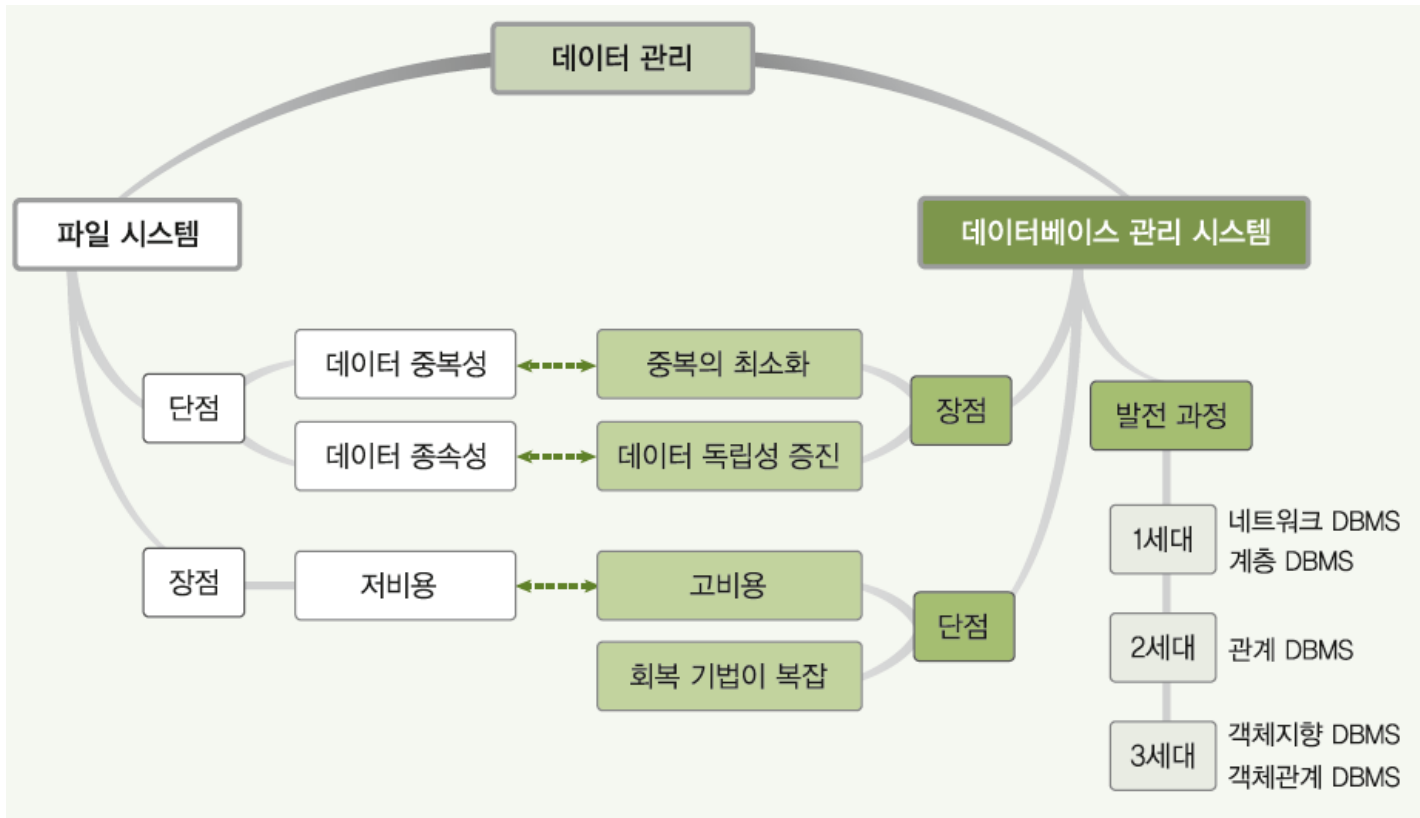




2장. 데이터베이스 관리 시스템

- 데이터베이스 관리 시스템의 등장 배경
- 데이터베이스 관리 시스템의 정의
- 데이터베이스 관리 시스템의 장단점
- 데이터베이스 관리 시스템의 발전 과정

학습목표



- ▶ 화일 시스템의 문제점과 데이터베이스 관리 시스템의 필요성을 알아본다.
- ▶ 데이터베이스 관리 시스템의 필수 기능을 살펴본다.
- ▶ 데이터베이스 관리 시스템의 장단점을 알아본다.
- ▶ 데이터베이스 관리 시스템의 발전 과정을 살펴본다.

01 데이터베이스 관리 시스템의 등장 배경

❖ 파일 시스템(file system)

- 데이터를 파일로 관리하기 위해 파일을 생성·삭제·수정·검색하는 기능을 제공하는 소프트웨어
- 응용 프로그램마다 필요한 데이터를 별도의 파일로 관리함



그림 2-1 파일 시스템에서의 데이터 관리

01 데이터베이스 관리 시스템의 등장 배경

❖ 화일 시스템의 문제점 해결

- 같은 내용의 데이터가 여러 화일에 중복 저장된다 → 데이터 중복성
 - 저장 공간의 낭비는 물론 데이터 일관성과 데이터 무결성을 유지하기 어려움



그림 2-1 파일 시스템에서의 데이터 관리

❖ 화일 시스템의 문제점

- 같은 내용의 데이터가 여러 화일에 중복 저장된다
- 응용 프로그램이 데이터 화일에 종속적이다
- 데이터 화일에 대한 동시 공유, 보안, 회복 기능이 부족하다
- 응용 프로그램 개발이 쉽지 않다

01 관계 데이터 모델의 개념

고객 릴레이션

고객아이디	고객이름	나이	등급	직업	적립금
apple	김현준	20	gold	학생	1000
banana	정소화	25	vip	간호사	2500
carrot	원유선	28	gold	교사	4500
orange	정지영	22	silver	학생	0

주문 릴레이션

주문번호	주문고객	고객이름	나이	등급	직업	적립금	주문제품	수량	단가	주문일자
1001	apple	김현준	20	gold	학생	1000	진짜우동	10	2000	2013-01-01
1002	banana	정소화	25	vip	간호사	2500	맛있는파이	5	500	2013-01-10
1003	carrot	원유선	28	gold	교사	4500	그대로만두	11	4500	2013-01-11

고객(고객아이디)를 참조하는
외래키

01 데이터베이스 관리 시스템의 등장 배경

❖ 파일 시스템의 문제점 해결

- 같은 내용의 데이터가 여러 파일에 중복 저장된다 → 데이터 중복성
 - 저장 공간의 낭비는 물론 데이터 일관성과 데이터 무결성을 유지하기 어려움
 - 체계적인 통합이 필요함

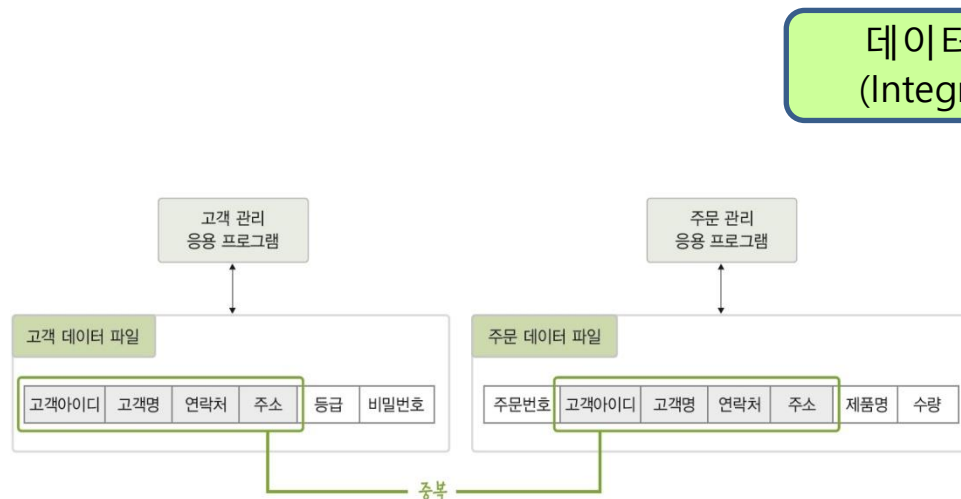


그림 2-1 파일 시스템에서의 데이터 관리

데이터 통합
(Integration)

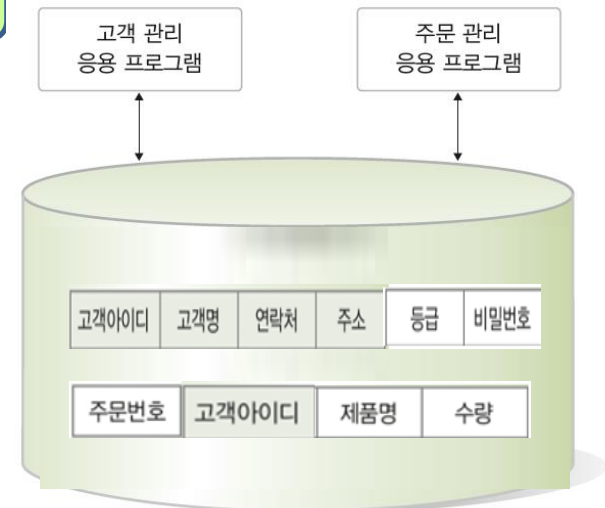


그림 2-2 파일 시스템의 데이터 중복성 문제에 대한 일차적 해결 방안

02 데이터베이스 관리 시스템의 정의

❖ 데이터베이스 관리 시스템

- DBMS(DataBase Management System)
- 화일 시스템의 문제를 해결하기 위해 제시된 소프트웨어
- 조직에 필요한 데이터를 데이터베이스에 통합하여 저장하고 관리함

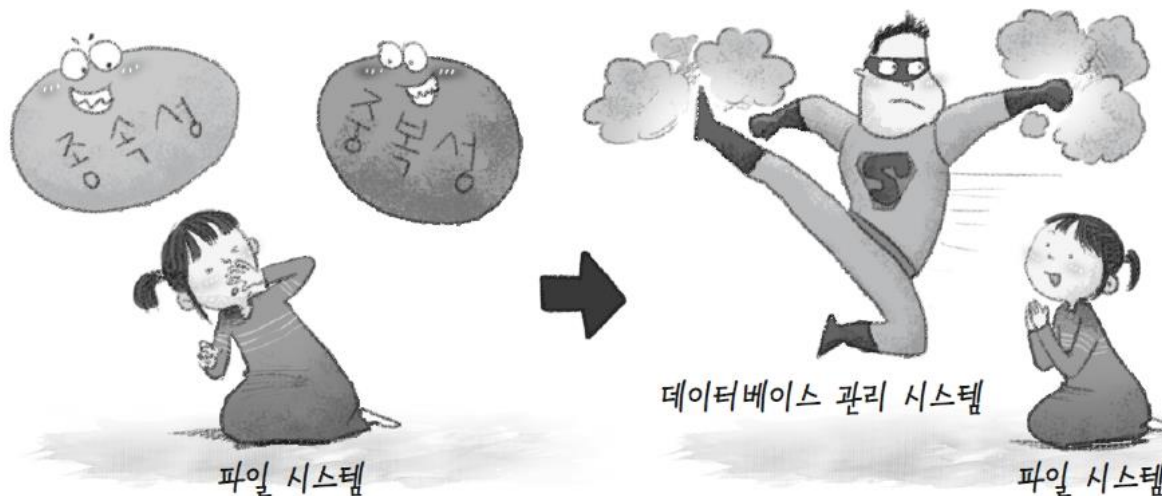
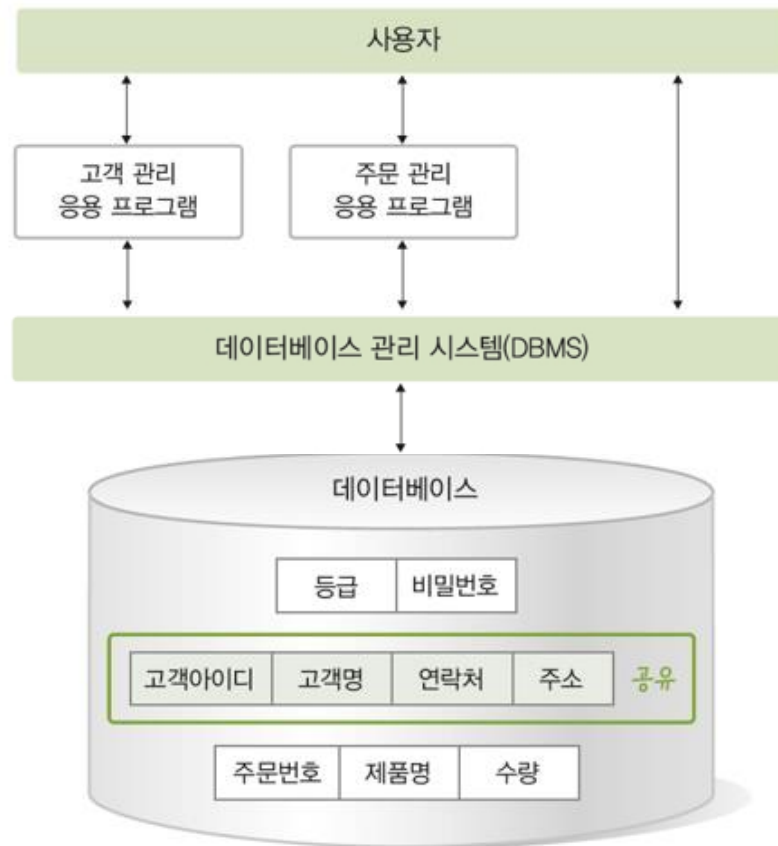


그림 2-4 파일 시스템과 데이터베이스 관리 시스템

02 데이터베이스 관리 시스템의 정의

❖ 데이터베이스 관리 시스템에서의 데이터 관리



02 데이터베이스 관리 시스템의 정의

❖ 데이터베이스 관리 시스템의 주요 기능

정의 기능	데이터베이스 구조를 정의하거나 수정할 수 있다.
조작 기능	데이터를 삽입·삭제·수정·검색하는 연산을 할 수 있다.
제어 기능	데이터를 항상 정확하고 안전하게 유지할 수 있다.

❖ 데이터베이스 언어

- 정의 기능
 - Data Definition Language : DDL
- 조작 기능
 - Data Manipulation Language : DML
- 정의기능
 - Data Control Language : DCL

03 데이터베이스 관리 시스템의 장단점

장점

- ☐ 데이터 중복을 통제할 수 있다
- ☐ 데이터 독립성이 확보된다
- ☐ 데이터를 동시 공유할 수 있다
- ☐ 데이터 보안이 향상된다
- ☐ 데이터 무결성을 유지할 수 있다
- ☐ 표준화할 수 있다
- ☐ 장애 발생 시 회복이 가능하다
- ☐ 응용 프로그램 개발 비용이 줄어든다

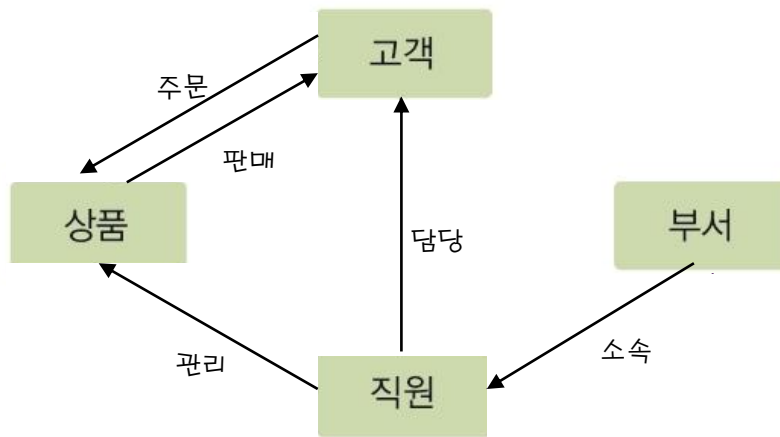
단점

- ☐ 비용이 많이 든다
- ☐ 백업과 회복 방법이 복잡하다
- ☐ 중앙 집중 관리로 인한 취약점이 존재한다

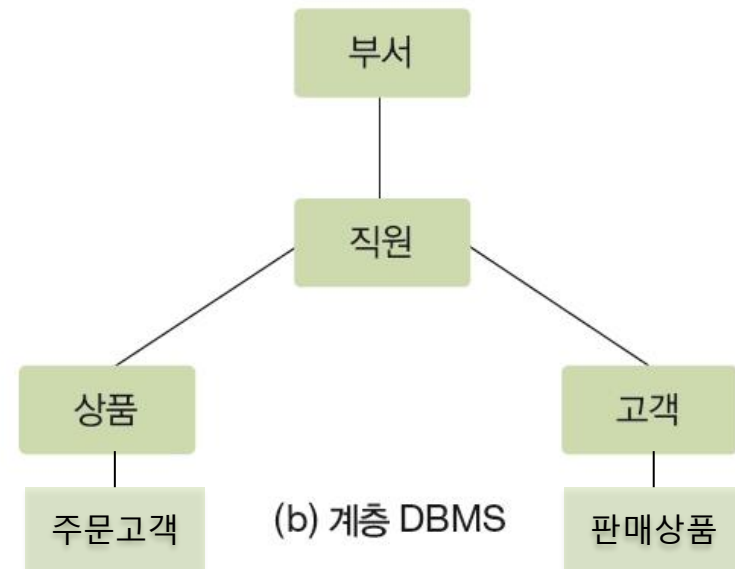
04 데이터베이스 관리 시스템의 발전 과정

❖ 1세대 : 네트워크 DBMS, 계층 DBMS

- 네트워크 DBMS : 데이터베이스를 그래프 형태로 구성
 - 예) IDS(Integrated Data Store)
- 계층 DBMS : 데이터베이스를 트리 형태로 구성
 - 예) IMS(Information Management System)



(a) 네트워크 DBMS



(b) 계층 DBMS

그림 2-8 1세대 DBMS 구조의 예

04 데이터베이스 관리 시스템의 발전 과정

❖ 2세대 : 관계 DBMS

- 관계 DBMS : 데이터베이스를 테이블 형태로 구성
 - 예) 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

04 데이터베이스 관리 시스템의 발전 과정

❖ 2세대 : 관계 DBMS

- 관계 DBMS : 데이터베이스를 테이블 형태로 구성
 - 예)

DBMS	제작사	작동 운영체제	기타
MySQL	Oracle	Unix, Linux, Windows, Mac	오픈 소스(무료), 상용
MariaDB	MariaDB	Unix, Linux, Windows	오픈 소스(무료), MySQL 초기 개발자들이 독립해서 만들
PostgreSQL	PostgreSQL	Unix, Linux, Windows, Mac	오픈 소스(무료)
Oracle	Oracle	Unix, Linux, Windows	상용 시장 점유율 1위
SQL Server	Microsoft	Windows	주로 중/대형급 시장에서 사용
DB2	IBM	Unix, Linux, Windows	메인프레임 시장 점유율 1위
Access	Microsoft	Windows	PC용
SQLite	SQLite	Android, iOS	모바일 전용, 오픈 소스(무료)

- 이 밖에 Informix, Sybase 등이 널리 쓰이고 있음

04 데이터베이스 관리 시스템의 발전 과정

❖ 3세대 : 객체지향 DBMS, 객체관계 DBMS

- 객체지향 DBMS : 객체를 이용해 데이터베이스를 구성
 - 예) 오투(O2), 온투스(ONTOS), 젬스톤(GemStone)
- 객체관계 DBMS : 객체 DBMS + 관계 DBMS

❖ 4세대

- 빅데이터 저장 및 처리기술 등장
- Hadoop과 NoSQL
 - 대부분 오픈소스이며 비정형 데이터 처리에 적합
 - 일관성 및 무결성 보다는 가용성과 확장성에 중점
 - Hbase, Casandra, MonggoDB, CouchDB



Thank You
