

2. 데이터베이스 관리 시스템

데이터베이스 관리 시스템의 등장 배경

- 과거에는 데이터를 관리하기 위해 파일 시스템을 사용함
 - 파일 시스템은 데이터를 파일로 관리할 수 있도록 파일을 생성, 삭제, 수정, 검색 하는 기능을 제공함
 - 운영체제와 함께 설치
 - 응용프로그램별로 필요한 데이터를 별도의 파일로 관리
 - 파일 시스템 환경에서는 응용 프로그래머가 파일의 논리적인 구조뿐 아니라 물리적인 구조까지 정확히 파악해야함
 - 그래야만 필요한 데이터에 직접 접근하여 처리하는 응용 프로그램을 개발할 수 있음



그림 2-1 파일 시스템에서의 데이터 관리

- 파일 시스템은 별도의 구매 비용이 들지 않는 장점이 있음
- 응용 프로그램마다 파일을 따로 유지하는 특징 때문에 다음과 같은 문제가 발생함
 - 같은 내용의 데이터가 여러 파일에 중복 저장됨
 - 파일 시스템에서는 응용 프로그램별로 파일을 유지하므로 같은 데이터가 여러 파일에 저장될 수 있음 (데이터 중복성 문제)
 - 데이터가 중복되면 저장 공간이 낭비될 뿐 아니라 데이터 일관성과 데이터 무결성을 유지하기 어려움
 - 응용 프로그램이 데이터 파일에 종속적임

- 응용 프로그램은 파일에 직접 접근하여 데이터 처리해야 하므로 사용하는 파일의 데이터를 구성하는 방법이나 물리적인 저장 구조에 맞게 작성되어야 함
 - 따라서 사용하는 파일의 구조를 변경하면 응용 프로그램도 함께 변경해야함
 - 데이터 종속성 문제
3. 데이터 파일에 대한 동시 공유, 보안, 회복 기능이 부족(없음)
- 일반 파일 시스템에서는 응용 프로그램 하나가 사용 중인 파일을 다른 응용 프로그램이 접근하여 사용할 수 있는 동시 공유 기능을 제공하지 않음
 - 파일 시스템에서는 사용자에게 보통 파일 단위로 읽기, 수정, 실행 권한을 부여하는 방식으로 데이터 접근을 통제함
 - 이는 레코드나 필드 같은 더 작은 단위에 대한 접근통제와 더 구체적인 권한 부여가 어렵음
 - 데이터가 중복된 모든 파일의 보안을 같은 수준으로 유지하기도 어려움
 - 파일 시스템을 사용하는 도중 장애가 발생하면 데이터를 일관된 상태로 회복하기 어려움(불가능함)
4. 응용 프로그램을 개발하기 쉽지 않다
- 파일 시스템에서는 파일에 접근하여 데이터를 관리하는 모든 작업을 응용 프로그램이 담당해야하기 때문에 사용자 요구에 맞는 응용 프로그램을 개발하는 데 어려움이 많음
-

데이터베이스 관리 시스템의 정의

- 파일 시스템이 지닌 모든 문제의 근본 원인은 데이터 중복성과 데이터 종속성으로 요약할 수 있음
- 이 문제를 해결하고 동시 공유, 보안, 회복등의 복잡한 기능을 제공해주는 대안이 데이터베이스 관리 시스템임
- **데이터 베이스 관리 시스템**
 - 파일 시스템의 데이터 중복과 데이터 종속 문제를 해결하기 위해 제시된 소프트웨어
 - 조직에 필요한 데이터를 데이터베이스에 통합하여 저장하고 이에 대한 관리를 집중적으로 담당
 - 응용 프로그램을 대신하여 데이터베이스에 들어있는 데이터를 삽입, 삭제, 수정, 검색하고, 모든 응용 프로그램이 데이터베이스를 공유할 수 있게 함

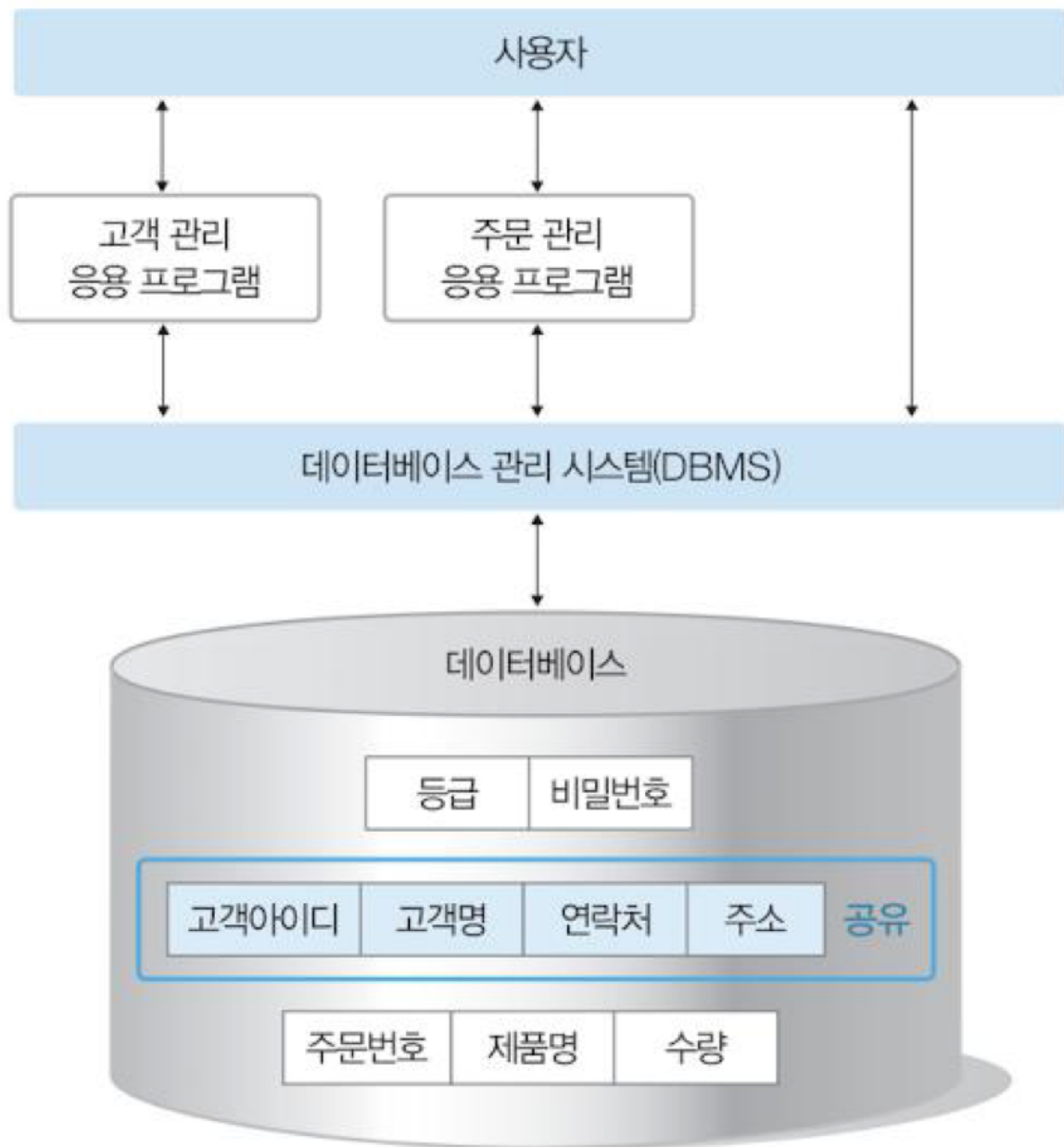


그림 2-5 데이터베이스 관리 시스템에서의 데이터 관리

- 데이터베이스를 생성하고 접근하며 관리하는 일은 데이터베이스 관리 시스템이 모두 담당함
- 사용자는 직접 또는 응용 프로그램을 통해 원하는 데이터가 무엇이며, 어떤 처리를 원하는지만 데이터베이스 관리 시스템에 요청하면 됨
- 즉, 데이터베이스와 관련한 작업은 데이터베이스 관리 시스템이 수행한 후 그 결과만 응용 프로그램을 통해 사용자에게 전달해 주는 것
- 사용자나 응용 프로그램은 데이터베이스의 물리적 저장 구조나 데이터 접근 및 처리 방법 등을 자세히 알 필요가 없음
- 또한 데이터베이스 구조나 접근 방법등이 변경되어도, 사용자가 미리 알거나 응용 프로그램을 변경할 필요가 없어 데이터 독립성이 확보됨

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

- 데이터베이스 관리 시스템의 주요 기능
 - 정의 기능
 - 데이터베이스 관리 시스템은 조직에 필요한 데이터를 저장하기 적합한 데이터베이스 구조를 저으이하거나, 이미 정의된 구조를 수정할 수 있다
 - 조작 기능
 - 데이터베이스 관리 시스템은 데이터베이스에 저장된 데이터에 접근하여 사용할 수 있는 기능을 제공함
 - 즉, 사용자 요구에 따라 데이터를 삽입, 삭제, 수정, 검색하는 연산을 효율적으로 처리
 - 제어 기능
 - 데이터베이스 관리 시스템은 데이터를 여러 사용자가 공유해도 항상 정확하고 안전하게 유지하는 기능을 제공
 - 데이터를 조작하는 연산을 한 수에도 내용이 일관되면서 무결성을 유지하게 하고, 장애가 발생해도 회복이 가능하도록 제어
 - 권한이 있는 사용자에게만 데이터 접근을 허용하여 보안이 유지되도록 제어
 - 여러 사용자가 동시에 접근하여 데이터를 처리할 수 있도록 제어

데이터베이스 관리 시스템의 장단점

데이터베이스 관리 시스템의 장점

1. 데이터 중복을 통제할 수 있다
 - 파일 시스템은 응용 프로그램마다 별도의 파일을 유지하기 때문에 같은 내용의 데이터가 여러 파일에 중복 저장될 수 있다. 이는 저장 공간이 낭비되고 데이터를 저장, 수정하는 비용도 증가한다. 또한, 중복된 데이터를 모두 수정하지 않으면 데이터 불일피로 인해 일관성이 유지되지 못할 수도 있다. 데이터베이스 관리 시스템은 데이터베이스에 데이터를 통합하여 관리하므로 데이터 중복 문제

를 해결할 수 있다. 또한, 효율성을 위해 중복을 허용하는 경우에도 중복을 최소화하도록 통제하므로 데이터 일관성도 유지할 수 있다.

2. 데이터 독립성이 확보된다

- 파일 시스템에는 파일 구조가 바뀌면 응용 프로그램도 함께 수정해야 하는 데이터 종속 문제가 존재한다. 반면, 데이터베이스 관리 시스템은 응용 프로그램을 대신해서 데이터베이스에 접근하고 이를 관리하는 모든 책임을 지기 때문에 데이터베이스 구조가 변경되어도 응용 프로그램이 영향을 받지 않는다. 즉, 응용 프로그램과 데이터베이스 사이에 독립성이 확보된다.

3. 데이터를 동시 공유할 수 있다

- 데이터베이스 관리 시스템은 데이터베이스에 통합된 데이터를 여러 응용 프로그램이 공유하여 같은 데이터에 동시 접근할 수 있도록 지원한다. 이는 데이터베이스 관리 시스템이 동일한 데이터를 각 응용프로그램의 요구에 따라 다양한 구조로 제오해줄 수 있고, 동시 접근을 제어하는 어려운 기술을 보유하고 있어 가능한 일이다. 그리고 동시 공유를 지원하기 때문에 불필요한 데이터 중복을 제한할 수 있다.

4. 데이터 보안이 향상된다

- 파일 시스템은 중복된 모든 파일의 보안을 같은 수준으로 유지하기 어렵고 사용 권한을 파일 단위로 제한하기 때문에 더 구체적이고 다양한 접근 제어를 제공하지 않는다. 반면, 데이터베이스 관리 시스템은 데이터베이스를 이용해 데이터를 중앙 집중식으로 관리하므로 데이터에 대한 효율적인 접근 제어가 가능하다. 권한이 없는 사용자의 접근, 허용되지 않은 데이터와 연상에 대한 요청을 사전에 차단할 수 있어 철저한 보안을 제공한다. 그리고 사용자별로 접근 가능한 데이터베이스 영역을 제한하거나 접근 수준을 차별화 할 수 있다.

5. 데이터 무결성을 유지할 수 있다

- 데이터 부결성은 저장된 데이터 값의 정확성을 의미한다. 예를 들어 학생의 성적이 100점인데 10점으로 기록하거나 고객의 나이를 음수로 저장한 경우 무결성을 위반했다고 할 수 있다. 정확하지 않은 값이나 허용되지 않은 값이 저장되어 있으면 잘못된 처리 결과를 유도하여 올바른 의사 결정이 어려워진다. 데이터베이스 관리 시스템은 데이터에 대한 관리를 집중적으로 수행하면서 데이터에 대한 연산이 수행될 때 마다 유효성을 검사하여 데이터 무결성을 유지할 수 있게 해준다.

6. 표준화할 수 있다

- 데이터베이스에 대한 모든 접근이 데이터베이스 관리 시스템을 통해 이루어지기 때문에 데이터에 접근하는 방법, 데이터 형식과 구조 등을 표준화하기 쉽다. 모든 응용 프로그램은 데이터베이스 관리 시스템이 미리 정한 표준화된 방식을 통해 데이터베이스에 접근한다.

7. 장애 발생 시 회복이 가능하다

- 저장된 데이터에 접근하여 처리하는 과정에서는 다양한 장애가 발생할 수 있다. 데이터베이스 관리 시스템은 장애가 발생해도 데이터 일관성과 무결성을 유지하면서 데이터를 장애가 발생하기 이전 상태로 복구하는 회복 기능을 지원한다.

8. 응용 프로그램 개발 비용이 줄어든다

- 데이터에 대한 모든 관리를 응용 프로그램 대신 데이터베이스 관리 시스템이 담당하기 때문에 파일 시스템을 사용할때보다 응용 프로그램 개발 비용이 적게 든다. 그리고, 데이터베이스의 구조가 변경되어도 응용 프로그램은 변경할 필요가 없어 유지 보수 비용이 줄어든다.

데이터베이스 관리 시스템의 단점

1. 비용이 많이 든다
 - 파일 시스템은 운영체제와 함께 설치되므로 따로 구매 비용이 들지 않지만, 데이터베이스 관리 시스템은 따로 설치해야 하므로 구매 비용이 많이 든다.
2. 백업과 회복 방법이 복잡하다
 - 데이터베이스는 데이터 양이 많아 구조가 복잡하고, 여러 사용자의 동시 공유를 지원하므로 장애가 발생했을 때 원인과 상태를 정확히 파악하기 힘들다. 그래서 장애 발생 전에 데이터를 미리 백업해 놓고 장애 발생 후 데이터를 원래의 일관된 상태로 회복하는 방법이 복잡할 수 밖에 없다.
3. 중앙 집중 관리로 인한 취약점이 존재한다
 - 모든 데이터가 데이터베이스에 통합되어 있고 이에 대한 관리 책임이 데이터베이스 관리 시스템에 집중되어, 데이터베이스나 데이터베이스 관리 시스템에 장애가 발생하면 전체 시스템의 업무 처리가 중단된다.

장점	단점
<input type="checkbox"/> 데이터 중복을 통제할 수 있다	<input type="checkbox"/> 비용이 많이 든다
<input type="checkbox"/> 데이터 독립성이 확보된다	<input type="checkbox"/> 백업과 회복 방법이 복잡하다
<input type="checkbox"/> 데이터를 동시 공유할 수 있다	<input type="checkbox"/> 중앙 집중 관리로 인한 취약점이 존재한다
<input type="checkbox"/> 데이터 보안이 향상된다	
<input type="checkbox"/> 데이터 무결성을 유지할 수 있다	
<input type="checkbox"/> 표준화할 수 있다	
<input type="checkbox"/> 장애 발생 시 회복이 가능하다	
<input type="checkbox"/> 응용 프로그램 개발 비용이 줄어든다	

그림 2-7 데이터베이스 관리 시스템의 장점과 단점

데이터베이스 관리 시스템의 발전 과정

1. 1세대 데이터베이스 관리 시스템 : 네트워크, 계층 DBMS



그림 2-8 1세대 DBMS 구조의 예

- 네트워크 DBMS는 데이터베이스를 노드와 간선을 이용한 그래프 형태로 구성하는 네트워크 데이터 모델을 사용한다
- 네트워크 DBMS는 간선을 이용해 데이터 간의 관계를 표현하기 때문에 데이터베이스의 구조가 복잡하고 변경하기 어렵다는 단점이 있다
- 계층 DBMS는 데이터베이스를 트리 형태로 구성하는 계층 데이터 모델을 사용한다
- 계층 DBMS는 네트워크 DBMS보다 구조가 단순하지만, 복잡한 현실 세계의 모습을 부모 자식 관계가 명확한 트리 형태만으로 표현하기 힘들고 구조 변경이 어렵다는 단점이 여전히 존재한다

2. 2세대 데이터베이스 관리 시스템 : 관계 DBMS

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

- 관계 DBMS는 데이터베이스를 테이블 형태로 구성하는 관계 데이터 모델을 사용한다
- 관계 데이터 모델은 데이터베이스를 단순하고 이해하기 쉬운 구조로 구성한다는 장점이 있다
 - 따라서 관계 데이터 모델을 사용하는 관계 DBMS가 많이 개발되었으며 연구도 많이 이루어졌다
 - 현재도 널리 사용되는 DBMS

3. 3세대 데이터베이스 관리 시스템 : 객체지향, 객체관계 DBMS

- **객체지향 DBMS**는 객체지향 프로그래밍에서 도입한 객체라는 개념을 이용해 데이터베이스를 구성하는 객체지향 데이터 모델을 사용한다
- 객체지향 DBMS는 더 다양하고 복잡한 응용 분야의 데이터를 관리하기 위해 제안되었다
- 이 시스템은 새로운 유형의 데이터를 저장하고, 데이터에 대한 복잡한 분석 및 처리를 지원한다

- **객체관계 DBMS**는 관계 데이터 모델에 객체지향 개념을 도입한 객체관계 데이터 모델을 사용한다
 - 객체지향 DBMS + 관계 DBMS
 - 일반적으로 단순하고 이해하기 쉬운 구조인 관계 DBMS가 많이 사용된다
 - 하지만 객체관계 DBMS의 사용도 늘고 있어 2세대와 3세대 DBMS가 공존한다고 볼 수 있다
-

4. 4세대 이후 데이터베이스 관리 시스템 : NoSQL, NewSQL DBMS

- 3세대 DBMS가 등장한 후에도 대표적인 DBMS는 관계 DBMS라는 인식이 일반적이었다
 - 대부분의 기업에서는 관계 DBMS의 여러 장점을 그대로 유지하고자 했고, 새롭게 등장한 DBMS로 전환하는데 큰 매력을 느끼지 못했던 것
- 하지만 SNS의 사용이 많아지며 사진, 동영상등 비정형 데이터가 대량으로 생산되고, 클라우드 컴퓨팅, 빅데이터 개념이 등장하며 관계DBMS에 대한 확신이 흔들리게 되었다
- 관계 DBMS는 대량의 비정형 데이터를 처리하는데 비효율적이므로 관계 DBMS는 그대로 유지한 채 CPU나 메모리 같은 하드웨어 장치의 성능을 향상하는 것만으로는 문제를 해결할 수 없었다
- 이런 흐름은 NoSQL DBMS의 등장을 이끌었다
- NoSQL DBMS는 관계 DBMS가 강점으로 제시하던 안정성과 일관성 유지를 위한 복잡한 기능을 포기하고, 데이터 구조를 미리 정해두지 않기 때문에 비정형 데이터를 저장하고 처리하는데 적합하다
- 또한 확장성이 뛰어나 여러 대의 서버 컴퓨터에 데이터를 분산하여 저장하고 처리하는 환경에서 주로 사용한다
- 하지만 기존의 관계 DBMS의 강점을 포기하지 않고 비정형 데이터를 처리하려면 NoSQL을 추가로 도입해야하는 부담이 발생했다
- 따라서 안정성과 일관성을 유지하면서도 SQL을 이용해 다양하고 복잡한 데이터 처리를 편리하게 요청할 수 있는 NewSQL DBMS가 등장했다

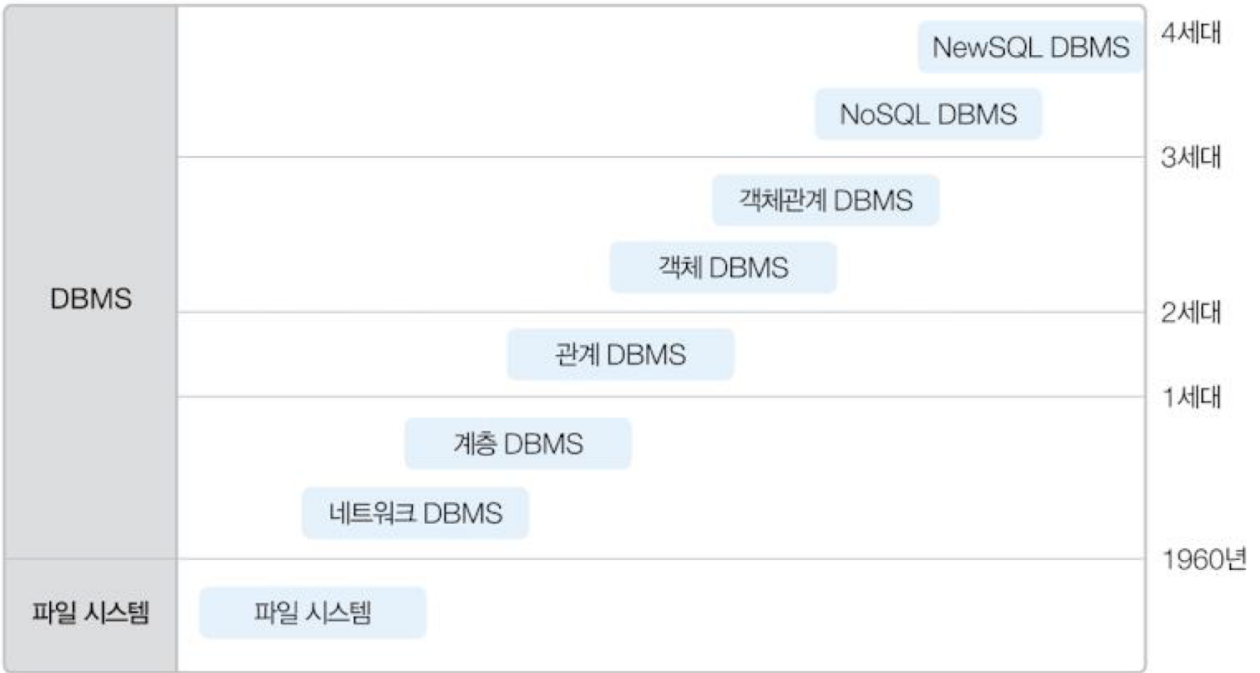


그림 2-10 DBMS의 발전 과정