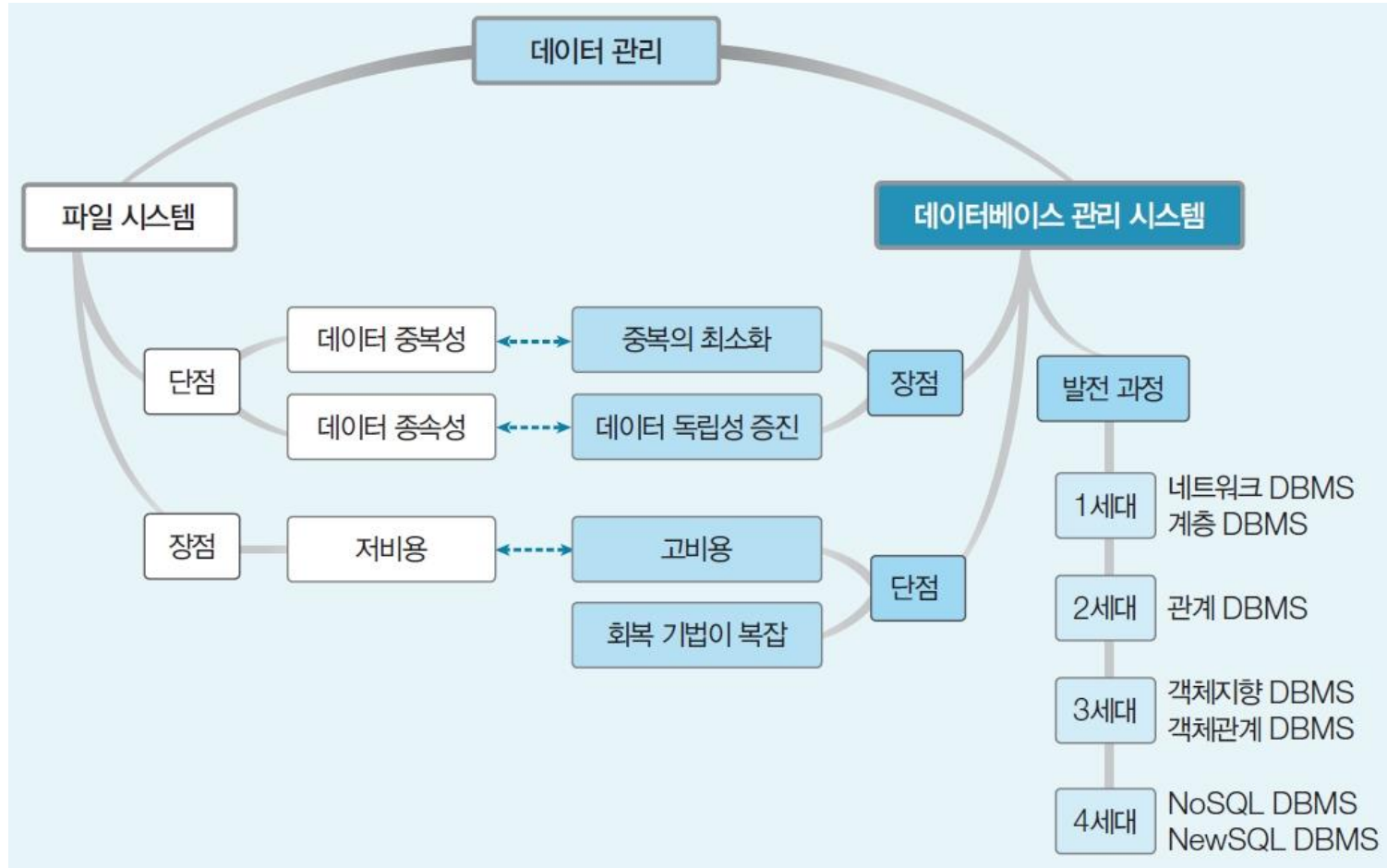


# 데이터베이스 관리 시스템

- 01 데이터베이스 관리 시스템의 등장 배경
- 02 데이터베이스 관리 시스템의 정의
- 03 데이터베이스 관리 시스템의 장·단점
- 04 데이터베이스 관리 시스템의 발전 과정

# 학습목표



- 파일 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 프로그램별로 필요한 데이터를 별도의 파일로 관리함



그림 2-1 파일 시스템에서의 데이터 관리

# 01 데이터베이스 관리 시스템의 등장 배경

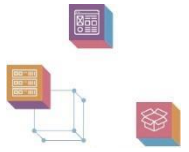
---



## ◆ 파일 시스템의 문제점

- 같은 내용의 데이터가 여러 파일에 중복 저장된다.
- 응용 프로그램이 데이터 파일에 종속적이다.
- 데이터 파일에 대한 동시 공유, 보안, 회복 기능이 부족하다.
- 응용 프로그램을 개발하기 쉽지 않다.

# 01 데이터베이스 관리 시스템의 등장 배경



## ◆ 파일 시스템의 주요 문제점

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



그림 2-1 파일 시스템에서의 데이터 관리

한 고객의 연락처를 고객 데이터 파일에서만 수정하고  
실수로 주문 데이터 파일에서 수정하지 않았다면?

# 01 데이터베이스 관리 시스템의 등장 배경



## ◆ 파일 시스템의 주요 문제점

- 데이터 중복성 문제를 해결하는 방법은?

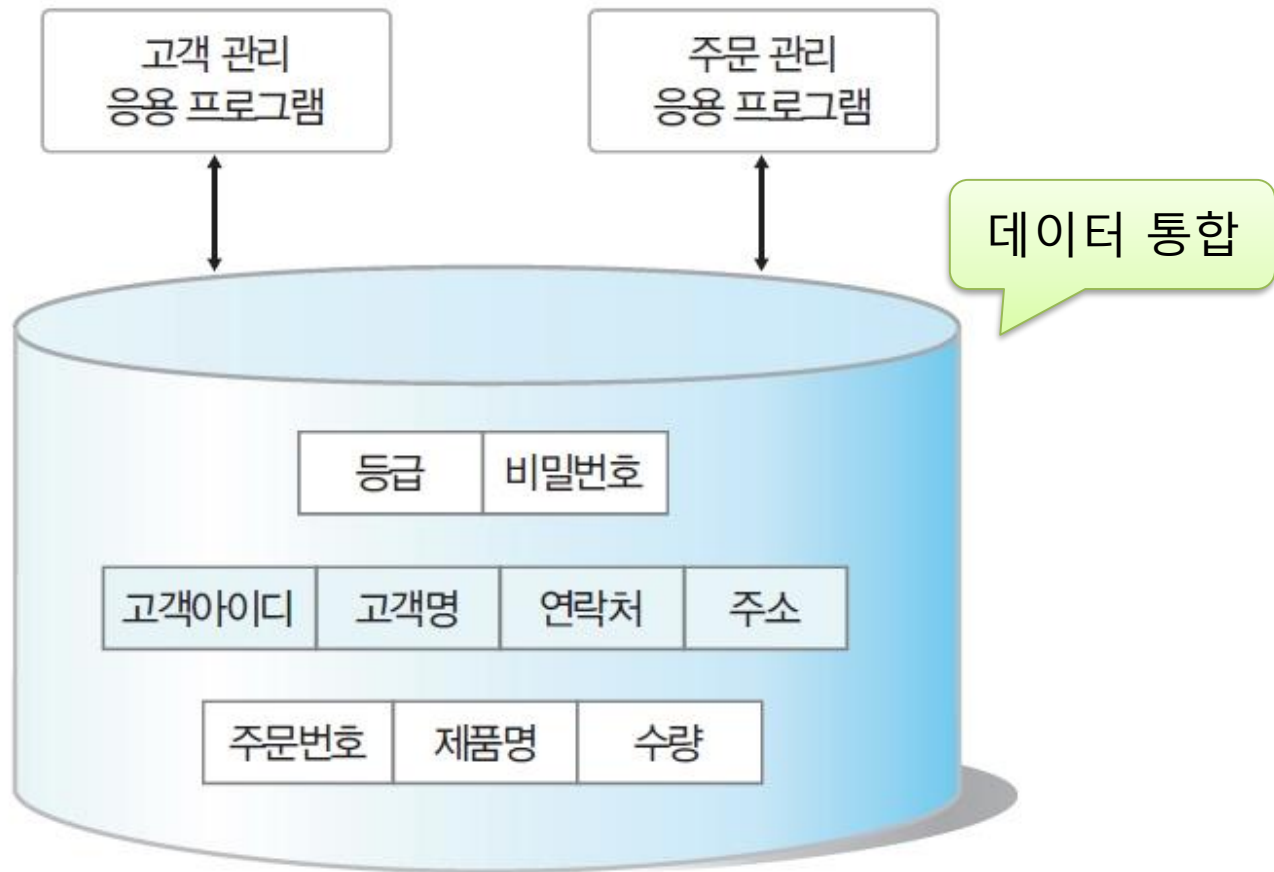


그림 2-2 파일 시스템의 데이터 중복성 문제를 해결하는 1차 방안

# 01 데이터베이스 관리 시스템의 등장 배경



## ◆ 파일 시스템의 주요 문제점

- 응용 프로그램이 데이터 파일에 종속적이다 → 데이터 종속성
  - 사용하는 파일의 구조를 변경하면 응용 프로그램도 함께 변경해야 함

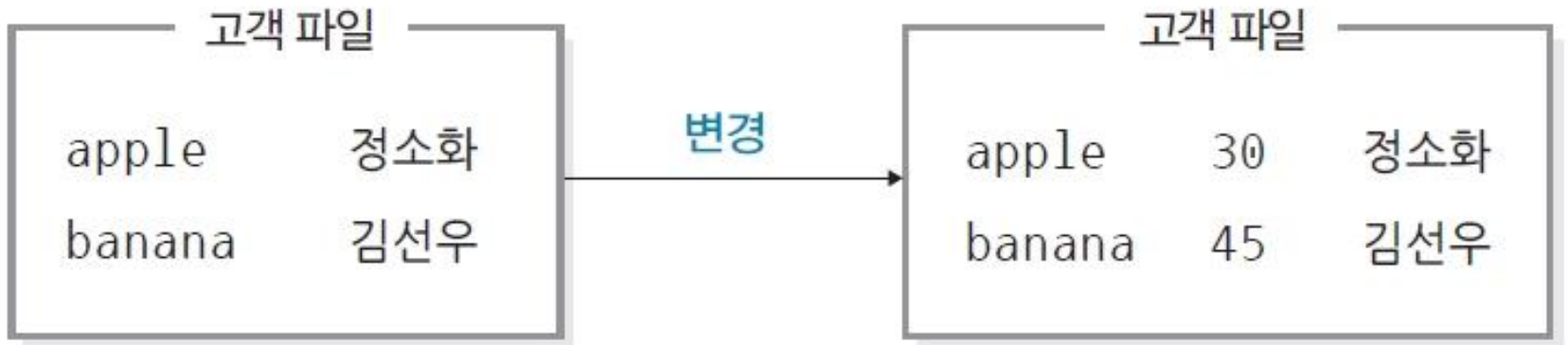


그림 2-3 파일 구조 변경 예

## 02 데이터베이스 관리 시스템의 정의



### ◆ 데이터베이스 관리 시스템

- DBMS(DataBase Management System)
- 파일 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

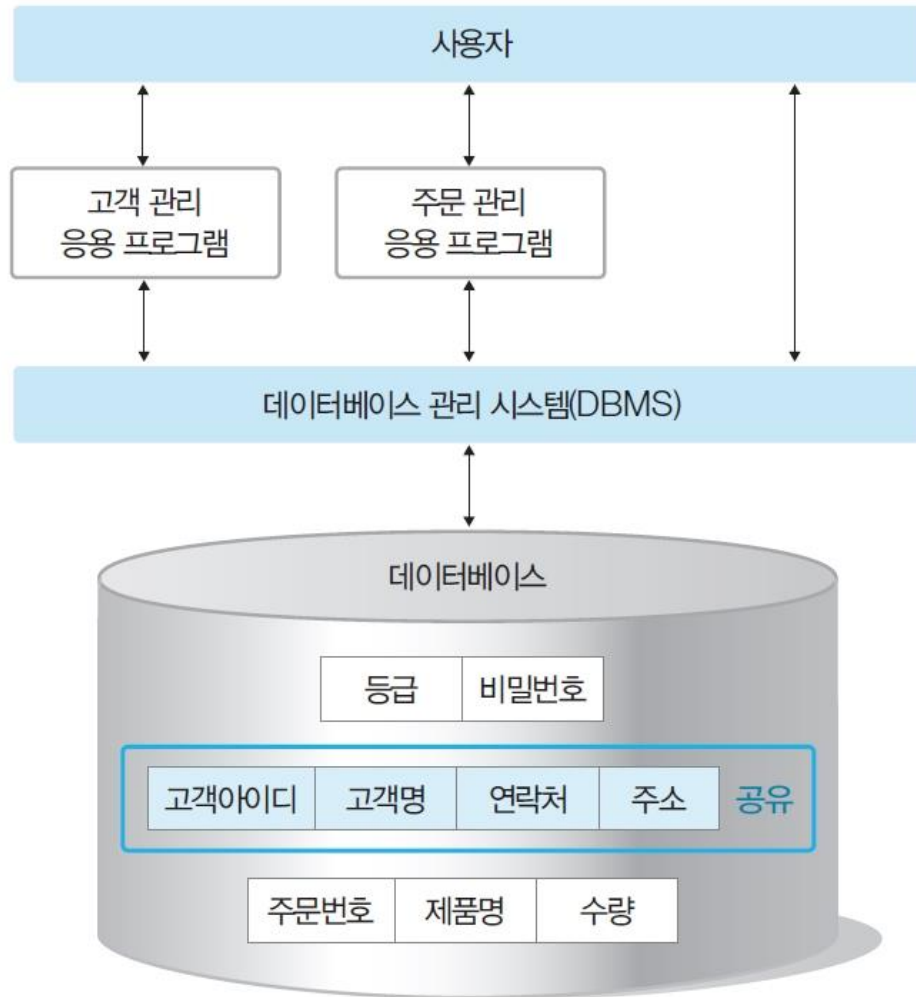


그림 2-5 데이터베이스 관리 시스템에서의 데이터 관리

## 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.

그림 2-6 데이터베이스 관리 시스템의 주요 기능

# 03 데이터베이스 관리 시스템의 장·단점



## ◆ 장점

- 데이터 중복을 통제할 수 있다.
  - 데이터베이스에 데이터를 통합하여 관리하므로 데이터 중복 문제 해결
- 데이터 독립성이 확보된다.
  - 응용 프로그램 대신 데이터베이스에 접근하고 관리하는 모든 책임 담당  
→ 응용 프로그램과 데이터베이스 사이에 독립성이 확보됨
- 데이터를 동시 공유할 수 있다.
  - 동일한 데이터를 여러 응용 프로그램이 공유하여 동시 접근할 수 있게 지원  
→ 동시 접근 제어 기술 보유
- 데이터 보안이 향상된다.
  - 중앙 집중식으로 데이터를 관리하므로 효율적인 접근 제어 가능
    - 권한이 없는 사용자의 접근, 허용되지 않은 데이터와 연산에 대한 요청 차단

# 03 데이터베이스 관리 시스템의 장·단점



## ◆ 장점

- 데이터 무결성을 유지할 수 있다.
  - 데이터 삽입·수정 등의 연산이 수행될 때마다 유효성을 검사하여 데이터 무결성(정확성)을 유지
- 표준화할 수 있다.
  - 데이터베이스 관리 시스템이 정한 표준화된 방식을 통해 데이터베이스에 접근
- 장애 발생 시 회복이 가능하다.
  - 데이터 일관성과 무결성을 유지하면서 장애 발생 이전 상태로 데이터를 복구하는 회복 기능 지원
- 응용 프로그램 개발 비용이 줄어든다.
  - 파일 시스템을 사용할 때보다 데이터 관리 부담이 줄어 응용 프로그램 개발 비용 및 유지 보수 비용이 줄어듦

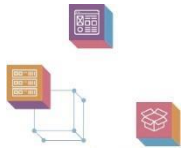
# 03 데이터베이스 관리 시스템의 장·단점



## ◆ 단점

- 비용이 많이 든다.
  - 별도 구매 비용이 들고, 동시 사용이 허용되는 사용자 수에 따라 가격 증가
- 백업과 회복 방법이 복잡하다.
  - 장애 발생의 원인과 상태를 정확히 파악하기 어렵고 회복 방법도 복잡함
- 중앙 집중 관리로 인한 취약점이 존재한다.
  - 데이터베이스나 데이터베이스 관리 시스템에 장애가 발생하면 전체 시스템의 업무 처리가 중단됨
  - 데이터베이스 의존도가 높은 시스템일수록 가용성과 신뢰성에 치명적임

# 03 데이터베이스 관리 시스템의 장·단점



## 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

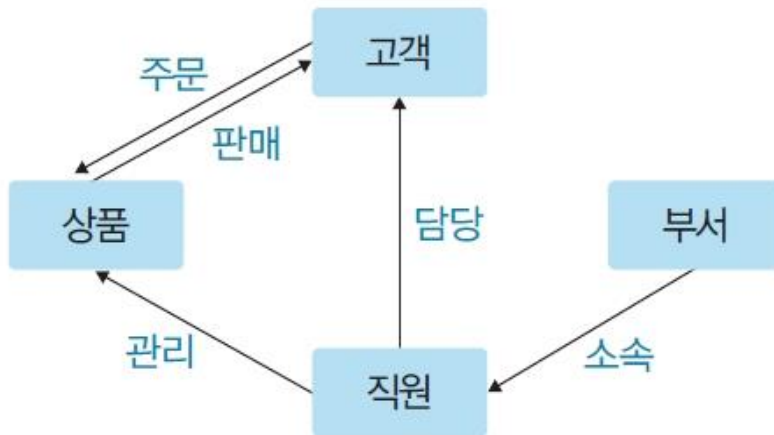
그림 2-7 데이터베이스 관리 시스템의 장점과 단점

# 04 데이터베이스 관리 시스템의 발전 과정

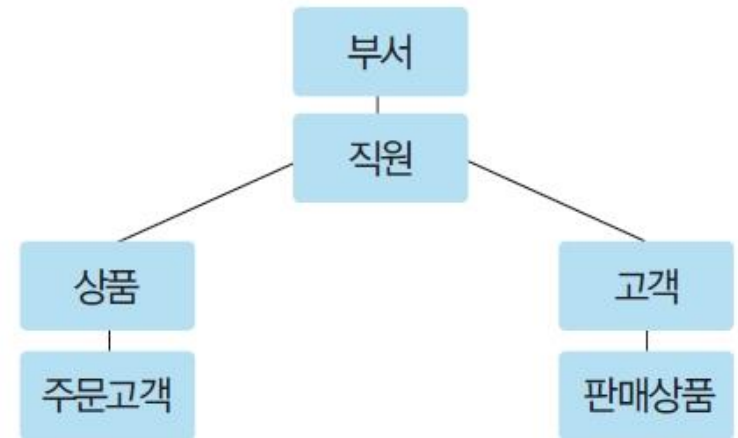


## ◆ 1세대 : 네트워크 DBMS, 계층 DBMS

- 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, 마리아DB(MariaDB)

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



## ◆ 3세대 : 객체지향 DBMS, 객체관계 DBMS

- 객체지향 DBMS : 객체를 이용해 데이터베이스를 구성
  - 예) 오투(O2), 온투스(ONTOS), 젬스톤(GemStone)
- 객체관계 DBMS : 객체지향 DBMS + 관계 DBMS
  - 예) 관계 DBMS 제품들이 객체지향 기능을 지원하면서 객체관계 DBMS로 분류되기도 함(오라클이 대표적)

# 04 데이터베이스 관리 시스템의 발전 과정



## ◆ 4세대 : NoSQL • NewSQL DBMS

- NoSQL DBMS : 비정형 데이터를 처리하는데 적합하고 확장성이 뛰어남
  - 안정성과 일관성 유지를 위한 복잡한 기능 포기
  - 데이터 구조를 미리 정해두지 않는 유연성을 가짐
  - 여러 대의 컴퓨터에 데이터를 분산하여 저장하고 처리하는 환경에서 주로 사용
  - 예) 몽고디비(MongoDB), H베이스(HBase), 카산드라(Cassandra), 레디스(Redis), 네오포제이(Neo4j), 오리엔트DB(OrientDB) 등
- NewSQL DBMS: 관계 DBMS의 장점 + NoSQL의 확장성 및 유연성
  - 정형 및 비정형 데이터를 안정적이고 빠르게 처리 가능
  - 예) 구글 스패너(Spanner), 볼트DB(VoltDB), 누오DB(NuoDB)

# 04 데이터베이스 관리 시스템의 발전 과정

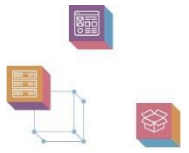


그림 2-10 DBMS의 발전 과정