# 도입

## 데이터베이스와 DBMS

컴퓨터에 체계적으로 저장한 데이터를 **데이터베이스(Database)**라 하며, 데이터베이스를 관리하는 시스템을 **DBMS(DataBase Management System)**라 한다. 'DBMS'와 '데이터베이스'라는 용어는 크게 구분하지 않고 사용된다.

### 파일 시스템과 데이터베이스의 비교

**파일 시스템**은 원시 데이터 파일을 컴퓨터의 하드 디스크 등에 저장하는 시스템이다. 중복 데이터가 많이 발생하고 데이터의 일관성이 떨어지며 보안, 백업·복구가 불편한 문제가 있었다.

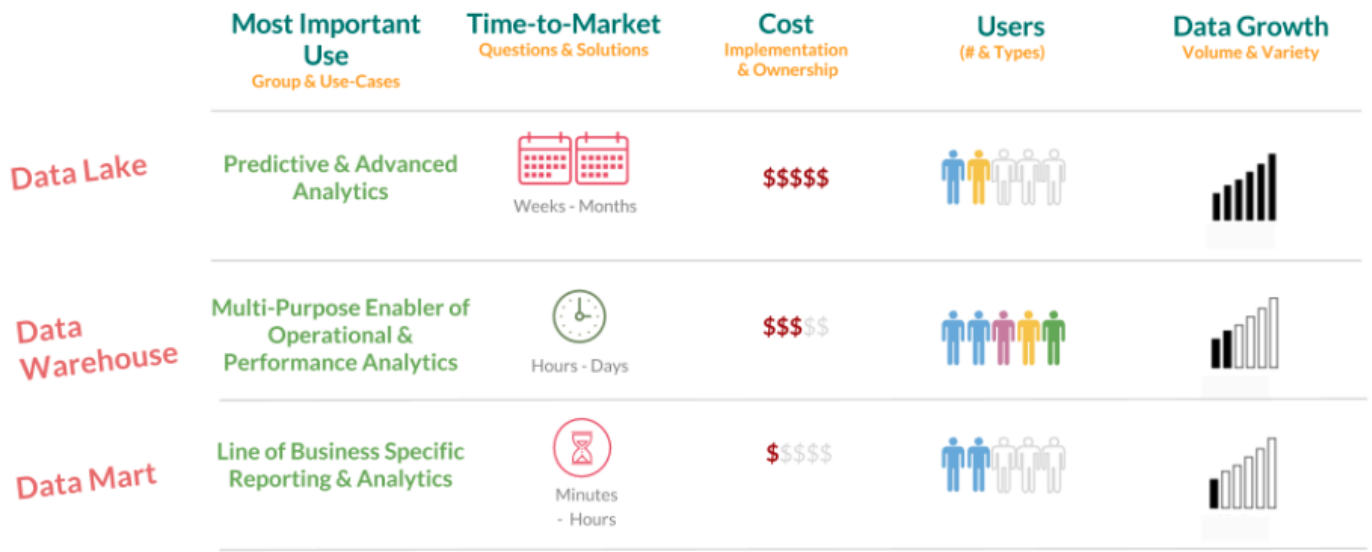
데이터베이스는 그러한 파일 시스템의 단점을 보완하고 데이터의 모델링, 무결성, 다수 사용자를 위한 동시성 제어 등을 제공한다.

참고로 이 책에서 다루는 SQLite는 하나의 데이터베이스를 한 개의 파일에 저장하지만, 모든 DBMS가 그런 것은 아니다. 데이터베이스를 여러 개의 파일에 저장하거나 아예 파일 시스템을 이용하지 않고 디스크에 직접 기록하는 기능을 지원하는 DBMS도 있다.[1]

### 데이터 레이크, 데이터 웨어하우스, 데이터 마트

**데이터 레이크**, **데이터 웨어하우스**, **데이터 마트** 등은 데이터베이스와 관련이 있기는 하지만, 각자 다른 의미를 갖는 용어다. 이 책에서 다루는 주제와 큰 관련이 없지만 간단히 알아보자.

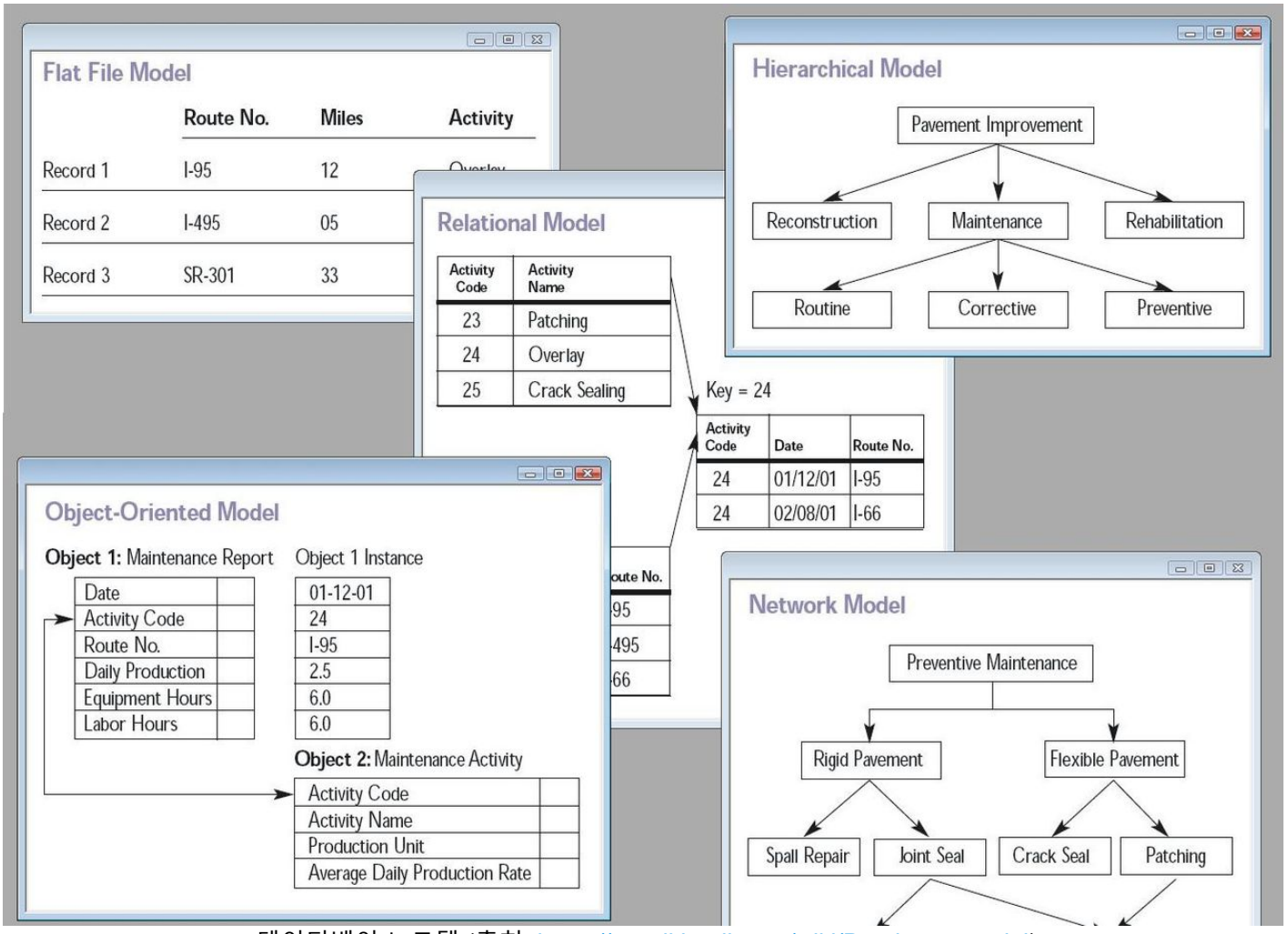
- 데이터 레이크(Data Lake): 정형 및 비정형(소셜, 센서, 이미지, 동영상 등)의 다양한 형태의 원시 데이터를 모은 저장소의 집합.
- 데이터 웨어하우스(Data Warehouse): 데이터 분석을 효율적으로 수행하기 위한 OLAP(온라인 분석 처리) 데이터베이스와 같이, 데이터베이스로부터 가져온 데이터의 계층을 생성한다. 주제 중심적이고 비휘발성의 특징을 갖는다.
- 데이터 마트(Data Mart): 특정 부서의 의사 결정 지원을 목적으로 하는 부서별 또는 부분별 데이터 웨어하우스. 분석 요건을 중심으로 한 요약 데이터로 구성된다.




데이터 레이크, 데이터 웨어하우스, 데이터 마트 (출처)

# 데이터베이스의 종류

다시 데이터베이스에 대한 이야기로 돌아가자. 데이터베이스는 데이터를 바라보는 관점에 따라 **관계형 데이터베이스**, **계층형 데이터베이스**, **그래프 데이터베이스** 등으로 나눌 수 있다.




데이터베이스 모델 (출처: https://en.wikipedia.org/wiki/Database_model)

## 관계형 데이터베이스(Relational Database)

관계형 데이터베이스는 데이터를 관계(relation)로 나타낸다.

일반적으로 DBMS라고 하면 RDBMS(Relational DBMS)를 가리킨다. 오라클 데이터베이스 서버, 마이크로소프트 SQL 서버, MySQL과 MariaDB, PostgreSQL 등이 이에 해당한다. 이 책에서 다루는 SQLite도 RDBMS이다.

- SQLite: SQLite는 가장 널리 사용되는 데이터베이스 엔진으로[2], 임베디드 디바이스, 사물 인터넷, 데이터 분석, 작은 규모의 웹사이트에 사용하기 적합하다.[3] SQLite의 특징은 다음과 같다.[4]
    - SQLite는 임베디드 SQL 데이터베이스 엔진으로, 독립적인 서버 프로세스를 갖지 않는다.
    - 설치 과정이 없고, 설정 파일도 존재하지 않는다.
    - 테이블, 인덱스, 트리거, 뷰 등을 포함한 완전한 데이터베이스가 디스크 상에 단 하나의 파일로 존재한다.
    - 퍼블릭 도메인[5]으로서 개인적 또는 상업적 목적으로 사용할 수 있다.
- 오라클(Oracle) 데이터베이스: 이 책을 참조.
- 포스트그레스큐엘(PostgreSQL): 이 곳을 참조.

## 계층형 데이터베이스(Hierachical Database)

계층형 데이터베이스는 데이터를 계층적인 트리(tree)로 표현한다. 데이터는 레코드(record)로서 저장되며, 레코드들은 링크(link)를 통해 연결된다. 레코드는 필드(field)들의 모음이며, 필드는 단일한 값을 갖는다. (위키백과)

### 그래프 데이터베이스(Graph Database)

그래프 데이터베이스는 데이터를 그래프 형태로 표현한다.

- Neo4J
- Amazon Neptune
- TigerGraph
- Oracle Graph Database
- Azure Cosmos DB (Gremlin API)

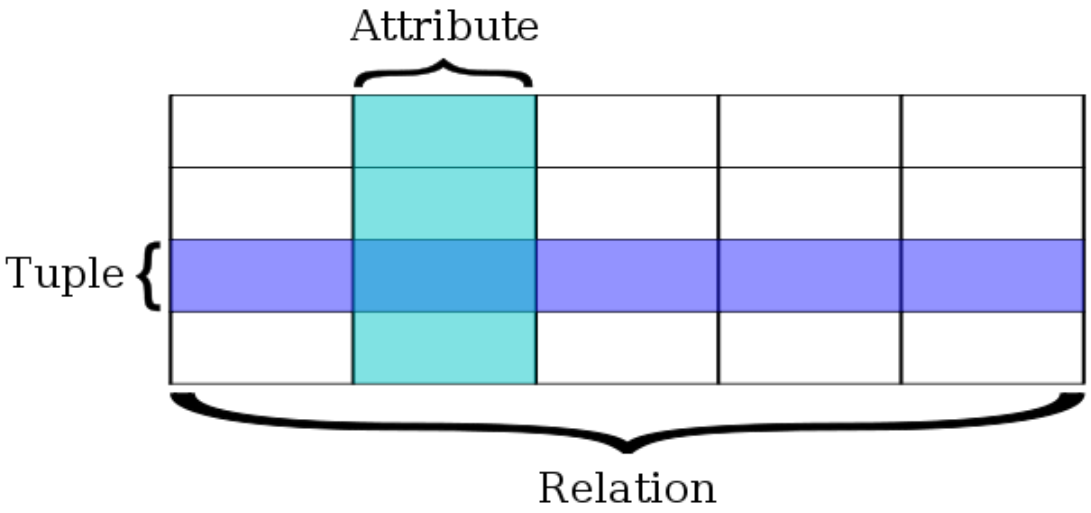
tip

**NoSQL**
근래에 들어 관계형 데이터베이스 모델을 벗어난 몽고DB(MongoDB), 카산드라(Cassandra), Neo4J 등을 NoSQL로 통칭한다.

## relation, tuple, attribute

관계형 데이터베이스에서는 데이터를 '관계'로 나타내며, 이는 '테이블'로 구현된다. '행'과 '열'에 해당하는 것을 '튜플'과 '속성'이라는 용어로 가리킨다. '레코드'와 '필드'라는 용어도 많이 사용한다. 이 책에서 주로 사용하는 용어를 굵게 표시했다.

- relation(관계, 릴레이션) = table(**테이블**)
- tuple(튜플)= row(**행**, 로우) = record(레코드)
- attribute(속성, 어트리뷰트) = column(열, **컬럼**, 칼럼) = field(**필드**)




(출처: 위키백과)

## 스프레드시트와 데이터베이스의 비교

마이크로소프트 엑셀(Microsoft Excel)이나 구글 스프레드시트(Google Sheets) 같은 스프레드시트와 (관계형) 데이터베이스의 차이는 다음과 같다.[6]

| 스프레드시트 | 데이터베이스 |
| --- | --- |

| 스프레드시트 | 데이터베이스 |
| --- | --- |
| 소프트웨어 애플리케이션이다 | SQL과 같은 질의 언어(query language)를 사용해 액세스하는 데이터 스토어(data store)다 |
| 행(row)과 열(column)의 형식으로 데이터를 구조화한다 | 규칙(rule)과 관계(relationship)로 데이터를 구조화한다 |
| 정보를 셀(cell)에 조직화(organize)한다 | 정보를 복합적인 컬렉션에 조직화한다 |
| 제한적인 양의 데이터에 대한 액세스를 제공한다 | 대량의 데이터에 대한 액세스를 제공한다 |
| 수동으로 데이터를 입력한다 | 엄격하고 일관적인 데이터 입력 |
| 일반적으로 한 번에 한 명의 사용자 | 다중 사용자 |
| 사용자에 의해 통제된다 | 데이터베이스 관리 시스템(database management system)에 의해 통제된다 |

## SQL(Structured Query Language)

SQL은 RDBMS의 데이터를 다루기 위해 사용하는 언어다. 이 책에서 다루는 주제이기도 하다.

- **DML**(Data Manipulation Language, 데이터 조작 언어): DML은 데이터를 추가, 삭제, 갱신, 조회하는 데 사용한다. `INSERT`, `DELETE`, `UPDATE`, `SELECT` 문이 DML에 해당한다.
- **DDL**(Data Definition Language, 데이터 정의 언어): DDL은 테이블 등을 생성, 변경, 제거하는 데 사용한다. `CREATE`, `ALTER`, `DROP`, `TRUNCATE` 문이 DDL에 해당한다. 참고로 SQLite에는 `TRUNCATE` 문이 없다.