

데이터베이스 서비스

알아둬야 할 데이터베이스 기초 지식

1. 데이터의 집합체, 데이터베이스

- 데이터베이스
 - 검색이나 등록을 쉽게 할 수 있게 정리된 데이터 모음
 - 응용 프로그램으로부터 데이터에 대해 업데이트, 참조, 삭제와 같은 작업을 요청받아 데이터 처리
- DBMS(DataBase Management System): 데이터를 관리하는 시스템
- RDBMS: 관계형 데이터 베이스 시스템
 - 하나의 표를 '테이블'이라는 단위로 관리
 - 여러 테이블이 관계성을 가지고 관리됨
- NoSQL 데이터베이스: 키-값 형태로 데이터를 관리

2. SQL로 조작하는 관계형 데이터 베이스

- SELECT: 데이터 검색
- INSERT: 데이터 추가
- UPDATE: 데이터 갱신
- DELETE: 데이터 삭제

3. 데이터 불일치를 없애기 위한 ACID 특성

- **ACID**
 - 관계형 데이터 베이스의 특징 중 하나
 - **원자성(Atomicity)**: 트랜잭션은 모두 실행되거나 실행되지 않아야 한다는 특성
 - **일관성(Consistency)**: 정해진 데이터베이스의 규칙을 만족해야 한다는 특성

- **독립성(Isolation):** 트랜잭션을 단독으로 실행할 때나 동시에 여러 개를 실행할 때 결과가 같아야 한다는 특성
- **지속성(Durability):** 장애가 있어도 완료된 트랜잭션의 결과는 손실되지 않아야 한다는 특성, 즉 해당 트랜잭션에 대한 로그가 남아있어야 함
- **트랜잭션**
 - 데이터베이스에서 수행하는 일련의 처리
 - 트랜잭션 단위가 ACID 특성을 가짐

AWS에서 관계형 데이터베이스 사용

1. 관계형 데이터베이스 서비스 RDS

- Amazon Relational Database Service: 관계형 데이터베이스 제공 서비스
- OS나 데이터베이스 엔진의 관리는 AWS 측에서 수행
- 종량 과금제 → RDS 인스턴스가 기동되는 시간만큼의 요금 발생
- **데이터베이스 엔진: 처리를 관리하는 소프트웨어**
 - **MySQL, SQL Server, MariaDB, PostgreSQL, Oracle, Amazon, Aurora**
 - Amazon Aurora: AWS가 클라우드용으로 구축한 데이터베이스

2. 데이터베이스 성능을 결정하는 인스턴스와 스토리지 유형

- RDS 인스턴스 유형
 - 인스턴스의 성능(CPU, 메모리) → 인스턴스 크기를 선택해 결정
 - 인스턴스 크기의 명명 규칙 → ex. db.m6g.large
 - db: 고정
 - m: 패밀리, 메모리가 우선인지 범용인지에 따라 지정하는 문자열
 - 6: 세대, 숫자가 클수록 새롭고 고성능
 - g: 추가 기능, 인스턴스가 실행될 서버의 특징 기재
 - large: 크기, 크기가 클수록 고성능, 크기가 클수록 요금도 비쌘
 - 종량제 요금이며 시간당 단가는 인스턴스 크기에 따라 다름

- 스토리지 유형

볼륨 유형	개요
범용 SSD	고성능으로 용량에 따라 읽기/쓰기 성능 결정
프로비저닝 IOPS SSD	고성능이며 지정한 읽기/쓰기 성능 확보
마그네틱 HDD	저렴한 비용

3. 가용성 향상을 위한 예비 복제본과 읽기 성능 향상을 위한 읽기 전용 복제본

- RDS는 데이터를 처리하는 프라이머리 인스턴스 외에 **예비 복제본(Standby replica)**을 다중 AZ에 배포해 가용성을 높일 수 있음
- 고가용성: 시스템이 중단되지 않고 작동할 수 있는 능력
 - RDS에서는 이를 예비 복제본으로 구현 → **다중 AZ 배포**
- **장애 조치(Fail Over):** 프라이머리 인스턴스나 AZ에 장애 발생 시 예비 복제본으로 전환
- 엔드포인트
 - 응용프로그램에서 RDS에 접속할 때 사용하는 DNS 형식의 접속 정보
 - 프라이머리 인스턴스에 장애 발생 시 엔드포인트의 목적지가 자동으로 예비 복제본으로 전환
- 읽기 전용 레플리카
 - 프라이머리 인스턴스에서 복제된 인스턴스
 - 읽기 요청이 늘어나면 인스턴스의 수를 늘리는 스케일 아웃 방식으로 성능 향상 가능
 - 읽기 전용이므로 쓰기 성능 향상 불가
- 쓰기 성능은 인스턴스 크기를 변경하는 스케일 업으로 성능을 향상해야 함
- 가용성 향상이 목적인 예비 복제본은 일반적으로 처리는 수행하지 않는 인스턴스이므로 성능 향상에 도움X

4. RDS 보안 설정과 백업

- RDS 접근 제어
 - RDS 인스턴스 생성을 위해서는 최소 2개의 서브넷 필요

- 서버넷은 리전 내의 서로 다른 AZ를 지정해야 함
- 데이터베이스 접근 제어는 보안 그룹과 네트워크 ACL에서 설정
- 데이터베이스 엔진 업데이트
 - 유지 관리 기간이라는 사용자가 설정한 시간대에 실시
 - **다중 AZ 배포 사용 시 예비 복제본 → 프라이머리 인스턴스 순서로 유지보수 실행**
 - 데이터베이스 엔진 업그레이드: 프라이머리 인스턴스와 예비 복제본 동시 업그레이드
- 데이터 백업
 - RDS: 자동 백업 기능이 있어 사용자가 지정한 일 수만큼의 스냅샷 자동 유지
 - **특정 시점 복구(PITR, Point In Time Recovery)**
 - 특정 시간을 지정하고 복원 가능
 - 백업으로 보존하는 용도보다 특정 시간으로 되돌리는 용도로 사용
 - 스냅샷과 데이터베이스 변경 로그를 조합해 기능 구현

고기능 Aurora의 편리한 특징

1. AWS에 최적화된 데이터베이스

- Amazon Aurora: AWS가 클라우드용으로 구축한 데이터베이스
- **DB 클러스터라는 단위로 관리**
- 처리를 수행하는 하나 이상의 DB 인스턴스와 데이터를 관리하는 클러스터 볼륨으로 구성
 - 클러스터 내의 인스턴스에서 스토리지가 공유되는 것이 특징
 - 클러스터 볼륨은 3개의 AZ에 2개씩 총 6개의 사본을 작성해 높은 내결함성을 가짐
- 일반 데이터베이스 엔진에서는 EBS가 인스턴스에 연결돼 인스턴스별로 데이터 관리
- **쿼럼모델**이라는 복제 관리 구조 사용
 - 데이터의 복제를 만들어 원본과 별도로 저장하는 것

- 쓰기 처리는 6개 중 4개, 읽기 처리는 6개 중 3개가 성공하면 클라이언트에 성공이라고 반환

2. AWS에 최적화된 Aurora 기능

- Aurora 복제
 - 1개로 가용성 향상과 읽기 성능 향상 모두 달성 가능
 - 평상시에는 읽기 처리를 수행해 읽기 성능 향상을 올리고, 장애가 발생하면 그 즉시 프라이머리 인스턴스로 페일 오버
- 읽기용 엔드포인트의 URL 지정 → 각 복제에 분산 처리 가능
- 스토리지 유형과 용량을 지정할 필요X
- 자체 자동 백업 기능 존재
- 다중 리전에 걸친 글로벌 데이터베이스
- 두 개의 기본 인스턴스가 있는 멀티 마스터 클러스터
- 인스턴스 크기 지정이 필요하지 않은 Aurora Serverless
- S3와 연계한 파일 내보내기, 가져오기 기능

키와 값의 조합으로 데이터 관리

1. 키-값 데이터베이스란

- Amazon DynamoDB: key-value 형 데이터베이스
 - 복잡한 검색 불가
 - 고속으로 데이터 추출 가능
- 키의 이름과 그에 대한 값으로 구성된 데이터
- **데이터 형태로는 자유도가 매우 높지만 집계나 검색이 매우 어려움**
- 일련의 데이터를 '레코드'라고 하고, 레코드의 각 항목을 '속성'이라고 함
- 기본 키를 가지고 있으면 다른 속성은 어떤 것을 가지고 있어도 상관X
- **각 레코드는 기본적으로 기본 키로만 검색 가능**

2. DynamoDB란

- AWS가 만든 고속 키-값 데이터베이스
- 내부적으로 프라이머리 키 검색을 쉽게 할 수 있게 데이터를 정리해 보존
- 서버리스
 - 사용자가 서버 관리를 하지 않아도 됨

3. DynamoDB 이용 요금

- 사용자는 데이터의 집합인 '테이블'만 소유하는 형태 (서버리스 서비스이므로)
- 데이터 보존 스토리지는 테이블의 데이터양에 따라 필요한 크기를 할당받으며, 관련 조작성은 DynamoDB가 직접 수행
- 요금
 - 요청을 처리할 양을 예약하고 그 양만큼 요금 부과, 데이터 보관료 발생
 - 처리량 무제한 설정 가능, 이 경우 이용한 만큼 요금 발생
- 요청 처리량 예약: 1초에 몇 번 읽기/쓰기 요청을 받아들일 것인지
- 예약량을 초과하는 요청에 대해 오류 반환 but Auto Scaling 사용 시 실제 요청 수에 따라 지정된 범위 내 예약량 자동 변경 가능
- **처리량 = 커패시티 유닛 = RCU(Read Capacity Unit) = WCU(Write Capacity Unit)**
- 처리량 예약 방식 = 프로비저닝 용량 모드
예약하지 않는 방식 = 온디맨드 용량 모드 → 요청량이 많을 시 예상 이상의 요금 발생 주의

4. DynamoDB 보수

- AWS에서는 장애 대비를 위해 DynamoDB 데이터를 이중화하여 저장
- 3개의 AZ에 실시간으로 데이터를 복제해 어느 하나의 AZ에 장애가 발생하더라도 문제없이 서비스 이용 가능
- 특정 시점 복구(PITR)이라는 실시간 백업 기능 존재
 - 지난 35일 내 임의의 시점으로 테이블 데이터 복구 가능 → 갑작스러운 데이터 손실에도 대응 가능

- 임의의 시점을 지정해 백업 취득 가능 → 수동으로 취득한 백업은 무기한 보관 가능

5. DynamoDB 이용 예시

- 데이터 교환이 매우 많고 사용자에게 빠른 응답이 필요한 응용 프로그램에 이용됨

데이터 분석을 위한 대량의 데이터 수집

1. 분석용 데이터베이스 Redshift

- 데이터 분석을 위한 데이터 웨어하우스 서비스
- 다른 리전에 있는 대량의 데이터를 가져와 SQL이나 BI(Business Intelligence) 도구로 분석
- 열 지향 스토리지 구조 → 열 별로 데이터를 분석
 - 열별로 데이터 보존 → 분석이 쉬워지고 성능 효율이 높아짐
- SQL 사용
 - SQL 연결을 받아들이는 리더 노드
 - 스토리지 및 SQL 문을 실행하는 컴퓨터 노드 → 처리를 분산해 고속으로 실행
- 최신 RedShift는 RA3이라는 노드 유형 사용
 - 캐시 데이터를 컴퓨팅 노드로 가져오고 실제 데이터는 S3에 보관
 - S3는 비용이 낮아 캐시 데이터로 빠른 분석을 가능하게 하면서도 낮은 비용으로 이용 가능
- 분석 및 데이터 읽기에 특화된 데이터베이스
 - INSERT, UPDATE 비권장
 - COPY 명령으로 여러 개의 텍스트 데이터를 병렬 업로드 하는 것을 추천

데이터베이스 마이그레이션에 도움이 되는 2가지 서비스

1. 데이터베이스 마이그레이션의 중요성

- 시스템 마이그레이션 시 데이터베이스 마이그레이션이 가장 어렵고, 다운타임(서비스 중단 시간)에도 큰 영향을 미침

2. AWS Database Migration Service

- 데이터베이스 마이그레이션을 위한 서비스
- 현재의 데이터베이스에서 새로운 데이터베이스로 쉽게 데이터를 옮기는 역할
- 데이터 형식을 맞춰 옮겨주거나, 도중 발생한 원본 데이터의 변경 내용 추적 반영 가능
- 마이그레이션 완료 후 데이터 검증 기능 제공

3. AWS Schema Conversion Tool

- 스키마: 데이터베이스의 데이터 구조 (데이터베이스 설계도)
- 마이그레이션 전 스키마 변환 작업 필요
- 스키마 변환을 위한 도구
- 스키마 변환에 대한 평가 보고서 확인 가능

다양한 데이터베이스를 지원하는 7가지 서비스

1. 다양한 데이터베이스 서비스

- **RDB(관계형 데이터베이스) 외의 데이터베이스 → NoSQL(Not Only SQL)**

2. Amazon ElastiCache

- 인 메모리 데이터 스토어 서비스
- 서버의 주기억장치(메모리)에 데이터를 저장해 고속으로 데이터를 읽고 쓸 수 있게 함
- 캐싱과 동일하게 사용되므로 '인 메모리 캐싱'이라고도 함
- 속도는 매우 빠르지만 시스템 정지 시 메모리에 저장된 데이터도 모두 삭제됨
- 완전 관리형 서비스로 응용 프로그램/데이터베이스 성능 가속화 가능
- 내구성이 필요하지 않은 서비스에서 기본 데이터 스토어로 사용 가능
- 카-값 데이터베이스 지원

3. Amazon MemoryDB

- 데이터의 지속성을 확보하면서 빠른 읽기 및 쓰기 가능
- 메모리뿐만 아니라 보조 저장 장치에도 데이터를 저장해 메모리의 데이터가 사라지더라도 복구할 수 있는 기능을 갖추
- 읽기 요청에 대한 처리는 빠르지만 쓰기 요청에 대한 처리는 약간 느림

4. Amazon DocumentDB

- 문서 지향 데이터베이스 제공 서비스
- MongoDB와 호환
- 데이터를 JSON 형식으로 보관하고 해당 내용 검색
- 기본 키로만 레코드 검색이 가능한 키-값 데이터베이스와는 달리 JSON 내의 모든 요소를 이용해 레코드 검색 가능

5. Amazon Neptune

- 그래프 데이터베이스
 - 데이터 간의 관계성을 나타내는 데 특화된 구조
 - 노드, 엣지, 속성으로 구성

6. Amazon Quantum Ledger Database

- 원장 데이터를 관리하는 데이터베이스
- 저널이라는 로그 데이터에 기록되는 데이터의 변경 이력을 차례로 기록하는 방식
- 저널은 추가만 가능하며 변경이나 삭제를 할 수 없는 구조 → 신뢰성 높음

7. Amazon Managed Blockchain

- 블록체인 네트워크를 구축하는 서비스
- 블록체인: 분산형 원장 기술의 한 종류

- 데이터의 변경 내역을 기록하는 원장을 네트워크에 참여한 여러 서버에서 협력해 관리하는 시스템
- QLDB와 비슷하지만 데이터를 여러 곳에서 분산시켜 보관하는 기술이므로 엄밀히 데이터베이스는 아님

8. Amazon Timestream

- 시계열 데이터베이스를 제공하는 서비스
- 시간마다 변화하는 값을 순차적으로 기록하기에 적합한 데이터 베이스
- 데이터의 평활화, 근사, 보관 기능 등을 이용한 분석도 가능

9. 기타 서비스

- Amazon Simple Email Service
- Amazon EventBridge
- Amazon Simple Notification Service
- AWS Marketplace
- 엣지 로케이션