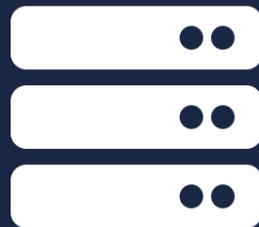


CHAPTER 02

# 데이터베이스 관리 시스템



*Database Management System (DBMS)*

---

**박상돈 조교수**

대전대학교 컴퓨터공학과



# 목차

01

## DBMS의 등장 배경

파일 시스템의 문제점과 데이터 관리의 한계

02

## DBMS의 정의

DBMS의 개념, 역할, 주요 기능 (정의·조작·제어)

03

## DBMS의 장·단점

데이터 독립성, 동시 공유, 무결성, 비용 문제

04

## DBMS의 발전 과정

1세대(네트워크·계층) → 2세대(관계) → 3세대(객체) → 4세대(NoSQL·NewSQL)

# 01

---

## DBMS의 등장 배경

*Background: Why File Systems Were Not Enough*

# 파일 시스템(File System)의 데이터 관리



**파일 시스템:** 과거에 데이터를 관리하기 위해 사용하던 소프트웨어. 각 응용 프로그램이 독립적인 데이터 파일을 별도로 유지·관리하는 방식.

고객 관리  
프로그램



고객 파일  
(고객ID, 이름, 연락처)

주문 관리  
프로그램



주문 파일  
(주문ID, 고객ID, 이름)

배송 관리  
프로그램



배송 파일  
(배송ID, 고객ID, 이름, 주소)



## 핵심 문제

고객 이름이 3개 파일에 중복 저장됨!

- 한 곳만 수정하면 불일치 발생
- 저장 공간 낭비
- 데이터 일관성 훼손

# 파일 시스템의 4가지 주요 문제점



## 데이터 중복성 (Redundancy)

같은 데이터가 여러 파일에 중복 저장되어 데이터 일관성(Consistency)과 무결성(Integrity) 유지가 어려움. 한 파일만 수정하면 다른 파일과 불일치 발생.



## 데이터 종속성 (Dependency)

응용 프로그램이 데이터 파일의 구조에 종속됨. DB 구조가 변경되면 모든 관련 응용 프로그램을 함께 수정해야 하는 부담 발생.



## 동시 공유·보안·회복 부족

다수의 프로그램이 동시에 데이터를 공유하기 어려움. 보안 기능이 제한적이고, 장애 발생 시 데이터 복구 (Recovery) 기능이 부족함.



## 응용 프로그램 개발 난이도

사용자 요구(읽기/삽입/삭제 등)에 맞는 프로그램을 매번 새로 개발해야 함. 유지보수 비용 증가 및 개발 생산성 저하.

# 02

---

## DBMS의 정의

*Definition of Database Management System*

# DBMS란 무엇인가?

## DBMS (Database Management System)

조직에 필요한 데이터를 통합 저장하고, 이에 대한 관리를 집중적으로 담당하는 소프트웨어 시스템



### Before: 파일 시스템

- 각 프로그램이 별도 데이터 파일 관리
- 데이터 중복 및 불일치 발생
- DB 구조 변경 시 프로그램 수정 필요
- 동시 접근·보안·회복 기능 부족
- 프로그램과 데이터가 종속적 관계



### After: DBMS 도입

- 데이터를 통합 저장·관리
- 데이터 중복 최소화 및 일관성 보장
- DBMS가 접근을 중재 → 데이터 독립성
- 동시 공유, 보안, 회복 기능 제공
- CRUD 연산이 용이해짐

# DBMS 기반 데이터 관리 구조



## 핵심 포인트

DBMS가 응용 프로그램과 DB 사이를 중재하여

데이터 독립성을 확보합니다.

DB 구조가 바뀌어도 응용 프로그램은 수정하지 않아도 됩니다.

# DBMS의 3대 주요 기능



## 정의 기능 (Definition)

데이터베이스 구조를  
생성·수정·삭제하는 기능

- 데이터 형식, 구조, 제약 조건 등 명세
- 테이블, 뷰, 인덱스 등의 스키마 정의
- DDL(Data Definition Language) 사용
- 예: CREATE TABLE, ALTER TABLE, DROP TABLE



## 조작 기능 (Manipulation)

데이터를 삽입·삭제·수정·검색하는 기능

- 사용자-DB 간의 인터페이스 수단 제공
- CRUD 연산을 체계적으로 수행
- DML(Data Manipulation Language) 사용
- 예: SELECT, INSERT, UPDATE, DELETE



## 제어 기능 (Control)

데이터의 정확성·안전성을 유지하는 기능

- 동시성 제어 (Concurrency Control)
- 접근 권한 관리 및 보안 (Security)
- 무결성 제약 조건 검사 (Integrity)
- 장애 시 데이터 회복 (Recovery)

# 03

---

## DBMS의 장·단점

*Advantages and Disadvantages of DBMS*

# DBMS의 장점 (1/2)



## 데이터 중복 통제

데이터를 통합 관리하여 중복을 최소화하고 데이터 일관성(Consistency)을 유지합니다. 저장 공간 절약 효과도 있습니다.



## 데이터 독립성 확보

DBMS가 응용 프로그램 대신 DB에 접근하므로, DB 구조가 변경되어도 응용 프로그램을 수정할 필요가 없습니다.



## 데이터 동시 공유

여러 사용자·프로그램이 동시에 같은 데이터를 공유할 수 있습니다. 동시성 제어를 통해 데이터 충돌을 방지합니다.



## 데이터 무결성 유지

데이터 연산 시 유효성을 자동 검사하여 무결성 제약 조건을 만족하는 데이터만 DB에 저장됩니다.

# DBMS의 장점 (2/2)



## 데이터 표준화

데이터 접근 방식, 형식 및 구조를 조직 전체에 걸쳐 통일할 수 있습니다. 부서 간 데이터 교환이 원활해집니다.



## 장애 시 회복 가능

트랜잭션 로그(Log)를 기반으로 장애 발생 이전 상태로 복구하는 기능을 제공합니다. UNDO/REDO 연산을 통한 회복이 가능합니다.



## 응용 프로그램 개발 비용 절감

파일 시스템 대비 유지보수 비용이 크게 절감됩니다. 데이터 접근 로직을 DBMS가 담당하므로 개발자는 비즈니스 로직에 집중할 수 있습니다.

# DBMS의 단점

## 높은 비용

DBMS 소프트웨어를 별도 구매·설치해야 하며, 하드웨어 및 전문 인력 비용도 발생합니다. 다만, MySQL, PostgreSQL 등 오픈소스 DBMS의 활용으로 비용을 절감할 수 있습니다.



## 복잡한 백업과 회복

장애 발생에 대비하여 주기적으로 DB를 백업해야 합니다. 장애 유형에 따라 회복 절차가 다르고 복잡하며, 전문적인 관리 지식이 필요합니다.



## 중앙 집중 관리의 취약점

모든 데이터가 하나의 DBMS에 집중되므로, DBMS에 장애가 발생하면 전체 시스템의 가용성(Availability)과 신뢰성(Reliability)에 치명적인 영향을 미칩니다.

# 04

---

## DBMS의 발전 과정

*Evolution of Database Management Systems*

# DBMS 발전 과정 타임라인



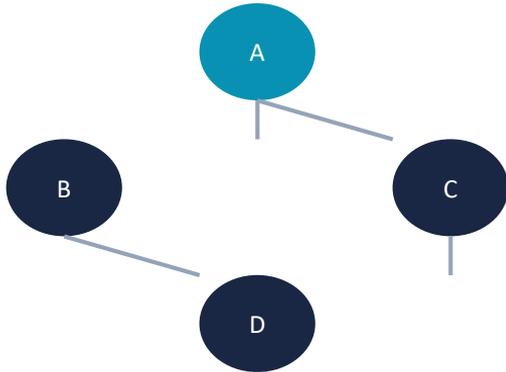
**발전 방향:** 복잡한 구조 → 단순한 테이블 → 객체 지원 → 비정형 데이터 대응

- 1세대: 데이터 간 관계를 네트워크/트리 구조로 표현 → 복잡하고 구조 변경 어려움
- 2세대: 테이블(관계) 기반으로 단순화 → SQL 표준 도입으로 범용적 사용 가능
- 3세대: 객체지향 프로그래밍과 결합 → 복잡한 데이터 타입 지원
- 4세대+: 빅데이터·SNS 시대 → 비정형 데이터 처리와 수평 확장성(Scale-Out) 강조

# 1세대: 네트워크-계층 DBMS (1960s)



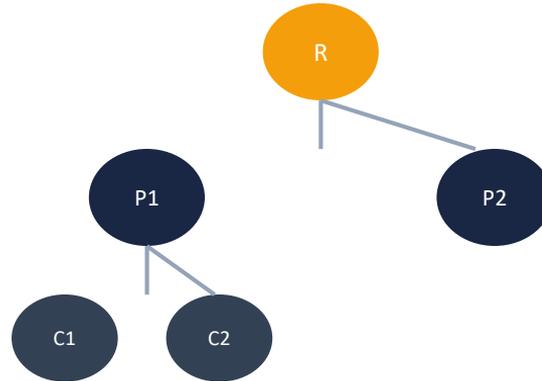
## 네트워크 DBMS



- 간선(Edge)으로 데이터 관계 표현
- 다대다(M:N) 관계 표현 가능
- 구조가 복잡하고 변경이 어려움
- 1960년 초반, IDS



## 계층 DBMS



- 트리(Tree) 구조로 부모-자식 관계 표현
- 1:N 관계만 표현 가능 (M:N 불가)
- 구조 변경이 어려움
- 1960년 후반, IMS (IBM)

## 2세대: 관계 DBMS — RDBMS (1980s)

관계 데이터 모델(Relational Data Model)을 기반으로 데이터베이스를 **테이블(Table) 형태**로 구성하는 DBMS

예시: 학생 테이블 (관계형 표현)

학번	이름	학과	학년
20210001	김철수	컴퓨터공학	3
20210002	이영희	전자공학	2

### SQL (Structured Query Language)

관계형 DB의 표준 언어로, 데이터 정의(DDL), 조작(DML), 제어(DCL) 기능을 모두 제공합니다. 1970년 E.F. Codd가 관계형 모델을 제안하고, 1980년대에 상용화되면서 DB의 주류가 되었습니다.

### 핵심 특징

- 단순한 테이블 구조
- SQL 표준 언어 사용
- 데이터 독립성 우수
- ACID 트랜잭션 지원
- 가장 널리 사용되는 모델

### 대표 제품:

Oracle, MySQL, PostgreSQL,  
MS SQL Server, MariaDB

# 3세대: 객체지향-객체관계 DBMS (1990s)



## 객체지향 DBMS (OODBMS)

객체지향 프로그래밍(OOP)의 개념을 데이터베이스에 적용한 모델

- 데이터를 객체(Object)로 저장·관리
- 상속, 캡슐화, 다형성 등 OOP 특성 지원
- 멀티미디어, CAD 등 복잡한 데이터에 적합
- 관계형 모델 대비 복잡도가 높음

시기: 1980년 후반

제품: O2, ONTOS, GemStone



## 객체관계 DBMS (ORDBMS)

OODBMS + RDBMS를 통합한 하이브리드 모델

- 관계형 DB의 테이블 구조 유지
- 객체지향의 사용자 정의 타입 지원
- 개발자가 데이터 형식·방법을 자유롭게 정의하여 DBMS에 통합 가능
- SQL을 확장하여 객체 기능 지원

시기: 1990년 후반

제품: Oracle, PostgreSQL

# 4세대+: NoSQL DBMS (2010s~)

**등장 배경:** SNS의 폭발적 성장으로 사진, 동영상, 검색 로그 등 대량의 비정형 데이터가 생산됨 → RDBMS만으로는 비효율적



## NoSQL (Not Only SQL) DBMS

비정형 데이터 처리에 특화된 DBMS — 고정된 스키마 없이 유연한 데이터 모델 제공

### Key-Value

Redis, DynamoDB

### Document

MongoDB, CouchDB

### Column-Family

Cassandra, HBase

### Graph

Neo4j, ArangoDB

**특징:** 수평 확장(Scale-Out) 용이 · 유연한 스키마 · 대용량 데이터 처리에 최적화

**한계:** ACID 트랜잭션 완전 지원이 어려움 · 정형 데이터 기반 DBMS와의 통합 문제

# 4세대+: NewSQL DBMS (2011~)

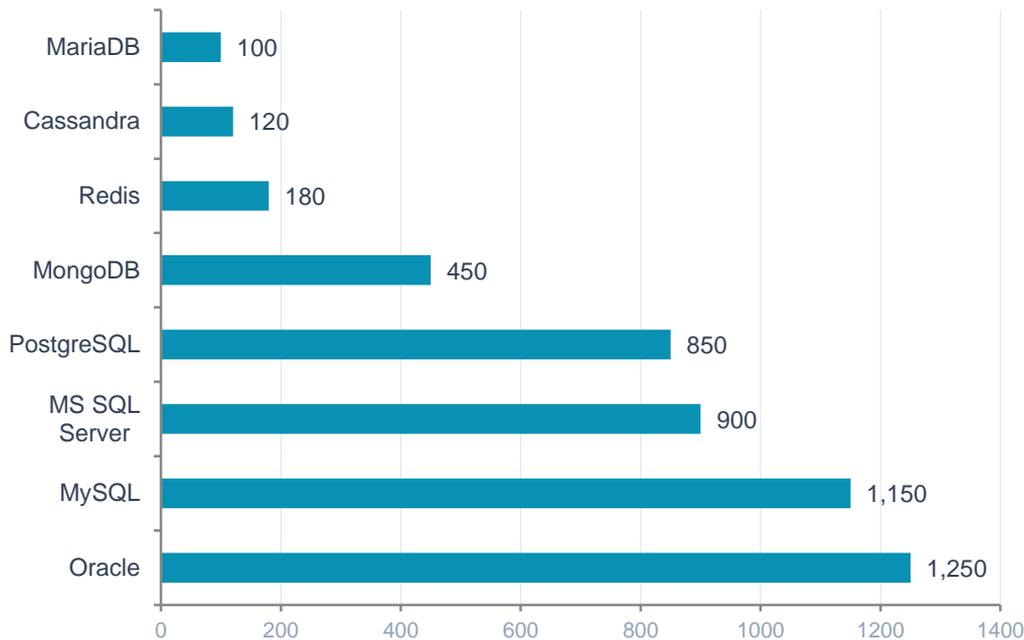
**NewSQL:** RDBMS의 ACID 보장 + NoSQL의 수평 확장성·유연성을 결합한 차세대 DBMS

구분	RDBMS	NoSQL	NewSQL
데이터 모델	테이블 (관계형)	다양 (KV, Doc 등)	테이블 (관계형)
스키마	고정 스키마	유연한 스키마	고정 스키마
ACID 트랜잭션	완전 지원	부분 지원	완전 지원
확장성	수직 확장 (Scale-Up)	수평 확장 (Scale-Out)	수평 확장 (Scale-Out)
SQL 지원	표준 SQL	자체 API	표준 SQL
대표 제품	Oracle, MySQL	MongoDB, Redis	Spanner, VoltDB

NewSQL의 대표적인 제품으로는 Google Spanner, VoltDB, NuoDB, CockroachDB, TiDB 등이 있으며, 클라우드 네이티브 환경에서 주목받고 있습니다.

# 현재 DBMS 인기도 (DB-Engines Ranking)

출처: [db-engines.com/en/ranking](https://db-engines.com/en/ranking)



## 분류

### 유료 DBMS

- 1위 Oracle
- 2위 MS SQL Server

### 무료/오픈소스 DBMS

- 1위 MySQL
- 2위 PostgreSQL

### NoSQL DBMS

- 1위 MongoDB
- 2위 Redis

# 요약 (Wrap-Up)

01

## DBMS의 등장 배경

파일 시스템의 데이터 중복성, 종속성, 동시 공유 부족, 보안·회복 기능 부재 등의 한계를 극복하기 위해 DBMS가 등장하였습니다.

02

## DBMS의 정의

DBMS는 조직의 데이터를 통합 저장·관리하는 소프트웨어로, 정의(DDL)·조작(DML)·제어(DCL) 기능을 제공하며 데이터 독립성을 확보합니다.

03

## DBMS의 장·단점

장점: 중복 통제, 독립성, 동시 공유, 무결성, 표준화, 회복, 비용 절감. 단점: 높은 초기 비용, 복잡한 백업/회복, 중앙 집중 취약점.

04

## DBMS의 발전 과정

1세대(네트워크·계층) → 2세대(관계형, SQL) → 3세대(객체지향·객체관계) → 4세대+(NoSQL·NewSQL)로 발전하였습니다.



# 감사합니다

*Thank You*

---

**박상돈 조교수**

대전대학교 컴퓨터공학과