <p>적재(Load): 변환된 데이터를 클라우드 데이터 웨어하우스 (AWS Redshift, Google BigQuery 등)에 적재합니다.</p> <p>고려 사항:</p> <p>데이터의 일관성 및 정확성 확보</p> <p>실시간 데이터 처리 요구 사항 반영</p>	

확장성과 유연성을 고려한 설계
성능 최적화
데이터 웨어하우스 성능 최적화:
인덱싱: 자주 조회되는 컬럼에 대해 인덱스를 생성하여 쿼리 성능을 향상시킵니다.
파티셔닝: 데이터를 날짜별, 사용자별 등으로 파티셔닝하여 쿼리 성능을 최적화합니다.
캐싱: 자주 조회되는 데이터를 캐시에 저장하여 응답 시간을 단축시킵니다.
물리적 설계: 비정규화된 테이블을 사용하여 읽기 성능을 향상시킵니다.

문제 2: 실시간 빅데이터 처리 시스템 설계
데이터 수집
실시간 데이터 수집 기술:
Apache Kafka를 사용합니다.
장단점:
장점: 높은 처리량, 확장성, 내결함성 제공
단점: 초기 설정과 운영의 복잡성, 학습 곡선이 가파름
데이터 저장
NoSQL 데이터베이스 선택:
Apache Cassandra를 사용합니다.
장점: 높은 쓰기 성능, 분산 구조로 높은 확장성
단점: 복잡한 쿼리 지원 부족, 데이터 모델링이 어렵습니다.

데이터 처리
실시간 데이터 처리 프레임워크:
Apache Spark를 사용합니다.
주요 기능: 대용량 데이터 처리, 실시간 스트리밍 분석, 다양한 언어 지원(Python, Scala, Java)
사용 사례: 실시간 사용자 활동 분석, 실시간 게임 추천 시스템

문제 3: 데이터 통합 및 품질 관리
데이터 통합
데이터 통합 방법론:
데이터 레이크(Data Lake)를 사용합니다.
장점: 다양한 데이터 형식 지원, 원본 데이터를 그대로 저장 가능
단점: 데이터 품질 관리가 어려움, 잘못된 데이터 혼재 가능
데이터 품질 관리
데이터 품질 관리 지표 및 전략:
주요 지표: 정확성(Accuracy), 일관성(Consistency), 완전성(Completeness), 적시성(Timeliness)
개선 전략:
데이터 프로파일링: 데이터의 특성을 파악하고 문제점 발견
데이터 클렌징: 오류 데이터를 수정하거나 제거
데이터 거버넌스: 데이터 관리 정책과 절차 수립 및 실행
메타데이터 관리

메타데이터 관리 시스템 설계:
 중요성: 데이터의 출처, 구조, 의미를 이해하고, 데이터 사용성을 높이는 데 중요
 설계 방법:
 중앙 메타데이터 저장소 구축: 모든 메타데이터를 중앙에서 관리
 자동화 도구 사용: 메타데이터 수집과 갱신을 자동화하여 일관성 유지
 사용자 접근성 향상: 사용자가 메타데이터에 쉽게 접근할 수 있도록 인터페이스 제공

문제 4: 클라우드 기반 데이터 엔지니어링
 클라우드 서비스
 클라우드 서비스 비교 및 선택:
 AWS Redshift,
 Google BigQuery,
 Azure Synapse Analytics 비교
 장단점:
 AWS Redshift: 높은 성능, 다양한 통합 도구 지원, 비용이 상대적으로 높음
 Google BigQuery: 서버리스 아키텍처로 관리 용이, 빠른 쿼리 성능, 복잡한 쿼리 비용 높음
 Azure Synapse Analytics: 데이터 통합 및 분석 기능 통합, 하이브리드 데이터 처리, 초기 설정 복잡
 데이터 보안
 클라우드 데이터 보안 전략:

데이터 암호화: 저장
 데이터와 전송 데이터
 를 모두 암호화
 접근 제어: 역할 기반
 접근 제어(RBAC) 및
 다단계 인증(MFA) 사
 용
 네트워크 보안: 가상
 사설망(VPN) 및 방화
 벽 사용
 비용 관리
 클라우드 비용 관리 방
 안:
 비용 예측 도구 사용:
 클라우드 제공사의 비
 용 관리 도구 사용
 자동 스케일링: 필요할
 때만 리소스를 사용하
 도록 자동 스케일링 설
 정
 예약 인스턴스 사용:
 장기적으로 사용할 리
 소스는 예약 인스턴스
 사용

문제 5: 데이터 분석과
 시각화
 데이터 분석
 데이터 분석 기법과 도
 구:
 기법:
 회귀 분석: 변수 간 관
 계 분석 및 예측 모델
 생성
 군집 분석: 데이터 포
 인트를 유사한 그룹으
 로 묶어 패턴 발견
 시계열 분석: 시간에
 따른 데이터 변화 분석
 및 트렌드 예측
 도구:
 Python (pandas,
 scikit-learn)
 R
 SQL
 데이터 시각화

데이터 시각화 도구 및
기술:

도구:

Tableau: 사용하기 쉽
고 다양한 데이터 소스
와 연결 가능, 비용이
높음

Power BI: Microsoft
생태계와 통합 우수,
사용 시 학습 필요

Matplotlib/Seaborn
(Python): 커스터마이
징 용이, 시각적 디자
인 제한적

장단점:

Tableau: 직관적 인터
페이스, 고급 시각화
가능, 비용 높음

Power BI: 뛰어난 통
합 기능, 사용 편의성,
복잡한 분석에 한계

Matplotlib/Seaborn:
코드 기반으로 유연성
높음, 시각화 디자인
기본적

보고서 작성

보고서 작성 원칙 및

전략:

원칙:

명확성과 간결성: 정보
를 명확하고 간결하게
전달

시각적 효과: 적절한
시각화 사용으로 데이
터의 의미 쉽게 이해
가능

목적에 맞는 내용: 경
영진이 필요로 하는 정
보 중심으로 구성

전략:

스토리텔링 기법 활용:
데이터를 이야기 형태
로 구성하여 이해 도움
주요 지표 강조: 중요
한 지표와 인사이트 강
조하여 주의 집중

			<p>피드백 반영: 경영진의 피드백 반영하여 보고서 개선</p> <p>위의 솔루션을 통해 K Software는 전 세계적으로 확장 가능한 클라우드 기반 게임 판매 사이트를 구축하고, 실시간 데이터 분석 및 사용자 성향 분석을 통해 맞춤형 게임 추천 시스템을 구현할 수 있습니다.</p>	
최인식	상황제시	<p>문제 제목 : 통신 부정이용 차단 시스템 구축</p> <p>[고객 상황]</p> <p>고객 상황 : 부정이용차단은 휴대전화 명의도용 및 자금 융통 목적의 단말기 폰깡과 보이스 피싱등으로 부터 고객 보호 목적으로 시행하고 있으며 부정이용자들의 통신사용패턴이 다양하게 변화되어가는 수법을 차단하기 위해서 노력하고 있습니다.</p> <p>Pain Point : 2개의 시스템에서 데이터를 수작업으로 추출하여 100개 이상의 차단패턴을 수작업으로 분류하고 있어서 소규모 팀에서 업무가 과중되어있는 상황</p> <p>업무 요건 :</p> <p>- 기존 온프레미스</p>		

기반의 Legacy
시스템의 데이터를
통합할 수 있는 데
이터마트 구성

- 신규 독자적 서
버 구축 필요
- Rule 기반의 패
턴 감지를 머신러
닝을 통한 패턴 예
측 및 이상징후 탐
지 시스템 개발
- 기존 Legacy 시
스템 중 A 시스템
은 실시간 연동 필
요
- WEB 상에서 패
턴결과에 대한 데
이터 관리가 가능
하도록 WebApp
개발

성공 지표 :

- 부정이용 대상
차단율
- 고객 수납율

[업무 시나리오 1]

1. 실시간 및 배치
기반의 수집 처리 :
데이터성격에 따른
실시간 및 배치로
수집
2. 수집된 데이터
에 대한 통합 및 적
재 처리
3. 적재된 데이터
소스로 Rule 기반
의 패턴분류 및 모
니터링 시스템 구
축
4. 사업부서는 구
축된 모니터링 시
스템을 통해서 결
제 차단
5. 차단된 내용에
대해서 과거 1년치

		<p>까지 관리</p> <p>6. 사업부서 이외에 다른 사용자가 접근할 수 없도록 보안시스템 구축</p> <p>7. Rule 기반과 병행할 수 있는 AI 머신러닝으로 구축</p>		
최인식	16	<p>업무시나리오 1번에서 Legacy 시스템에서 실시간 및 배치로 수집 및 대시보드까지 Data Pipe Line을 설계하세요.</p>		
최인식	17	<p>FDS 시스템을 Container 기반으로 설계하세요.</p>		
김재환	18	<p>문제1: 데이터 품질 관리와 비즈니스 인텔리전스를 위한 데이터 아키텍처 설계</p> <p>[고객 상황] 미디어 스트리밍 서비스</p> <p>'PlayStream'은 사용자 활동 데이터와 콘텐츠 소비 데이터를 활용하여 데이터 기반 비즈니스 인텔리전스 (BI) 시스템을 구축하고자 합니다.</p> <p>업무 요건:</p> <ul style="list-style-type: none"> ●실시간 사용자 활동 데이터와 콘텐츠 소비 데이터를 수집 ●대규모 데이터 저장 및 관리 ●데이터 품질 문제 해결 및 데이터 정제 		

		<p>●사용자 행동 패턴, 선호 콘텐츠 분석</p> <p>●보고서 생성 및 비즈니스 인사이트 도출</p> <p>질문:</p> <p>1.실시간 사용자 활동 데이터와 콘텐츠 소비 데이터를 수집하기 위한 데이터 파이프라인을 설계하세요. 두 가지 안을 제시하고 각 선택의 이유를 설명하세요.</p> <p>2.데이터 품질 관리 및 정제를 위해 필수적인 단계와 기술을 제시하세요.</p> <p>3.사용자 행동 패턴과 선호 콘텐츠 분석을 위한 데이터 모델과 분석 방식을 설계하세요.</p> <p>4.비즈니스 인사이트를 도출하기 위한 데이터 시각화 도구와 방법을 제안하세요.</p> <p>5.해당 아키텍처가 데이터 보안을 강화하기 위한 방법을 설명하세요.</p>		
김재환	19	<p>문제2: IoT 데이터 분석 및 예방적 유지 보수를 위한 데이터 아키텍처 설계</p> <p>[고객 상황] 산업 장비 제조업체 'Industech'는 IoT 장비로부터 발생하는 데이터를</p>		

활용하여 장비의 예방적 유지 보수를 실시하고자 합니다.

업무 요건:

- IoT 센서 데이터 실시간 수집과 처리
- 대규모 IoT 데이터 저장
- 데이터 정제 및 품질 관리
- 장비 이상 탐지 및 예측 모델 개발
- 예방적 유지 보수 스케줄링
- 실시간 알림 시스템 구축

질문:

1.IoT 장비로부터 발생하는 실시간 데이터를 수집하고 처리하기 위한 데이터 파이프라인을 설계하세요. 두 가지 안을 제시하고 각 선택의 이유를 설명하세요.

2.대규모 IoT 데이터를 저장하고 관리하기 위한 데이터베이스 아키텍처를 설계하세요.

3.데이터 정제 및 품질 관리를 위해 필요한 단계와 자동화 기술을 제시하세요.

4.장비 이상 탐지와 예측 모델을 위한 머신러닝 워크플로우를 설계하세요.

5.실시간 알림 시스템을 구축하기

		위한 아키텍처와 기술을 제안하세요.		
이유진	상황제시	<p>문제 제목 : 실손보험료 자동 청구 시스템 구축</p> <p>상황 설명</p> <p>고객의 동의 하에 국민건강보험공단의 개인별 의료정보를 보험기관에 연계하여 급여/비급여 의료비에 대한 실손보험 청구료를 계산하여 자동 지급될 수 있도록 시스템 구축</p> <ul style="list-style-type: none"> - 각 보험사에서 요구하는 보험료 산정을 위한 의료정보 표준모델 구축 - 고객개인정보 연동에 따른 개인정보 암호화 시스템 구축 - 개인정보 전송에 따른 전송이력 관리 - 의료정보 전송기록 관리를 위한 플랫폼 구축 		
이유진	20	개인의 의료정보에 대한 데이터 표준 모델을 구축할 때 고려해야 할 사항을 기술하시오.		
이유진	21	개인의 의료기록을 수집하여 각 보험사로 제공하기 위		

		해 사용해야 할 데이터 파이프라인을 설계하고 이를 관리할 수 있는 모니터링 방안을 제시하시오.	
--	--	--	--

실무인터뷰 질의내용:

Q:인터넷에 있는 날씨 정보를 가져와서 온디멘드 환경에서 사용하려고 합니다.해당 데이터를 운용하는 효율적인 방법으로 어떤 설계를 하면 좋을까요?

A: 해당 데이터를 운용하는 효율적인 방법으로는 웹 크롤링을 통해 인터넷에서 날씨 정보를 가져오는 것이 좋습니다.

데이터를 실시간으로 사용하고자 한다면, BIBOT을 이용하여 사용자의 질의를 통해 웹 크롤링 작업을 실행하고,

가져온 데이터를 저장하여 온디멘드 환경에서 사용할 수 있도록 설계할 수 있습니다.

웹 크롤링 시 주의해야 할 사항은 다음과 같습니다:

1. 법률과 사용 약관 준수: 웹 사이트의 크롤링은 해당 사이트의 소유자 또는 운영자의 허가가 필요할 수 있습니다. 웹 사이트의 사용 약관을 확인하여 크롤링이 허용되는지 확인하십시오. 법률적으로 금지되거나 제한된 경우 크롤링을 시도하면 법적 문제에 직면할 수 있습니다.
2. 로봇 배제 표준 준수: 로봇 배제 표준은 웹 사이트 소유자가 크롤러에 대한 접근을 제어하는 규칙입니다. 웹 사이트의 `robots.txt` 파일을 확인하여 크롤러가 허용되지 않은 페이지를 방문하지 않도록 하십시오. 로봇 배제 규칙을 무시하는 경우 웹 사이트에 대한 액세스가 차단될 수 있습니다.
3. 요청 속도 제한: 크롤러가 너무 빠르게 웹 사이트에 액세스하면 서버 과부하로 인해 웹 사이트가 다운될 수 있습니다. 적절한 요청 속도를 유지하고 웹 사이트의 서버 부하를 줄이기 위해 적정 시간 간격을 두고 크롤링하세요.
4. 도덕적인 사용: 크롤링한 데이터를 사용할 때에는 도덕적이고 법적인 한계를 준수해야 합니다. 예를 들어, 개인정보 보호 및 저작권을 준수해야 합니다. 크롤링한 데이터를 무단으로 상업적인 목적으로 사용하거나 다른 사람들을 사칭하는 등의 행위는 피해야 합니다.
5. 웹 사이트에 대한 부하 생성 최소화: 웹 크롤러가 큰 부하를 발생시키면 웹 사이트의 성능에 영향을 줄 수 있습니다. 크롤러가 필요한 정보에만 액세스하고 필요하지 않은 자원을 요청하지 않도록 하는 것이 좋습니다. 또한, 네트워크 트래픽을 줄이기 위해 크롤링한 데이터를 적절하게 압축하거나 최적화하는 것도 중요합니다.

6. 크롤러 식별: 웹 사이트는 로봇 배제 표준 또는 기타 수단을 사용하여 크롤러를 식별할 수 있습니다. 헤더 또는 사용자 에이전트 문자열을 포함하여 크롤러를 식별하거나 웹 사이트의 API를 사용하여 액세스하는 것이 좋습니다. 그러면 웹 사이트의 소유자가 크롤링 활동을 모니터링하거나 제어할 수 있습니다.

Q:당신은 SI구축 전문가입니다. 대량의 데이터를 수집하여 데이터레이크를 구축하려고 합니다. 어떤 절차로 진행하면 될지 단계별로 알려주세요.

A: 데이터 레이크를 구축하기 위해서는 다음과 같은 절차를 단계별로 진행할 수 있습니다:

1. 데이터 수집 계획 수립: 먼저 어떤 종류의 데이터를 수집할 것인지 정하고, 해당 데이터를 수집하기 위한 계획을 수립합니다. 데이터의 소스와 형식, 수집 주기 등을 고려하여 계획을 세웁니다.
2. 데이터 수집 구현: 수집 계획에 따라 데이터를 수집하기 위한 코드나 도구를 구현합니다. 데이터 소스와 연결하여 데이터를 추출하고, 필요한 가공과정을 거친 뒤 수집된 데이터를 저장합니다. 이때, 데이터의 크기와 속도에 따라 효율적인 수집 방법을 선택합니다.
3. 데이터 저장: 수집된 데이터를 안정적으로 저장하기 위한 저장소를 선택합니다. 데이터 레이크 환경에서는 보통 분산 파일 시스템 또는 객체 스토리지를 사용합니다. 저장소에는 데이터의 크기와 성격을 고려하여 최적의 설정을 적용합니다.
4. 데이터 적재: 수집된 데이터를 저장소로 적재합니다. 이때, 데이터의 형식과 구조를 고려하여 적절한 방법으로 데이터를 저장합니다. 예를 들어, 대용량의 정형 데이터는 병렬로 분산 처리할 수 있는 방식으로 적재할 수 있습니다.
5. 데이터 처리: 저장된 데이터를 가공하고 분석하기 위한 처리 과정을 수행합니다. 이 단계에서는 개발환경과 알고리즘을 활용하여 원하는 형태로 데이터를 가공합니다. 데이터 처리 결과는 다시 데이터 레이크에 저장되거나 다른 시스템으로 전달될 수 있습니다.
6. 데이터 접근 및 분석: 데이터 레이크에 저장된 데이터에 접근하여 분석 작업을 수행합니다. 이를 위해 데이터 레이크에 대한 쿼리 기능이나 분석 도구를 활용할 수 있습니다. 데이터의 높은 확장성과 유연성을 활용하여 다양한 방법으로 데이터를 분석하고 인사이트를 도출합니다.

이러한 절차를 따르면 대량의 데이터를 수집하여 데이터 레이크를 구축할 수 있습니다.

Q: 당신은 AI전문 컨설턴트입니다. 사용자 질의에 답변해주세요.

카드사에서 사기적발용 AI프로그램을 수행하고 있습니다. 해당 기능의 성능을 향상시키기 위해 운영자가 추가로 할 수 있는 작업들을 알려주세요.

A: AI 프로그램의 성능을 향상시키기 위해 운영자는 다음과 같은 작업들을 수행할 수 있습니다:

1. 더 많은 데이터 수집: AI 프로그램은 데이터에 기초하여 학습하므로, 더 많은 사기 거래에 관한 데이터를 수집할수록 더 정확한 예측이 가능합니다. 운영자는 사기적인 거래에 대한 데이터 수집을 늘림으로써 AI의 성능을 높일 수 있습니다.
2. 레이블링: 수집한 데이터를 레이블링하여 AI 알고리즘이 이를 보다 잘 이해할 수 있도록 도와줍니다. 레이블링은 사기적인 거래와 정상적인 거래를 분류하는 작업을 포함합니다. 이렇게 레이블링된 데이터는 AI 모델의 학습에 사용됩니다.

3. 실시간 모니터링: AI 프로그램은 사기적인 거래를 실시간으로 감지하기 위해 카드 사의 시스템과 연결되어야 합니다. 운영자는 AI 프로그램이 시스템과 원활하게 통신할 수 있도록 실시간 모니터링 기능을 구현할 수 있습니다.
4. 효율적인 알고리즘 개발: AI 모델의 성능을 향상시키기 위해 운영자는 알고리즘을 계속해서 개선해야 합니다. 이를 위해 운영자는 최신의 AI 기술과 방법론에 대한 연구를 지속적으로 수행하고, 이를 AI 프로그램에 반영합니다.
5. 성능 테스트 및 평가: 운영자는 AI 프로그램의 성능을 정기적으로 테스트하고 평가해야 합니다. 이를 통해 프로그램이 제대로 작동하는지 확인하고 개선할 부분을 찾을 수 있습니다. 성능 테스트는 실제 데이터를 사용하여 모의 시나리오를 구축하고 프로그램의 예측 정확도를 측정하는 과정을 포함합니다.

이러한 작업들을 통해 운영자는 AI 프로그램의 사기적발용 기능을 향상시킬 수 있고, 정확성과 효율성을 높일 수 있습니다.

Q: Data Science 기술사 시험에서 머신러닝 관련된 구체적 질문과 답변 예시를 알려줘

A: 머신러닝 관련된 구체적인 질문과 답변 예시를 알려드리겠습니다.

질문: 머신러닝은 무엇인가요?

답변: 머신러닝은 컴퓨터 시스템이 데이터를 분석하고 패턴을 학습하여, 인간의 개입 없이 스스로 학습하고 예측하는 인공지능 기술입니다.

질문: 머신러닝의 주요 분야는 무엇인가요?

답변: 머신러닝은 주로 예측, 분류, 군집화, 회귀 등의 작업에 사용됩니다. 예를 들어, 스팸 이메일 필터링, 음성 인식, 이미지 분류, 주가 예측 등 다양한 분야에서 활용됩니다.

질문: 머신러닝 알고리즘에는 어떤 종류가 있나요?

답변: 머신러닝 알고리즘에는 지도학습(Supervised Learning), 비지도학습(Unsupervised Learning), 강화학습(Reinforcement Learning) 등이 있습니다. 지도학습은 레이블이 있는 데이터로 모델을 학습시키는 방법이고, 비지도학습은 레이블이 없는 데이터에서 패턴을 찾아내는 방법입니다. 강화학습은 시행착오를 통해 보상을 최대화하는 방법을 학습하는 방법입니다.

질문: 머신러닝을 위해 어떤 도구나 라이브러리가 사용되나요?

답변: 머신러닝을 위해 Python과 R 같은 프로그래밍 언어가 주로 사용됩니다. 또한, Scikit-learn, TensorFlow, Keras, PyTorch와 같은 라이브러리와 프레임워크도 널리 사용됩니다. 이러한 도구와 라이브러리를 사용하여 데이터 전처리, 모델 학습, 모델 평가 등을 수행할 수 있습니다.

Q: MLOps에 대해서 설명하고, 장단점을 같이 말하십시오

A : MLOps는 ML애플리케이션 개발(Dev)을 ML시스템 배포 및 운영(Ops)과 통합하는 ML 문화 및 관행

프로덕션 환경에서 기계학습 모델을 안정적이고 효율적으로 배포하고 유지하는 것을 목표로 하는 패러다임

모델생성, 오케스트레이션 및 배포 통합부터 상태, 진단, 거버넌스 및 비즈니스 지표에 이르기까지 전체 수명주기에 적용됨

모델Ops의 하위 개념 > MLOps

ML, DevOps, DE(Data Engineering) 세가지 기술의 교차지점을 MLOps라고 볼 수 있음

프로덕션 품질 개선, 릴리즈 속도 가속, 패치 적용, 높은 고객만족도 달성이 목표

MLOps 플랫폼 : 실시간 공동작업과 반복 데이터 탐색기능을 지원하여 실험추적, 모델관리, 피처엔지니어링 등을 간편하게 수행하게 도와줌

MLOps 조건 : 지속적 통합(CI), 지속적 배포/제공(CD), 지속적 학습(CT)

데이터 엔지니어링

빅데이터 플랫폼 구축/ 운영관련

~~ 고객의 환경에서, 네트워크 제약, 비용 등 이슈로 6개 사이트 동일한 빅데이터 플랫폼 구축하고 있습니다. 일부 구축, 일부 구축 중인 상태입니다.

○ 구성 요소 : 수집/워크플로우, 저장소 하둡, 데이터 처리 HIVE, GREENPLUM, 마트 DB, 시각화 Spotfire, 앱가상화, VM 환경(sandbox용), 분석 플랫폼 등

시각화를 통한 정형/ 비정형 리포트를 제공하고 있음

○ 글로벌 데이터 수집/처리, 스토리지, data mart, 분석환경(앱 가상화), 시각화 툴, 분석툴, VM Sandbox, 멀티 instance 구성

○ data source - data staging, data 프로세싱/store/ analytics / 연동 등 전체적으로 구성을 제공해주고

ETL 단위 job 개발시 (파란색선) 에 대한 솔루션 선정 방안 제시

- 1안) 다양한 원천 데이터를 지원하는 단일 ETL 솔루션(EX. DATA sTAGE, Informatica, teraStream등) 으로 단위 job 개발

- 2안)

안을 선정하고, 선정한 사유 등 등 기술

문제 1] 하기 시나리오를 참조하여, 문제에 답하시오.

시나리오 01. A사 빅데이터플랫폼 구축 및 운영 (1/2)

[고객사 및 생산시스템 현황]

- A사는 global 6개의 사이트에 생산시설들이 있다. (사이트: 한국, 중국, 폴란드, 미국, 캐나다, 인도네시아 HQ:한국)

- 생산관련 운영시스템들은 생산에 직접적이거나, 데이터사이즈가 큰 시스템들은 각 사이트에 multi instance로 위치하며,

생산에 덜 민감하고 데이터 사이즈가 작은 시스템들은 global single instance로 한국에 위치한다.

- 사이트별 multi instance: MES(생산), IoT(측정), SPC(검사), RMS(레시피), FDC(이상 감지)
- global single instance: QMS(품질), PLM(제품), MMD(기준정보), WMS(창고,물류)
- 6개 사이트의 생산시스템에서 발생하는 데이터의 전체 사이즈는 20PB 이상이다.(5년 기준)
- 대부분의 제품(95%)은 각 사이트에서 모든 공정을 완료하나, 일부 제품(5%)은 타사이트의 반제품을 물류 이동하여 제품을 생산하기도 한다.
- 원천데이터 현황 및 업무요건: 사이트별 multi instance 시스템의 경우, 전체 범위의 90%는 테이블 및 업무 로직이 동일하나, 10%는 테이블

및 업무 로직이 사이트별로 일부 상이함

[빅데이터플랫폼 현황]

- A제조사는 최근 생산 불량분석을 위한 빅데이터플랫폼을 on-prem(private 클라우드)방식으로 구축 중이다.

- 네트워크 제약, 비용 등의 이슈로, 6개 사이트에 동일한 빅데이터플랫폼을 구축 중이다.(일부 사이트는 오픈하였고, 일부사이트는 구축 중임)

- 빅데이터플랫폼 구성요소: 수집/워크플로우(airflow), 저장소(Hadoop), 데이터처리(hive, greenplum), 마트DB(greenplum),

시각화(spotfire), 앱가상화, VM환경(sandbox용), 분석플랫폼(MLDL)

- 빅데이터플랫폼을 통해 제공하고자 하는 서비스는 시각화를 통한 정형리포트 및 비정형분석, VM sandbox기반의 고급분석 등이다.

- 분석 요구사항: 각 사이트별 분석(90%), 사이트간 교차분석(10%)

시나리오 01. A사 빅데이터플랫폼 구축 및 운영 (2/2)

앱가상화: 데스크톱 가상화 처럼 애플리케이션 단위의 가상화로, 사용자 로컬환경이 아닌 앱가상화

환경에서 앱을 수행할 수 있음. A사이트의 데이터를 분석하려면, A사이트의 앱가상화 환경에 접속하여

분석하면 됨

- 주요 폴란드/ 중국 / 그외 나머지 나라는 사이트는 한국에 구축
- 각국의 사용자들은 나라간 교차분석 필요

폴란드에서 발생한 불량 원인 분석 업무

중국에서 폴란드로 이동한 LOT 영향도 분석 업무요건

중국에서 발생한 불량 원인 분석 업무

- 각국의 DataLake (스토리지)간 교차생산에 따른 관련 데이터 이동이 필요

첨부된 [시나리오 01. A사 빅데이터플랫폼 구축]에서, 통합 플랫폼구축이 아닌, 사이트별 분리구축으로 인해, 발생할 수 있는

파생 이슈들을 아래 정리된 항목 중에서 선택하여 기술하시고, 각 이슈를 해결(또는 완화)할 수 있는 방안들을 제시 하시오.

1. 파생 이슈를 기술

- 사용자 활용(분석) 관점 - 유지보수 관점

< 선택 >

- 메타데이터 관점 - 데이터연계(I/F, ETL) 관점 - 데이터 정합성 관점
- 기타 관점(단, 인프라,성능,비용 관점은 제외)

문제 제목 : 온라인 쇼핑몰 데이터 분석 기반 마케팅 자동화

[고객 상황]

고객 상황 : 글로벌 패션 브랜드 '스타일 잇' 은 온라인 쇼핑몰 운영 데이터를 활용하여 고객 맞춤형 마케팅 전략을 수립하고자 한다.

Pain Point : 막대한 구매 데이터와 고객 행동 데이터를 효율적으로 분석하고 활용하기 어려움

업무 요건 :

- 다양한 데이터 소스(웹로그, 거래 데이터, CRM 등)를 통합하고 처리할 수 있는 데이터 파이프라인 구축
- 고객 구매 패턴, 인구 통계, 선호도 분석을 위한 머신러닝 모델 개발
- 실시간 고객 추천 및 맞춤형 프로모션 자동화 시스템 구축
- 데이터 기반 마케팅 캠페인 효과 측정 및 최적화

성공 지표 :

- 구매 전환율 증대
- 고객 유지율 향상
- 마케팅 투자 수익률 (ROI) 최대화

[업무 시나리오 1]

- 1. 다양한 데이터 소스 통합: 웹로그, 거래 데이터, CRM 등 다양한 데이터 소스를 통합하고 처리할 수 있는 데이터 파이프라인 구축
- 2. 실시간 데이터 처리: 구매, 검색, 클릭 등 사용자 행동 데이터를 실시간으로 처리 및 분석
- 3. 데이터 저장 및 관리: 대규모 데이터 저장 및 관리를 위한 분산 데이터베이스 구축
- 4. 데이터 품질 관리: 데이터 누락, 오류, 불일치 등 데이터 품질 문제 해결 및 데이터 정제
- 5. 고객 분석: 고객 구매 패턴, 인구 통계, 선호도 등을 분석하기 위한 머신러닝 모델 개발
- 6. 실시간 추천: 고객 맞춤형 상품 추천을 위한 실시간 추천 시스템 구축
- 7. 마케팅 자동화: 고객 세분화, 타겟팅, 캠페인 자동화 등 마케팅 자동화 시스템 구축
- 8. A/B 테스트: 다양한 마케팅 전략의 효과를 비교 분석하기 위한 A/B 테스트 플랫폼 구축
- 9. 데이터 시각화: 분석 결과를 시각화하고 이해하기 쉽게 제시하기 위한 데이터 시각화 도구 활용
- 10. 보안 및 개인정보 보호: 고객 데이터 보안 및 개인정보 보호를 위한 강력한 보안 시스템 구축

[문제1]

업무 시나리오 1에서 ①②에 대해 수집 파이프라인을 2가지 안으로 설계하세요. 각 각의 안에 대해서 선택한 사유를 기술하세요.

[문제2]

문제 1에서 설계된 수집 레이어 기반으로 ③④을 기준으로 data warehouse/data lake 레이어를 설계하세요.