



# 데이터베이스 소개

## Database Fundamentals

발표자 이름

날짜

© 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved.

데이터베이스 소개를 시작하겠습니다.

## 교육 내용

### 이 강의의 핵심

배울 내용은 다음과 같습니다.

- 관계형 데이터베이스와 비관계형 데이터베이스의 차이점을 포함하여 데이터베이스의 다양한 측면 파악
- 데이터베이스 관리 시스템(DBMS)의 목적과 기능 설명
- 관계형 데이터베이스와 상호 작용하는 다양한 방법 파악
- 새로운 데이터 기술 인식

주요 용어:

- 데이터
- 데이터베이스
- 데이터 모델
- 관계형 데이터 모델
- 관계형 데이터베이스
- 비관계형 데이터베이스
- 데이터베이스 관리 시스템(DBMS)



이 모듈에서 학습할 내용은 다음과 같습니다.

- 관계형 데이터베이스와 비관계형 데이터베이스의 차이점을 포함하여 데이터베이스의 다양한 측면 파악
- 데이터베이스 관리 시스템(DBMS)의 목적과 기능 설명
- 관계형 데이터베이스와 상호 작용하는 다양한 방법 파악
- 새로운 데이터 기술 인식

# 데이터 및 데이터베이스

## 데이터란 무엇입니까?

- 데이터는 정보의 원시 비트 및 조각입니다.
  - 이미지, 단어, 전화 번호는 데이터의 예입니다.

## 데이터베이스란 무엇입니까?

- 데이터베이스는 **테이블**이라고 하는 파일로 구성된 데이터의 모음입니다.
  - 테이블은 데이터에 액세스하고 데이터를 관리 및 업데이트하는 논리적 방법입니다.

샘플 테이블: 동물 선호도

개	고양이	말	기타
17	12	7	4

데이터는 정보의 원시 비트 및 조각입니다. 휴대폰이나 컴퓨터 모니터에서 볼 수 있는 이미지, 단어, 전화 번호를 구성하는 정보 비트가 데이터의 예입니다.

데이터베이스란 무엇입니까?

데이터베이스는 **테이블**이라고 하는 파일로 구성된 데이터 비트의 모음입니다. 테이블은 데이터에 액세스하고 데이터를 관리 및 업데이트하는 논리적 방법입니다.

데이터베이스에서 데이터는 그림, 그래픽, 이미지, 음성/화상 레코딩 등 다른 형식으로 나타낼 수도 있습니다.

## 심층 탐구: 데이터 모델 및 E.F. Codd

### 데이터 모델

- 데이터 모델은 데이터베이스의 **논리적 구조**입니다.
- 데이터 모델은 데이터를 저장하고 구성하는 방법을 결정하는 데 사용됩니다.

### E.F. Codd 관계형 모델

- **관계형 모델(RM)**은 수학자인 Edgar F. Codd 박사가 IBM에서 1960년대 후반에 개발한 데이터 모델입니다.
  - 대규모 데이터 처리를 향상하기 위해 개발됨
  - **관계 대수**라는 수학 영역을 기반으로 함

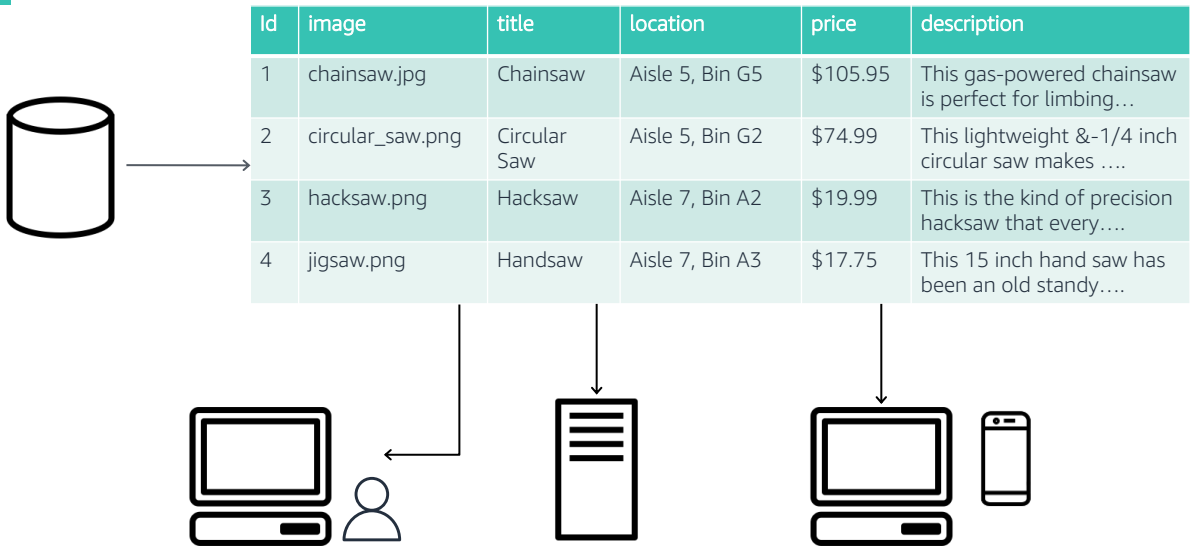


aws re/start

데이터 모델은 데이터베이스의 **논리적 구조**를 나타냅니다. 데이터 모델은 데이터를 저장하고 구성하는 방법을 결정하는 데 사용됩니다.

관계형 모델은 수학자인 Edgar F. Codd 박사가 IBM에서 1960년대 후반에 개발한 데이터 모델입니다. 대규모 데이터 처리를 향상하기 위해 개발되었습니다. 관계형 모델은 관계 대수라는 수학 영역을 기반으로 합니다.

## 관계형 모델



5

aws re/start

이 예는 관계형 모델입니다. 모델을 보면, **Products**라는 데이터베이스로 시작합니다. Products 데이터베이스의 내부에는 테이블이 있습니다. **Products** 테이블에는 여러 레코드(행이라고도 함)와 필드(열이라고도 함)가 있을 수 있습니다. 각 레코드 또는 행에는 제품에 대한 정보가 포함됩니다. 예를 들어 행에는 제품의 이름, 제품의 비용 또는 창고의 위치가 포함될 수 있습니다. 또한 재고 관리 단위(SKU) 또는 ID, 제품의 재고 번호가 포함될 수도 있습니다.

데이터베이스는 **백엔드**, 즉 대부분의 데이터 중심 애플리케이션, 데이터 중심 웹 사이트, 데이터 중심 모바일 앱의 배후에 존재합니다.

## 소규모 또는 분산



데이터베이스는 소규모로 운영할 수 있습니다. 한 대의 컴퓨터를 실행하고 단일 사용자를 지원할 수 있습니다.

그 외 데이터베이스는 회사 네트워크의 여러 시스템과 위치에 걸쳐 분산되거나 수백만 명의 사용자를 지원하는 클라우드 기반 데이터베이스일 수 있습니다.

## 2가지 데이터베이스 유형

### 관계형

- **관계형 데이터베이스**는 서로 간에 사전 정의된 관계가 있는 데이터 항목의 모음입니다.
- 데이터의 구조에 대한 고정된 정의가 필요합니다.

### 비관계형

- **비관계형 데이터베이스**는 관계형 모델을 따르지 **않는** 데이터베이스입니다.
- **비관계형 데이터베이스**는 데이터의 구조에 대한 고정된 정의가 필요하지 **않습니다**.

**관계형 데이터베이스**는 서로 간에 사전 정의된 관계가 있는 데이터 항목의 모음입니다. 관계형 데이터베이스는 **데이터의 구조에 대한 고정된 정의**가 필요합니다.

**비관계형 데이터베이스**는 관계형 모델을 따르지 않는 데이터베이스입니다.

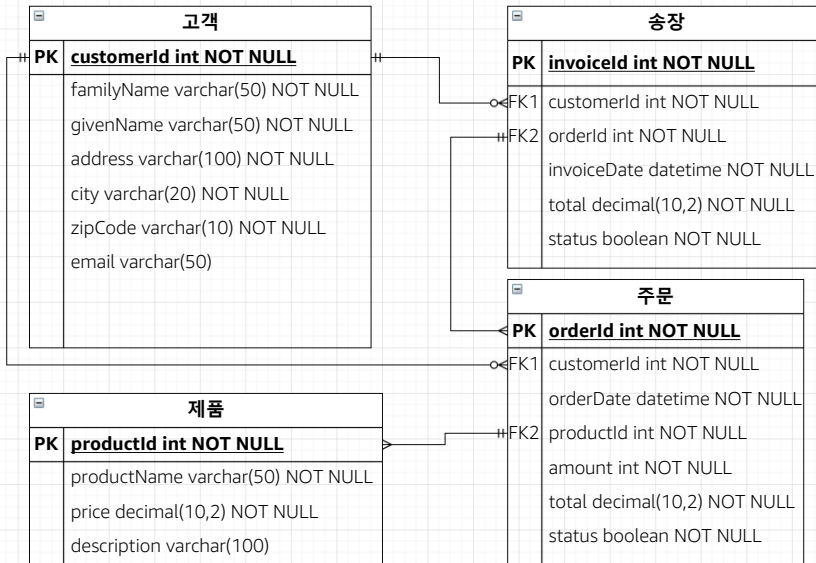
다음으로, 이러한 모델의 몇 가지 차이점에 대해 알아보겠습니다.

## 관계형 데이터베이스

이제 관계형 데이터베이스에 대해 자세히 알아보겠습니다.



## 관계형 데이터베이스



9

관계형 데이터베이스 설계에서:

- **관계형 데이터베이스는 종종 정형 쿼리 언어(SQL) 데이터베이스라고도 하며 여러 테이블에 데이터를 분산시킬 수 있는 데이터베이스입니다.**
- 각 테이블은 열과 행으로 구성됩니다. 각 열(또는 필드)에는 숫자, 문자열 또는 날짜와 같은 특수한 종류의 데이터가 들어 있습니다. 각 행은 테이블의 열에 대한 값의 집합입니다.
- 다음을 위해 열 또는 테이블에 제한을 설정할 수 있습니다.
  - 테이블에 올바른 데이터를 삽입하기 위해
  - 테이블을 연결하기 위해

- 테이블, 필드 및 제한의 정의는 데이터베이스 스키마에 그룹화됩니다.

## 관계형 데이터베이스(계속)

customerId	familyName	givenName	address	city	zipCode	email
1	Rosalez	Alejandro	123 Any Street	Any Town	12345	a.rosalez@anymail.com
2	Doe	Jane	100 Main Street	AnyTown	54321	

orderId	customerId	orderDate	productId	amount	total	status
1345	2	12/05/2020	2567	2	371.8	1
1346	2	03/02/2021	3492	1	349.8	1

productId	productName	price	description
2567	Snowboard x	185.9	Kids Snowboard
3492	Snowboard z	349.8	Adults Snowboard

## 비관계형 데이터베이스

이제 비관계형 데이터베이스에 대해 자세히 알아보겠습니다.

## 비관계형 데이터베이스

JSON 문서로 저장된 데이터 예

```
[
  {
    "CustomerId": 1234,
    "FirstName": "John",
    "LastName": "Doe",
    "orders": [
      {
        "ProductId": "1",
        "Type": "Book",
        "Title": "Odyssey",
        "Author": "Homer",
        "Year": 1871
      },
      {
        "ProductId": "3",
        "Type": "Movie",
        "Title": "The Kid",
        "Genre": ["Comedy", "Drama"],
        "Director": "Chaplin"
      }
    ]
  }
]
```

유연한 스키마의 예: 스키마는 항목별로 정의됨

Product id	Type			
1	Book	Odyssey	Homer	1871
2	Album	6 Partitas	Bach	
2	Album : Track	Partita No.1		
3	Movie	The Kid	Comedy	Chaplin

12

aws re/start

흔히 NoSQL 데이터베이스라고 하는 비관계형 데이터베이스는 데이터를 저장하기 위해 테이블 구조가 필요하지 않은 데이터베이스입니다.

객체를 여러 테이블(고객, 주문 등)로 나누는 대신 고정된 스키마를 사용하여 제한 없이 유연한 스키마로, 또는 JSON 또는 XML 형식의 문서로 저장할 수 있습니다.

NoSQL 데이터베이스는 데이터에 액세스하고 이를 관리하기 위해 다양한 데이터 모델을 사용합니다. 이러한 데이터베이스 유형은 대용량 데이터 볼륨, 짧은 대기 시간, 유연한 데이터 모델이 필요한 애플리케이션에 최적화되었습니다. 이러한 요구 사항은 다른 데이터베이스에서 사용되는 데이터 일관성에 대한 일부 제한을 완화하여 달성할 수 있습니다.

# 관계형 데이터베이스와 비관계형 데이터베이스 비교

## 관계형 데이터베이스

- **용례:** 전자 상거래, 고객 관계 관리(고객과의 상호 작용 관리), 비즈니스 인텔리전스 도구(재무 보고, 데이터 분석)
- **관계형 데이터베이스 예:** MySQL, Amazon Aurora, PostgreSQL, Microsoft SQL Server, Oracle
- **관계형 데이터베이스를 사용해야 하는 주요 이유:**
  - SQL이 기본적으로 지원됨
  - 데이터 무결성 보장
  - 트랜잭션 지원

SQL

데이터 무결성

트랜잭션

이 슬라이드는 관계형(SQL) 데이터베이스를 사용할 때의 장점과 단점을 보여줍니다.

- **장점**
  - 알려져 있고 신뢰할 수 있는 기술
  - 제한이 데이터베이스에 쓰인 데이터의 정확성을 보장함(쓰기 시 스키마)
  - 복잡한 쿼리를 쉽게 쓸 수 있음
  - SQL 언어는 잘 알려지고 문서화된 모든 관계형 데이터베이스의 기본 언어임
  - 트랜잭션이 잘 지원됨: 트랜잭션은 테이블에서 모두 성공해야 하는 일련의 연산입니다. 한 문제가 발생하면 데이터베이스는 데이터가 여전히 일관된 저장 지점으로 롤백할 수 있습니다. 이것의 일반적인 경우는 장바구니 체크아웃입니다. 결제가 성공하면 주문 테이블과 재고 테이블이 함께 업데이트되어야 합니다.

- 단점
  - 수직 크기 조정(더 빠른 CPU의 추가 RAM과 같은 리소스를 서버에 추가)을 사용하여 비용이 높음
  - 스키마가 고정되어 애플리케이션을 크게 확장할 수 없음

# 관계형 데이터베이스와 비관계형 데이터베이스 비교

## 비관계형(NoSQL) 데이터베이스

- 용례: 사기 감지, 사물 인터넷(IoT), 소셜 네트워크
- NoSQL 데이터베이스 예: AWS DynamoDB, AWS Redshift, MongoDB, Apache HBase
- NoSQL 데이터베이스를 사용해야 하는 주요 이유:
  - 유연한 모델
  - 대량의 데이터를 처리할 수 있음
  - 크기 조절이 용이

유연성

확장 가능

고성능

이 슬라이드는 비관계형(NoSQL) 데이터베이스를 사용할 때의 장점과 단점을 보여 줍니다.

- 장점
  - 스키마가 유연하며 쉽게 발전시킬 수 있음
  - 다양한 유형의 대용량 데이터를 빠르게 저장하고 검색하는 데 적합
  - 수평 크기 조정(데이터베이스가 여러 서버에 분산됨)으로 수직 크기 조정에 비해 비용이 저렴
  - 계층 구조 데이터를 저장하는 데 적합
- 단점:
  - 비교적 새로운 기술
  - 데이터 무결성이 보장되지 않으며 데이터를 읽는 애플리케이션에 의해 충족되어야 함(읽기 시 스키마)
  - 복잡한 쿼리 또는 트랜잭션 애플리케이션에 적합하지 않음



- 대표적인 애플리케이션: 사기 감지, 사물 인터넷(IoT), 소셜 네트워크...
- 대표적인 NoSQL 데이터베이스: AWS DynamoDB, AWS Redshift, MongoDB, Apache HBase

## 데이터베이스 관리 시스템(DBMS)

저장된 데이터를 어떻게 관리합니까?

데이터베이스 관리 시스템(DBMS)은 데이터베이스 기능을 제공하는 소프트웨어입니다.

# 데이터베이스 관리 시스템(DBMS)

## 데이터베이스 관리 시스템(DBMS)

데이터베이스 관리 시스템(DBMS)은 데이터베이스 기능을 제공하는 소프트웨어입니다.

2개의 유형은 다음과 같습니다.

Microsoft Access와 같은 **단일 사용자** DBMS 애플리케이션

Oracle Database, Microsoft SQL Server, MySQL, IBM Db2와 같은 **다중 사용자** DBMS 애플리케이션

저장된 데이터를 어떻게 관리합니까?

데이터베이스 관리 시스템(DBMS)을 사용하여 저장된 데이터를 관리합니다.

데이터베이스 관리 시스템(DBMS)은 소프트웨어 또는 **서비스 형태의 데이터베이스(DBaaS)**입니다. 주로 데이터베이스를 생성하거나 데이터베이스에 데이터를 삽입하거나 데이터베이스에서 데이터를 저장, 검색, 업데이트 또는 삭제하는 데 사용됩니다.

DBMS는 데이터베이스를 생성하는 도구를 제공합니다. 기업은 DBMS에서 데이터베이스를 설계 및 모델링하고 데이터를 관리하기 위한 명령을 실행할 수 있습니다. 사용자에게 데이터베이스의 데이터와 소프트웨어 간의 인터페이스를 제공합니다.

## 데이터베이스 관리 시스템(DBMS) (계속)

### 위치



**온프레미스(데이터 센터)** – 데이터는 조직이 소유한 컴퓨터와 네트워크에 저장됩니다. 이러한 컴퓨터와 네트워크는 조직의 자체 데이터 센터에 위치합니다.



**클라우드 내(가상화된 데이터 센터)** – 데이터는 Amazon Web Services(AWS)와 같은 클라우드 공급업체가 소유한 원격 데이터 센터 등과 같이 조직 외부에 위치합니다.

기존의 데이터베이스 관리 방법으로 기업은 DBMS를 사용하여 온프레미스 데이터 센터 또는 가상 환경에 데이터베이스를 생성하고 구축할 수 있습니다. 기존의 DBMS를 사용하면 비용과 인프라가 문제가 될 수 있습니다.

DBaaS는 클라우드에 있기 때문에 가상화된 데이터 센터입니다. 따라서 기존 DBMS의 여러 문제를 해결합니다.

# 서비스 형태의 데이터베이스(DBaaS)



## 클라우드 기반 데이터베이스에 대한 몇 가지 핵심 사항

- **데이터베이스 서버가 제3자 공급업체에 의해 호스트됨**
  - 로컬 네트워크에서 호스트되지 않고 제3자 데이터 센터에서 호스트되며 인터넷(클라우드)을 통해 액세스
- **완전관리형**
  - 예를 들어 일부 AWS 데이터베이스에서는 서버 프로비저닝, 패치, 설정, 구성, 백업 또는 복구와 같은 데이터베이스 관리 태스크가 필요 없음
- **빠른 속도**
  - AWS와 같은 서비스를 사용하여 데이터 센터에 대용량 스토리지 및 처리 능력 제공
- **비용 절감**
  - 서버 설치 및 유지 관리에 드는 비용 절감

데이터베이스 서버는 제3자 데이터 센터에서 호스트됩니다. 로컬 네트워크에서 호스트되지 않고 인터넷(클라우드)을 통해 액세스됩니다.

**비용 절감:** DBaaS로 서버를 직접 설치하고 및 유지 관리하는 비용이 줄어듭니다.

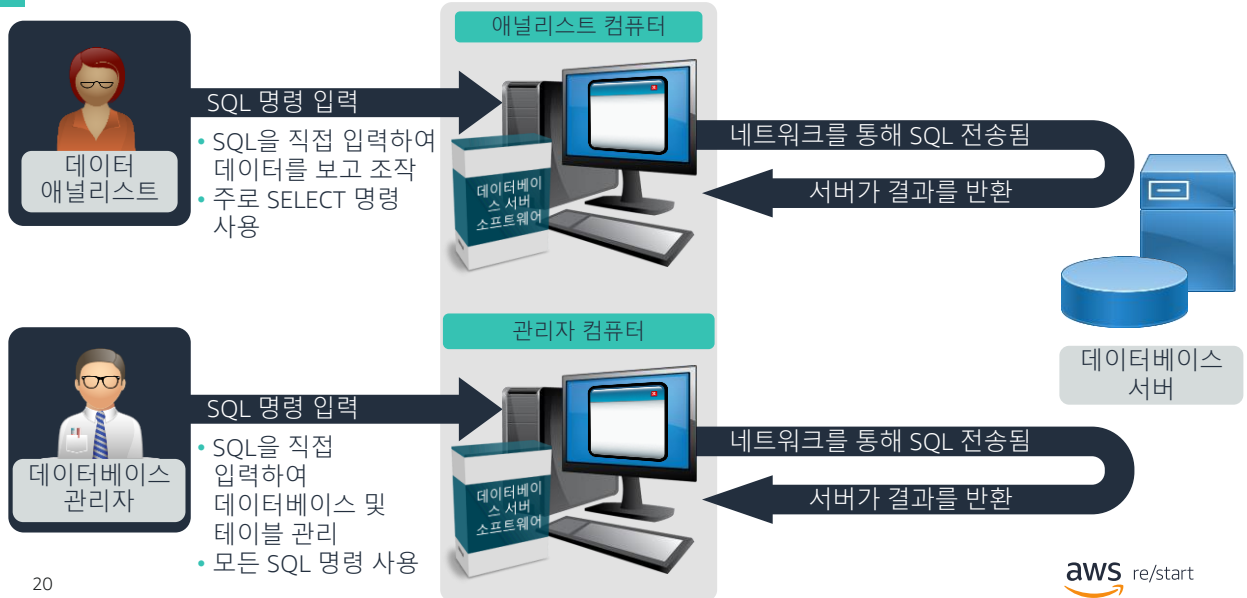
### 빠른 속도:

- DBaaS는 데이터베이스를 사용자에게 더 가깝게 배치하여 더 빠른 데이터베이스 액세스를 제공하며, 전 세계 모든 곳에 있는 클라이언트에 대한 액세스를 용이하게 만듭니다.
- AWS와 같은 서비스를 사용하여 데이터 센터에 대용량 스토리지 및 처리 능력을 제공합니다.
- 논리 서버를 사용해서는 처리할 수 없는 대량 데이터 세트를 처리할 수 있도록 합니다.
- DBaaS는 클라우드 기반 가상 머신(VM) 또는 클라우드 공급업체의 서비스로 설치할 수 있습니다.

## 데이터 상호 작용(DI)의 예

다음 몇 장의 슬라이드에서는 데이터베이스의 데이터와 상호 작용하는 다양한 방식을 검토합니다.

## 직접 상호 작용



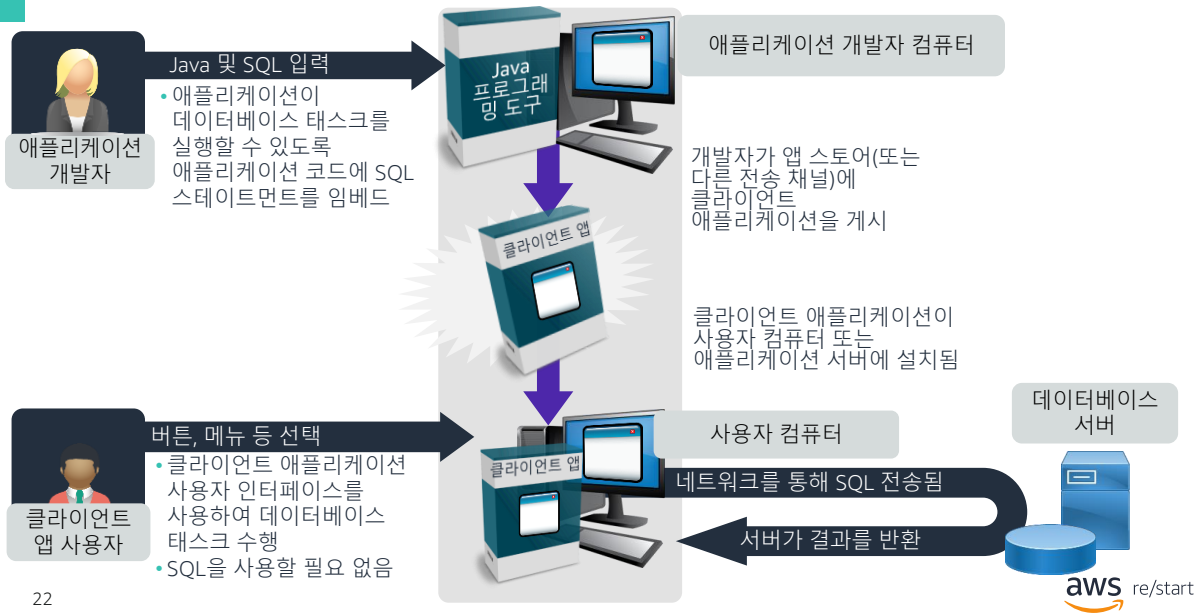
## 클라이언트-서버



클라이언트-서버 모델에서 애플리케이션은 DBMS의 사용자와 인터페이스 간의 상호 작용을 위한 도구를 제공하기 위해 생성됩니다.

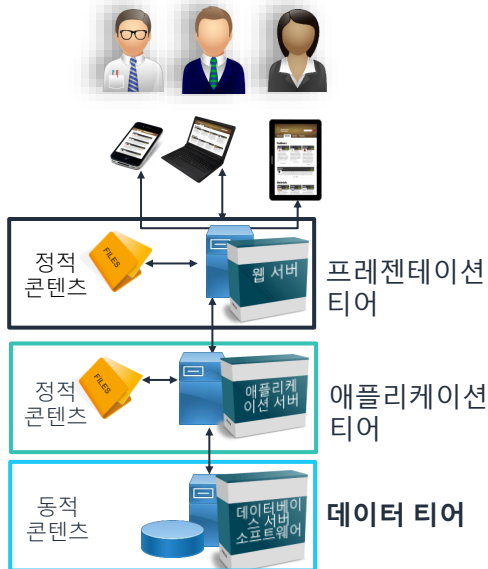


## 애플리케이션 코드에 임베드됨



직접 상호 작용을 위해서는 사용자가 특정 DBMS에서 SQL 명령을 실행하는 방법을 알아야 합니다.

## 웹 애플리케이션이 있는 3티어 애플리케이션



웹 사용자

웹 브라우저를 실행하는 클라이언트 컴퓨터와 디바이스가 웹 서버에 입력을 보냄

웹페이지를 실행하는 서버가 서버에서 정보를 수집하여 웹페이지 또는 클라이언트 앱에 데이터 표시

서버가 애플리케이션 서버와 데이터베이스 서버의 요청을 수신

데이터베이스 애플리케이션을 실행하는 서버

3티어 애플리케이션에서 사용자는 데이터베이스를 사용할 때 간접적으로 상호 작용에 참여합니다. 사용자와 데이터베이스 간의 커뮤니케이션에는 프론트 엔드 인터페이스 또는 프레젠테이션 티어(또는 계층)가 포함됩니다.

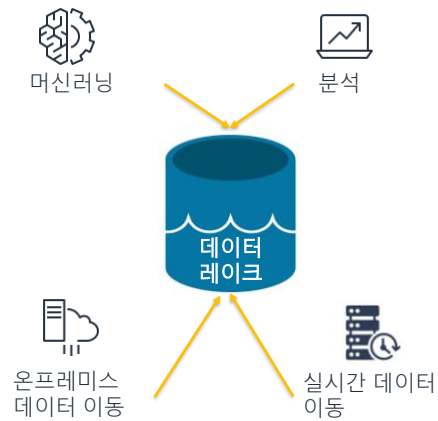


새로운 데이터 스토리지 기술

# 데이터 레이크

## 클라우드 메모리 बैं크

- 대시보드, 시각화, 빅 데이터 처리, 실시간 분석, 머신러닝을 비롯한 다양한 유형의 분석을 실행하여 더 나은 의사 결정을 지원할 수 있습니다.



데이터 레이크는 조직이 생성하고 저장하여 운영하는 모든 종류의 데이터를 저장, 관리 배포하는 중앙 집중식 리포지토리입니다. 데이터 유형으로는 사용자 데이터베이스 애플리케이션, 파일 또는 문서의 데이터가 포함됩니다. 먼저 데이터를 구조화할 필요 없이 데이터를 있는 그대로 저장할 수 있습니다.

## 학습 내용 확인 질문



\_\_\_\_\_ 데이터베이스는 종종 SQL 데이터베이스라고도 합니다.



데이터베이스 관리 시스템(DBMS)의 목적은 무엇입니까?

26

aws re/start

정답:

1. 관계형 데이터베이스
2. 데이터베이스 관리 시스템(DBMS)은 데이터베이스 기능을 제공하는 소프트웨어입니다.

## 핵심 사항



© 2020, Amazon Web Services, Inc. 또는 계열사. All rights reserved.

27

- 흔히 **SQL 데이터베이스**라고도 하는 **관계형 데이터베이스**를 사용하면 데이터를 여러 테이블에 분산할 수 있습니다.
- 흔히 **NoSQL 데이터베이스**라고도 하는 **비관계형 데이터베이스**는 데이터를 저장하기 위해 테이블 구조를 사용하지 않습니다.
- **데이터베이스 관리 시스템(DBMS)**은 데이터베이스 기능을 제공하는 소프트웨어입니다.
- **데이터 상호 작용**의 4가지 예로는 직접 상호 작용, 클라이언트-서버, 애플리케이션 코드에 임베드, 웹 애플리케이션이 있는 3티어 애플리케이션이 있습니다.
- **데이터 레이크**는 클라우드 메모리 बैं크와 같습니다.

aws re/start

이 강의에서 다룬 핵심 사항은 다음과 같습니다.

- 흔히 **SQL 데이터베이스**라고도 하는 **관계형 데이터베이스**를 사용하면 데이터를 여러 테이블에 분산할 수 있습니다.
- 흔히 **NoSQL 데이터베이스**라고도 하는 **비관계형 데이터베이스**는 데이터를 저장하기 위해 테이블 구조를 사용하지 않습니다.
- **데이터베이스 관리 시스템(DBMS)**은 데이터베이스 기능을 제공하는 소프트웨어입니다.
- **데이터 상호 작용**의 4가지 예로는 직접 상호 작용, 클라이언트-서버, 애플리케이션 코드에 임베드, 웹 애플리케이션이 있는 3티어 애플리케이션이 있습니다.
- **데이터 레이크**는 클라우드 메모리 बैं크와 같습니다.