

2-02. 데이터베이스 관리 시스템의 정의

데이터베이스 관리 시스템

DBMS(DataBase Management System) ex) oracle, MySQL

파일 시스템의 문제를 해결하기 위해 제시된 소프트웨어

조직에 필요한 데이터를 데이터베이스에 통합하여 저장하고 관리함



그림 2-4 파일 시스템과 데이터베이스 관리 시스템

데이터베이스 관리 시스템의 기능

정의 기능	데이터베이스 구조를 정의하거나 수정할 수 있다.	DDL
조작 기능	데이터를 삽입·삭제·수정·검색하는 연산을 할 수 있다.	DML
제어 기능	데이터를 항상 정확하고 안전하게 유지할 수 있다.	DCL --> SQL

그림 2-6 데이터베이스 관리 시스템의 주요 기능

데이터베이스 관리 시스템에서의 데이터 관리

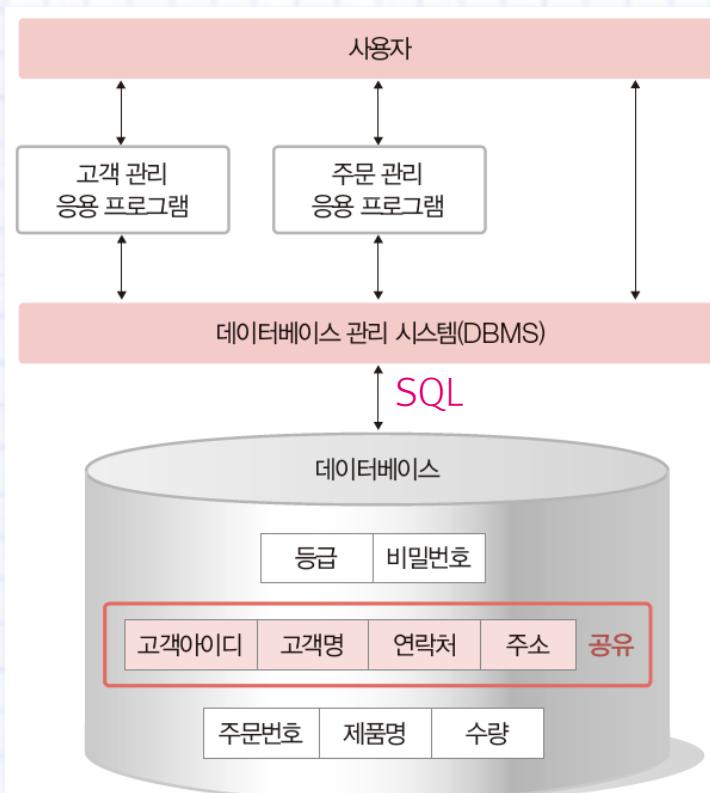


그림 2-5 데이터베이스 관리 시스템에서의 데이터 관리

2-03. 데이터베이스 관리 시스템에서의 장단점

장점	단점
<ul style="list-style-type: none">데이터 중복을 통제할 수 있다데이터 독립성이 확보된다데이터를 동시 공유할 수 있다데이터 보안이 향상된다데이터 무결성을 유지할 수 있다표준화할 수 있다장애 발생 시 회복이 가능하다응용 프로그램 개발 비용이 줄어든다	<ul style="list-style-type: none">비용이 많이 듈다백업과 회복 방법이 복잡하다중앙 집중 관리로 인한 취약점이 존재한다

그림 2-7 데이터베이스 관리 시스템의 장점과 단점

2-04. 데이터베이스 관리 시스템의 발전 과정

1세대 : 네트워크 DBMS, 계층 DBMS

네트워크 DBMS : 데이터베이스를 그래프 형태로 구성
ex) IDS(Integrated Data Store)

계층 DBMS : 데이터베이스를 트리 형태로 구성
ex) IMS(Information Management System)

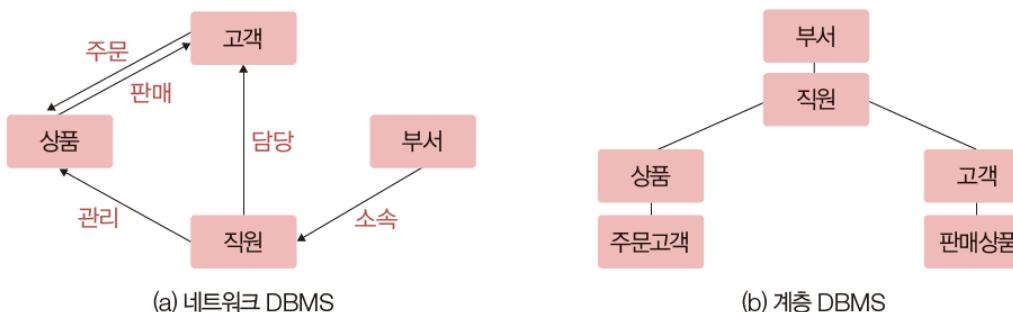


그림 2-8 1세대 DBMS 구조의 예

2세대 : 관계 DBMS (가장 많이 쓰이고 있음)

관계 DBMS : 데이터베이스를 테이블 형태로 구성
ex) 오라클(Oracle), MS SQL 서버, 액세스(Access),
인포믹스(Informix), MySQL

아이디	비밀번호	이름	연락처	주소	적립금
apple	1234	정소화	02-111-1111	서울시 마포구	1000
banana	9876	김선우	02-222-2222	경기도 부천시	500

그림 2-9 관계 DBMS의 테이블 예 : 고객 테이블

3세대 : 객체지향 DBMS, 객체관계 DBMS

객체지향 DBMS : 객체를 이용해 데이터베이스를 구성
ex) 오투(O2), 온투스(ONTOS), 젬스톤(GemStone)

객체관계 DBMS : 객체 DBMS + 관계 DBMS

4세대 : NoSQL • NewSQL DBMS

NoSQL DBMS : 비정형 데이터를 처리하는데 적합하고 확장성이 뛰어남
- 안정성과 일관성 유지를 위한 복잡한 기능을 포기
- 데이터 구조를 미리 정해두지 않는 유연성 (요새는 동영상 형식이 많으니 자주사용)
- 확장성이 뛰어나 여러 대의 서버 컴퓨터에 데이터를 분산하여 저장하고
처리하는 환경에서 주로 사용
- ex) 몽고디비(MongoDB), H베이스(HBase), 카산드라(Cassandra),
레디스(Redis), 네오포제이(Neo4j), 오리엔트DB(OrientDB) 등

NewSQL DBMS: 관계 DBMS의 장점+ NoSQL의 확장성 및 유연성
ex) 구글 스패너(Spanner), 볼트DB(VoltDB), 누오DB(NuoDB)

DBMS의 발전과정



그림 2-10 DBMS의 발전 과정

3-01. 데이터베이스 시스템의 정의

데이터베이스 시스템(DBMS)

데이터베이스에 데이터를 저장하고,
이를 관리하여 조직에 필요한 정보를 생성해주는 시스템

3-02. 데이터베이스의 구조

스키마와 인스턴스

스키마 (schema) : 데이터베이스에 저장되는 데이터 구조와 제약조건을 정의 “실제 DB”

인스턴스(instance) : 스키마에 따라 데이터베이스에 실제로 저장된 값

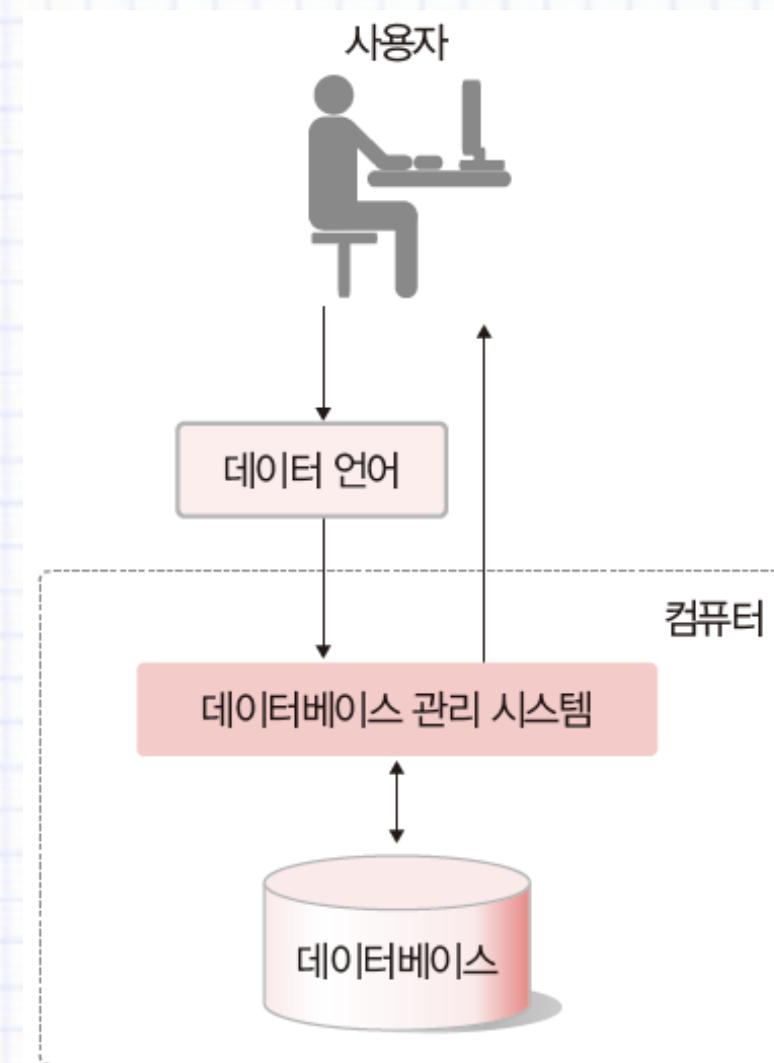


그림 3-1 데이터베이스 시스템의 구성



그림 3-2 스키마의 예

3단계 데이터베이스의 구조

미국 표준화 기관인 ANSI/SPARC에서 제안

데이터베이스를 쉽게 이해하고 이용할 수 있도록 하나의 데이터베이스를 관점에 따라 세 단계로 나눈 것

1. 외부 단계(external level) : 개별 사용자 관점

필요한 실제 데이터

2. 개념 단계(conceptual level) : 조직 전체의 관점

전체 시스템(DBMS)

3. 내부 단계(internal level) : 물리적인 저장 장치의 관점

실제 저장 장소

각 단계별로 다른 추상화(abstraction) 제공 : 내부 단계에서 외부 단계로 갈수록 추상화 레벨이 높아짐

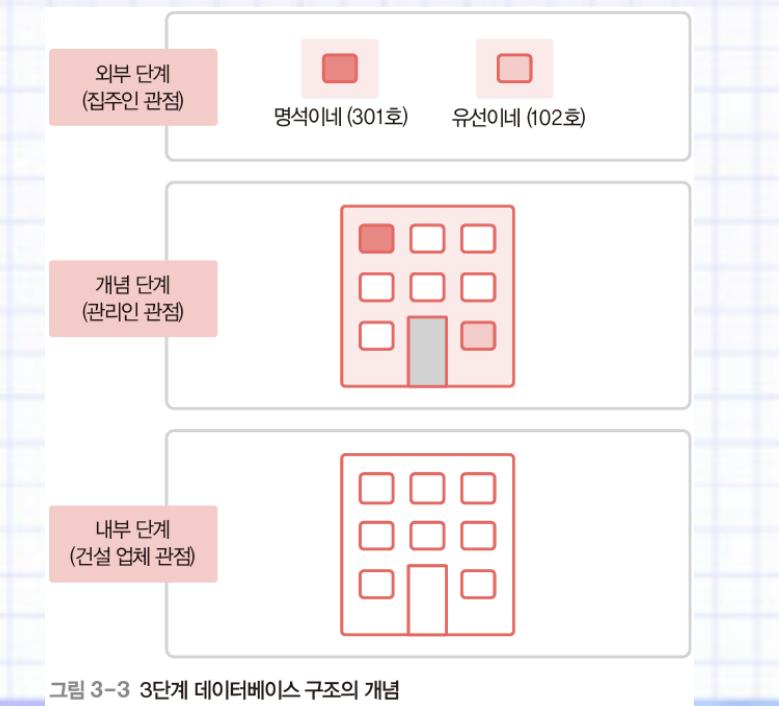


그림 3-3 3단계 데이터베이스 구조의 개념

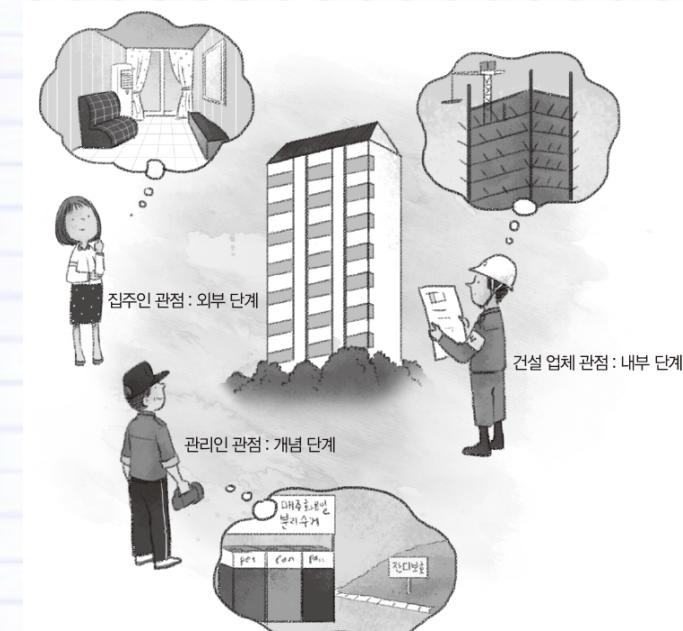


그림 3-4 3단계 데이터베이스 구조의 이해

3-02. 데이터베이스의 구조

3단계 데이터베이스의 구조 : 외부단계

데이터베이스를 개별 사용자 관점에서 이해하고 표현하는 단계

데이터베이스 하나에 외부 스키마가 여러 개 존재할 수 있음

외부 스키마(external schema)

- 외부 단계에서 사용자에게 필요한 데이터베이스를 정의한 것
- 각 사용자가 생각하는 데이터베이스의 모습, 즉 논리적 구조로 사용자마다 다름
- 서브 스키마(sub schema)라고도 함

3단계 데이터베이스의 구조 : 개념단계

데이터베이스를 조직 전체의 관점에서 이해하고 표현하는 단계

데이터베이스 하나에 개념 스키마가 하나만 존재함

개념 스키마(conceptual schema)

- 개념 단계에서 데이터베이스 전체의 논리적 구조를 정의한 것
- 조직 전체의 관점에서 생각하는 데이터베이스의 모습
- 전체 데이터베이스에 어떤 데이터가 저장되는지,
- 데이터들 간에는 어떤 관계가 존재하고 어떤 제약조건이 있는지에 대한 정의뿐만 아니라,

데이터에 대한 보안 정책이나 접근 권한에 대한 정의도 포함

3단계 데이터베이스의 구조 : 내부단계

데이터베이스를 저장 장치의 관점에서 이해하고 표현하는 단계

데이터베이스 하나에 내부 스키마가 하나만 존재함

내부 스키마(internal schema)

- 전체 데이터베이스가 저장 장치에 실제로 저장되는 방법을 정의한 것
- 레코드 구조, 필드 크기, 레코드 접근 경로 등 물리적 저장 구조를 정의

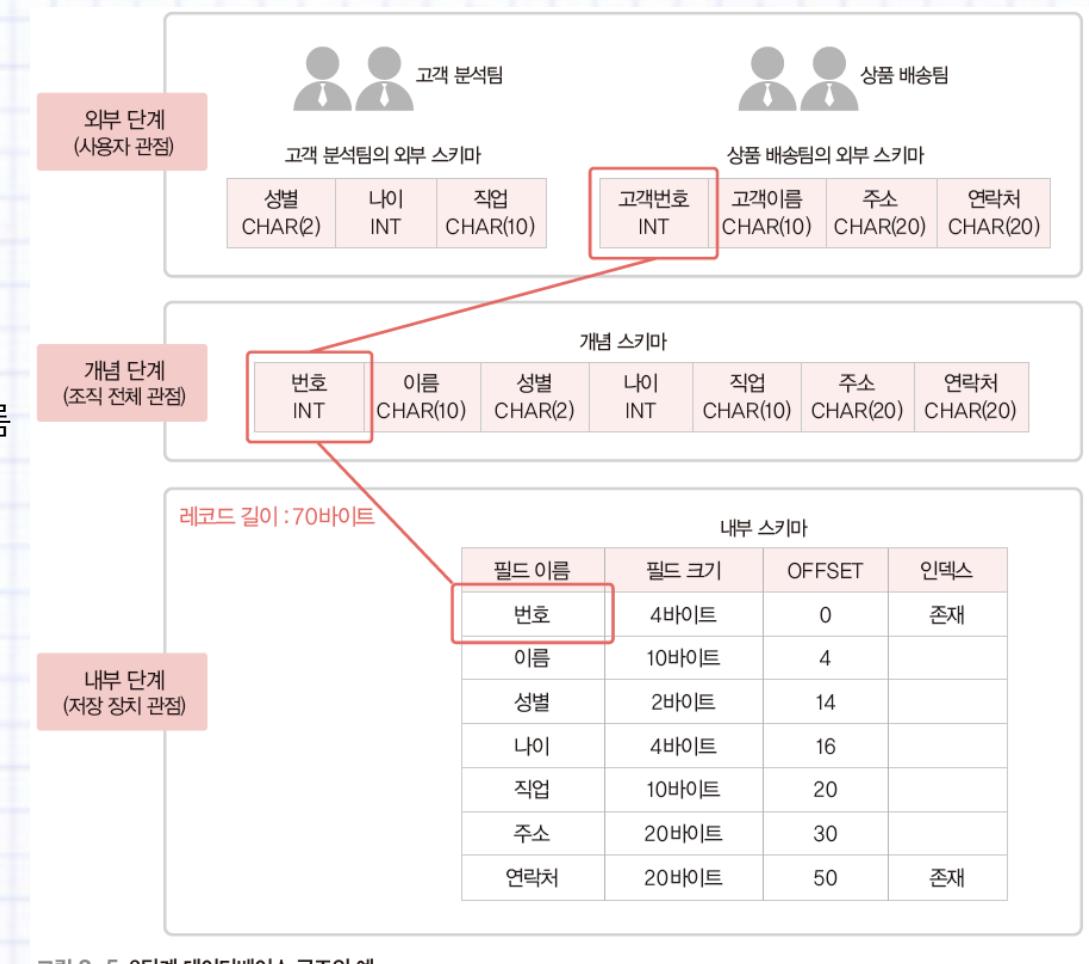


그림 3-5 3단계 데이터베이스 구조의 예

3단계 데이터베이스의 구조의 사상 또는 매팅

스키마 사이의 대응 관계

- 외부/개념 사상 : 외부 스키마와 개념 스키마의 대응 관계
응용 인터페이스(application interface)라고도 함
- 개념/내부 사상 : 개념 스키마와 내부 스키마의 대응 관계
저장 인터페이스(storage interface)라고도 함

데이터베이스를 3단계 구조로 나누고 단계별로 스키마를 유지하며
스키마 사이의 대응 관계를 정의하는 궁극적인 목적

→ 데이터 독립성의 실현