

[2-1. 데이터베이스 기본 개념 소개]

1. 데이터베이스

1-1. 데이터베이스 개념

- 조직적으로 저장되어 쉽게 접근, 관리, 업데이트할 수 있도록 구성된 데이터의 집합 또는 데이터의 저장공간 자체를 의미
- 데이터를 구조화하여 저장함으로써 정보 검색 및 관리의 효율성을 높이고, 데이터간의 관계를 정의하여 복잡한 데이터 분석과 보고가 가능하게 함.

1-2. 데이터베이스 주요 특징

- 구조화된 데이터 : 데이터를 테이블, 레코드, 필드 등으로 구조화하여 저장합니다. 이 구조는 데이터를 쉽게 저장하고, 검색하며, 관리할 수 있게 해줍니다.
- 데이터 무결성과 일관성 : 데이터의 정확성과 일관성을 유지하기 위한 규칙과 제약 조건을 설정할 수 있습니다. 이는 데이터의 품질을 보장하는데 중요한 역할을 합니다.
- 데이터 보안 : 사용자 권한 및 액세스 제어를 통해 데이터의 보안을 유지합니다.
- 백업 및 복구 : 데이터 손실을 방지하기 위해 정기적인 백업과 복구 매커니즘을 제공합니다. 이는 시스템의 장애나 데이터 손실 사건 발생 시 중요한 역할을 합니다.
- 동시성 관리 : 여러 사용자가 동시에 데이터베이스에 접근하고 작업을 수행할 때, 데이터의 일관성을 유지하며 충돌을 방지하는 기능을 제공합니다.

2. 데이터베이스 관리 시스템(DBMS, Database Management System)

2-1. DBMS 개념

- 데이터베이스 내의 데이터를 생성, 검색, 업데이트 및 관리할 수 있게 해주는 소프트웨어.
- DBMS는 사용자와 데이터베이스 사이의 인터페이스 역할을 하며, 데이터의 무결성, 보안, 백업 및 복구를 관리합니다.

2-2. DBMS 주요기능

- 데이터 정의 : DBMS는 데이터 모델을 사용하여 데이터베이스의 구조를 정의합니다. 이를 통해

데이터베이스에 저장될 데이터의 타입, 관계, 제약 조건 등을 지정할 수 있습니다.

- 데이터 조작 : 사용자는 SQL과 같은 쿼리 언어를 사용하여 데이터를 검색, 추가, 수정, 삭제할 수 있습니다. DBMS는 이러한 요청을 처리하고 결과를 반환합니다.
- 데이터의 무결성 및 보안 유지 : DBMS는 데이터의 정확성과 일관성을 유지하기 위한 무결성 제약 조건을 강제 적용합니다. 또한, 사용자 권한 및 역할을 관리하여 데이터 접근을 제어함으로써 데이터의 보안을 유지합니다.
- 동시성 제어 : 동시에 여러 사용자가 데이터베이스에 접근할 때, DBMS는 동시성 제어 메커니즘을 통해 데이터의 일관성을 보장하고 충돌을 방지합니다.
- 백업 및 복구 : DBMS는 데이터 손실을 방지하기 위한 백업 기능을 제공하며, 시스템 장애나 데이터베이스 손상 시 데이터를 복구할 수 있는 기능을 지원합니다.

2-3. DBMS의 유형

- 관계형 데이터베이스 관리 시스템(RDBMS) : 데이터를 테이블로 구성하고, 테이블 간의 관계를 기반으로 데이터를 조직화합니다.
- NoSQL 데이터베이스 관리 시스템 :
 - 비관계형 데이터 모델을 사용하여, 대규모 분산 데이터를 유연하게 저장하고 관리합니다.
 - NoSQL은 Not Only SQL의 약자로, 전통적인 관계형 데이터베이스 시스템의 구조화된 쿼리 언어(SQL)만을 사용하지 않는 데이터베이스 관리 시스템(비관계형 데이터베이스 시스템)을 지칭합니다.

[RDBMS와 NoSQL DB 관리 시스템 비교]

구분	관계형 데이터베이스 관리 시스템 (RDBMS)	NoSQL 데이터베이스 관리 시스템
데이터 모델	테이블 기반. 데이터는 열과 행으로 구성된 테이블에 저장됨.	키-값 쌍, 문서, 그래프, 열 기반 등 다양한 데이터 모델 사용.
스키마	엄격한 스키마. 테이블 구조는 사전에 정의되어야 함.	유연한 스키마. 데이터 구조는 동적으로 조정될 수 있음.
확장성	수직 확장성. 데이터베이스 서버의 성능을 높여 확장.	수평 확장성. 더 많은 서버를 추가하여 데이터베이스를 확장.
트랜잭션 지원	강력한 트랜잭션 지원(ACID 속성 준수).	트랜잭션 지원은 선택적이며, 일부는 ACID를 완전히 지원하지 않음.
복잡한 쿼리	복잡한 JOIN 연산 등 복잡한 쿼리에 강력한 성능을 제공.	쿼리 기능이 제한적일 수 있으나, 특정 모델에서는 고유한 쿼리 최적화 제공.
일반적인 사용 사례	금융 시스템, ERP, CRM과 같이 복잡한 트랜잭션이 필요한 애플리케이션.	빅 데이터 처리, 실시간 웹 애플리케이션, IoT, 콘텐츠 관리 등.
대표적인 시스템 예시	MySQL, PostgreSQL, Oracle, Microsoft SQL Server	MongoDB, Cassandra, Redis, Couchbase

3. MySQL

3-1. MySQL 소개

- MySQL은 세계에서 가장 널리 사용되는 오픈 소스 관계형 데이터베이스 관리 시스템(RDBMS) 중 하나입니다.
- 1995년에 스웨덴의 프로그래머인 마이클 와이드니어스(Michael Widenius)와 데이비드 악스마크(David Axmark)가 공동으로 개발했습니다. 처음에는 개인 프로젝트로 시작되었으나, 이후 오픈 소스 관계형 데이터베이스 관리 시스템(RDBMS)으로서 빠르게 성장했습니다.
- 현재 MySQL은 오라클(Oracle Corporation)의 제품 포트폴리오에 포함되어 있으며, 오라클 주도 하에 개발 및 유지보수되고 있고, 커뮤니티 에디션은 여전히 오픈 소스로 제공되고 있습니다.
- 오라클은 MySQL을 계속해서 발전시키고 있으며, 기업용 솔루션으로서의 MySQL을 강화하기 위해 여러 상업적 라이선스와 지원 옵션을 제공하고 있습니다. MySQL은 그 사용의 용이성, 성능, 유연성 덕분에 전 세계적으로 널리 사용되며, 다양한 분야에서 인기 있는 데이터베이스 관리 시스템으로 자리잡고 있습니다.
- MySQL은 웹 사이트 및 웹 애플리케이션, 블로그, 콘텐츠 관리 시스템(CMS), 온라인 쇼핑몰, 포럼, 데이터 웨어하우스 등 다양한 영역에서 데이터 저장과 관리를 위해 사용됩니다. Facebook, Twitter, YouTube, Wikipedia 등 많은 대형 웹사이트와 서비스들이 MySQL을 데이터베이스 솔루션으로 사용하고 있습니다.

3-2. MySQL 주요 특징

- 오픈 소스 : MySQL은 GPL(General Public License)에 따라 배포되어 무료로 사용할 수 있습니다.
- 포터빌리티 : MySQL은 다양한 운영체제에서 실행될 수 있습니다. Linux, Windows, macOS 등 주요 플랫폼을 지원합니다.
- 높은 성능 : MySQL은 속도와 효율성을 강조하는 설계로 알려져 있습니다. 복잡한 쿼리, 대용량 데이터베이스, 높은 동시성 요구사항 등을 효과적으로 처리할 수 있도록 최적화되어 있습니다.

4. SQL

4-1. SQL 개요

- SQL(Structured Query Language)은 관계형 데이터베이스 관리 시스템(RDBMS)에서 데이터를 관리하기 위해 설계된 프로그래밍 언어입니다.
- 데이터베이스에서 데이터를 조회, 생성, 수정, 삭제하는데 사용되며, 데이터베이스 스키마 생성과 수정, 데이터베이스 객체 관리 등의 다양한 기능도 수행합니다.
- SQL은 데이터베이스와 상호작용하는 표준적인 방법을 제공하며, 다양한 RDBMS에서 광범위하게 사용됩니다.

4-2. SQL 주요 특징

- 비절차적 언어 :
 - SQL은 사용자가 '무엇(What)'을 하고 싶은지를 선언하는 방식으로, '어떻게(How)' 해당 작업을 수행할지에 대한 과정을 명시하지 않는 비절차적 언어입니다.
 - 절차적 언어의 경우 프로그램의 명령어들이 실행되어야 할 정확한 순서와 방법을 개발자가 명시적으로 기술해야 합니다.
 - 비절차적 언어의 경우 "데이터베이스에서 나이가 20세 이상인 모든 사람의 이름을 나열하라"와 같이 원하는 결과를 선언하고, DBMS의 SQL엔진이 해당 작업을 어떻게 처리할지 결정합니다.
- 표준화 : SQL은 ANSI(American National Standards Institute)와 ISO(International Organization for Standardization)에 의해 표준화되었습니다. 이로 인해 대부분의 RDBMS에서 공통된 형태의 SQL을 사용할 수 있습니다.
- 다양한 데이터베이스 작업 지원 : SQL은 데이터 조회(SELECT), 데이터 삽입(INSERT), 데이터 수정(UPDATE), 데이터 삭제(DELETE)와 같은 기본적인 데이터 조작 작업뿐만 아니라, 테이블 생성(CREATE TABLE), 테이블 구조 변경(ALTER TABLE), 테이블 삭제(DROP TABLE) 등 데이터베이스 스키마를 관리하는 작업도 지원합니다.

5. 주요 용어

5-1. Schema(스키마)

- 스키마는 데이터베이스 내에서 데이터의 조직, 구조, 관계를 정의하는 메타데이터의 집합입니다. 스키마는 데이터베이스의 설계도와 같으며, 테이블, 뷰, 인덱스, 관계 등 데이터베이스 객체의 구

조를 정의합니다.

- 하나의 데이터베이스는 하나 이상의 스키마를 가질 수 있으며, 각 스키마는 여러 테이블을 포함할 수 있습니다.
- 데이터베이스와 스키마는 밀접하게 연관되어 있기 때문에, 일상적인 대화나 문서에서는 두 용어가 혼용되어 사용되는 경우가 많습니다. 특히, 하나의 데이터베이스에 하나의 스키마만 사용하는 경우가 많아, 두 용어가 동일한 의미로 받아들여지는 경우가 있습니다.

5-2. Table(테이블)

- 테이블은 데이터베이스에서 데이터를 저장하는 기본 단위입니다.
- 테이블은 행(Row)과 열(Column)로 구성되며, 각 행은 테이블에 저장된 개별 레코드를 나타내고, 각 열은 데이터의 속성(필드)을 나타냅니다.

5-3. Column(열, 필드)

- 열은 테이블에서 하나의 속성을 나타냅니다.
- 예를 들어, '고객' 테이블에는 '고객 이름', '주소', '전화번호' 등의 열이 있을 수 있으며, 각 열은 해당 속성의 데이터 타입(예: 문자열, 숫자, 날짜 등)을 갖습니다.

5-4. Primary Key(기본키)

- 기본키는 테이블의 각 행(레코드)을 고유하게 식별하는 데 사용되는 하나 이상의 열(필드)의 집합입니다.
- 기본키의 값은 해당 테이블 내에서 유일해야 하며, NULL 값이 될 수 없습니다. 이러한 특성으로 인해 기본키는 데이터의 무결성을 유지하는데 핵심적인 역할을 합니다.

[기본키 주요 특징]

구분	내용
고유성 (Uniqueness)	기본키로 지정된 열의 값은 테이블 내에서 고유해야 합니다. 이는 같은 테이블 내에 중복된 값을 가진 레코드가 존재할 수 없음을 의미합니다.
불변성 (Immutability)	일단 생성된 후에는 기본키의 값이 변경되어서는 안됩니다. 레코드의 식별자로서의 기능을 유지하기 위해, 기본키의 값은 시간이 지나도 변하지 않아야 합니다.
NOT NULL	기본키로 지정된 열은 NULL 값을 가질 수 없습니다. 모든 레코드는 반드시 유효한 식별자를 가져야 하기 때문입니다.

5-5. Foreign Key(외래키)

- 외래키는 다른 테이블의 기본키를 참조하는 열입니다. 외래키를 사용하여 테이블 간의 관계를 정의하고, 데이터의 무결성을 유지합니다.
- 예를 들어, '주문' 테이블의 '고객ID' 열이 '고객' 테이블의 기본키인 '고객ID'를 외래키로 참조할

수 있습니다.