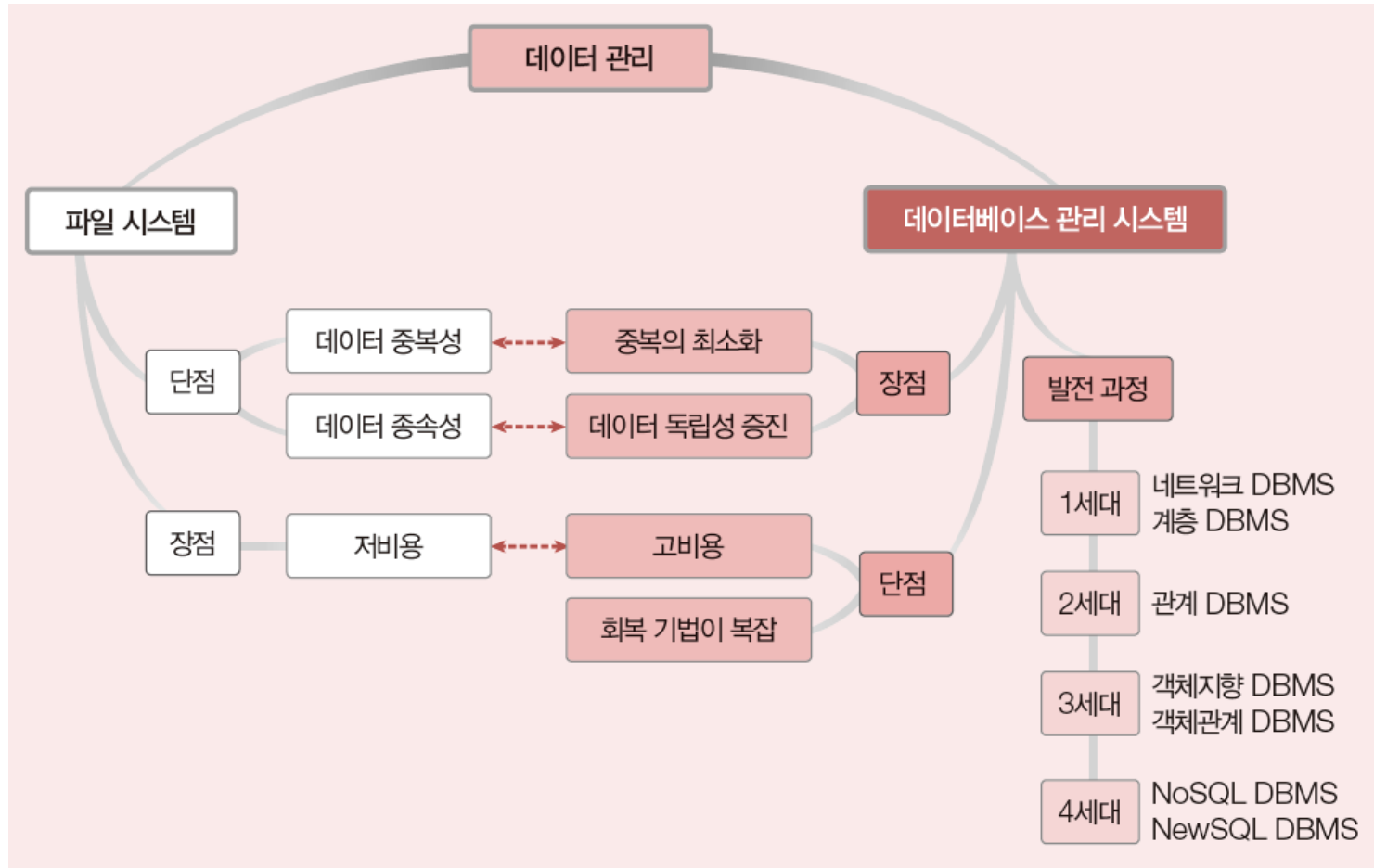
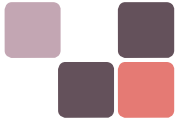


2장. 데이터베이스 관리 시스템

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



- ❖ 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 프로그램마다 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 프로그램 개발이 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

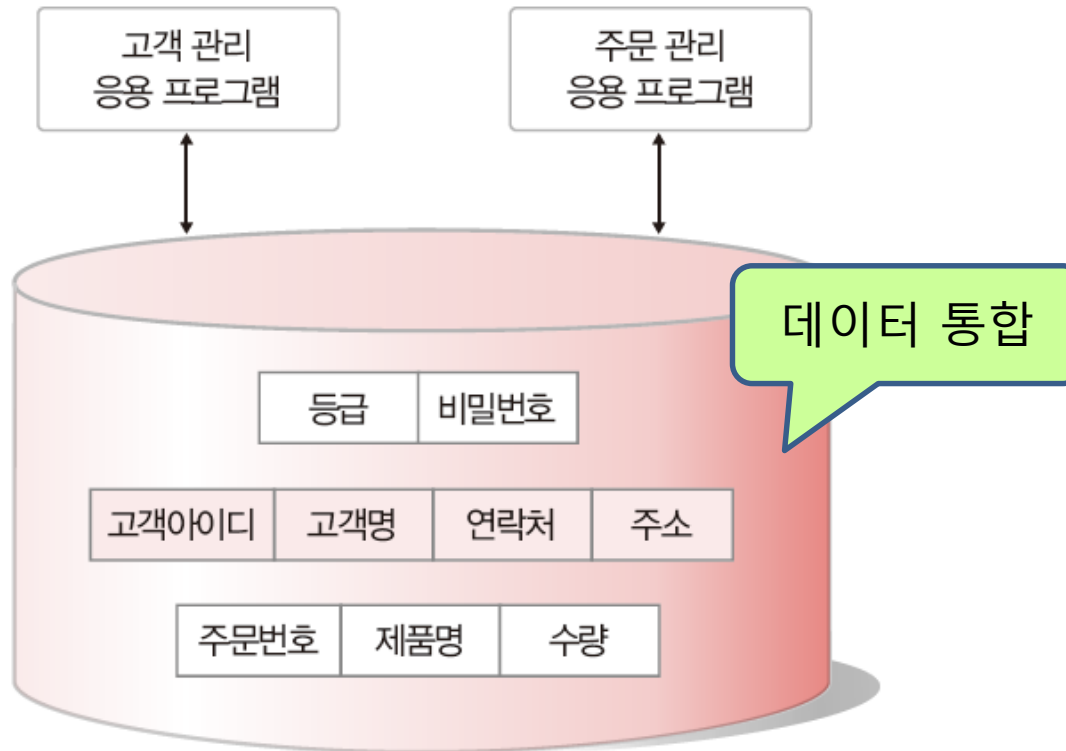


그림 2-2 파일 시스템의 데이터 중복성 문제를 해결하는 1차 방안

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

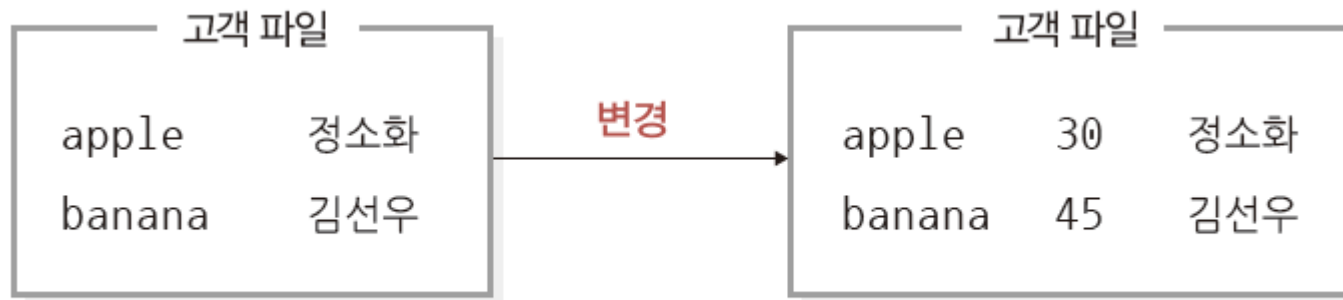
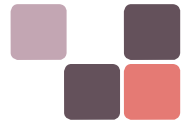


그림 2-3 파일 구조 변경 예



❖ 데이터베이스 관리 시스템

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

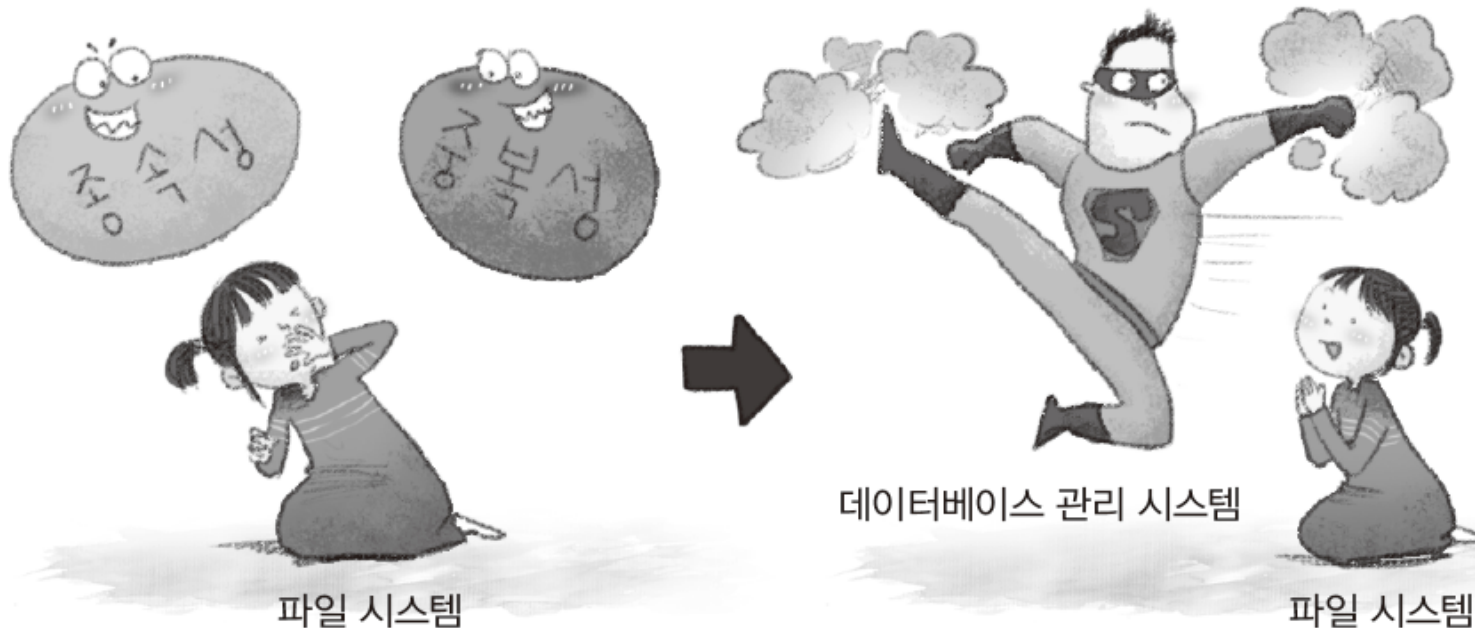
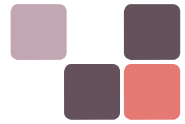


그림 2-4 파일 시스템과 데이터베이스 관리 시스템

02 데이터베이스 관리 시스템의 정의



❖ 데이터베이스 관리 시스템에서의 데이터 관리

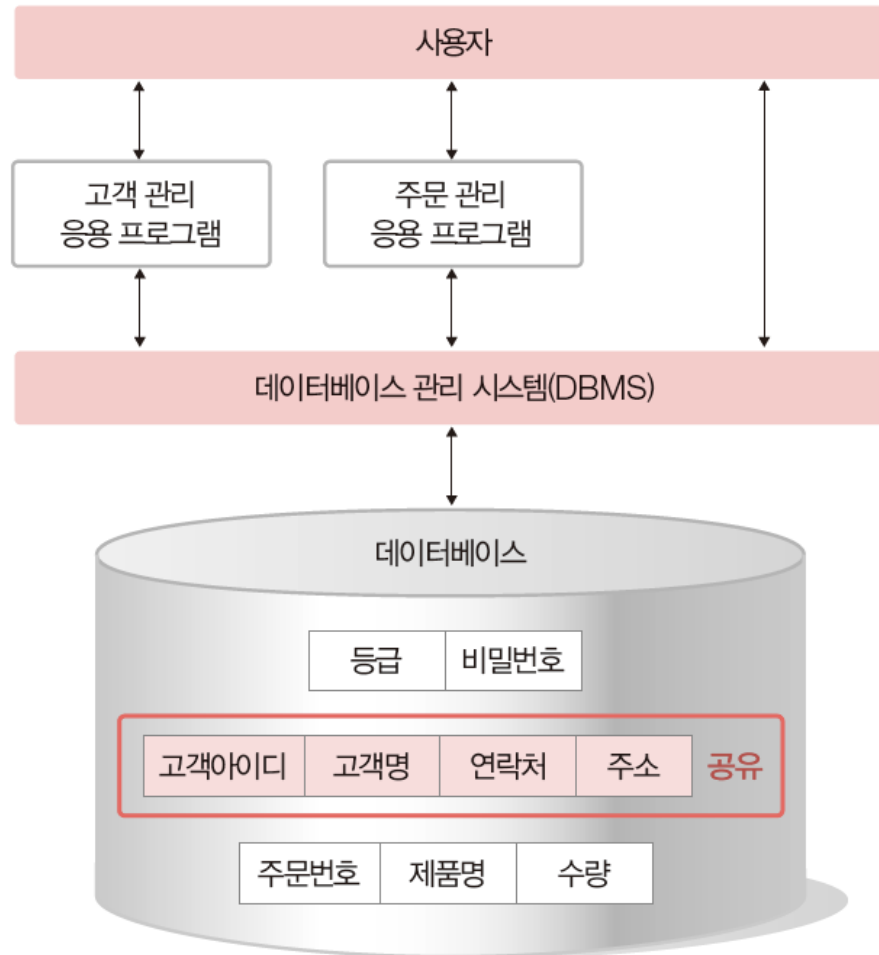
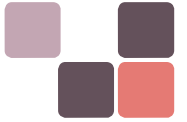


그림 2-5 데이터베이스 관리 시스템에서의 데이터 관리

02 데이터베이스 관리 시스템의 정의

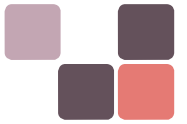


❖ 데이터베이스 관리 시스템의 주요 기능

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 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

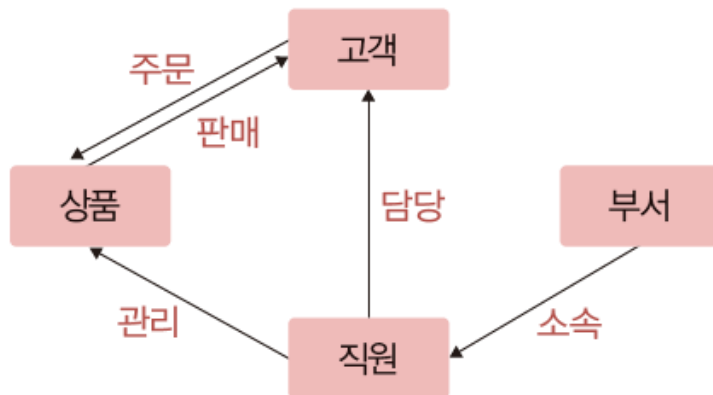
- ☐ 비용이 많이 든다
- ☐ 백업과 회복 방법이 복잡하다
- ☐ 중앙 집중 관리로 인한 취약점이 존재한다

그림 2-7 데이터베이스 관리 시스템의 장점과 단점

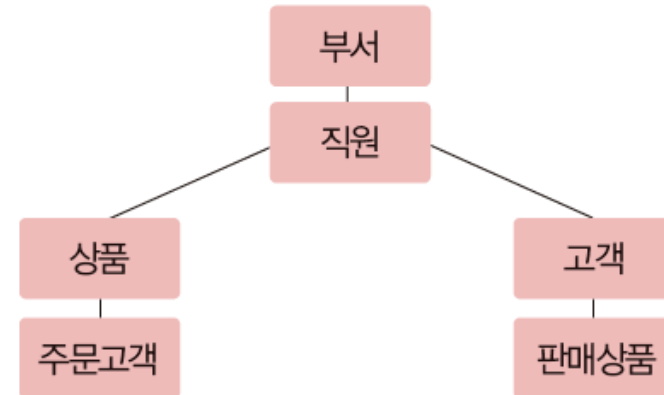
04 데이터베이스 관리 시스템의 발전 과정

❖ 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

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 : 객체 DBMS + 관계 DBMS

04 데이터베이스 관리 시스템의 발전 과정

❖ 4세대 : NoSQL • NewSQL DBMS

- NoSQL DBMS : 비정형 데이터를 처리하는데 적합하고 확장성이 뛰어남
 - 안정성과 일관성 유지를 위한 복잡한 기능을 포기
 - 데이터 구조를 미리 정해두지 않는 유연성
 - 확장성이 뛰어나 여러 대의 서버 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
 - 예) 구글 스패너(Spanner), 볼트DB(VoltDB), 누오DB(NuoDB)

04 데이터베이스 관리 시스템의 발전 과정



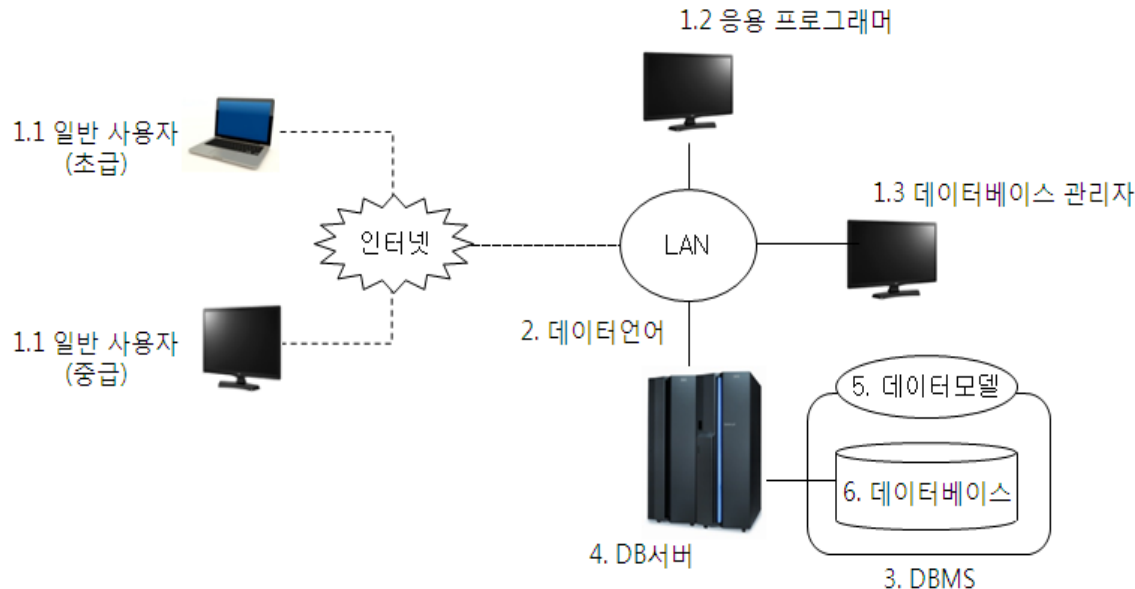
그림 2-10 DBMS의 발전 과정

1. 데이터베이스 시스템의 구성요소

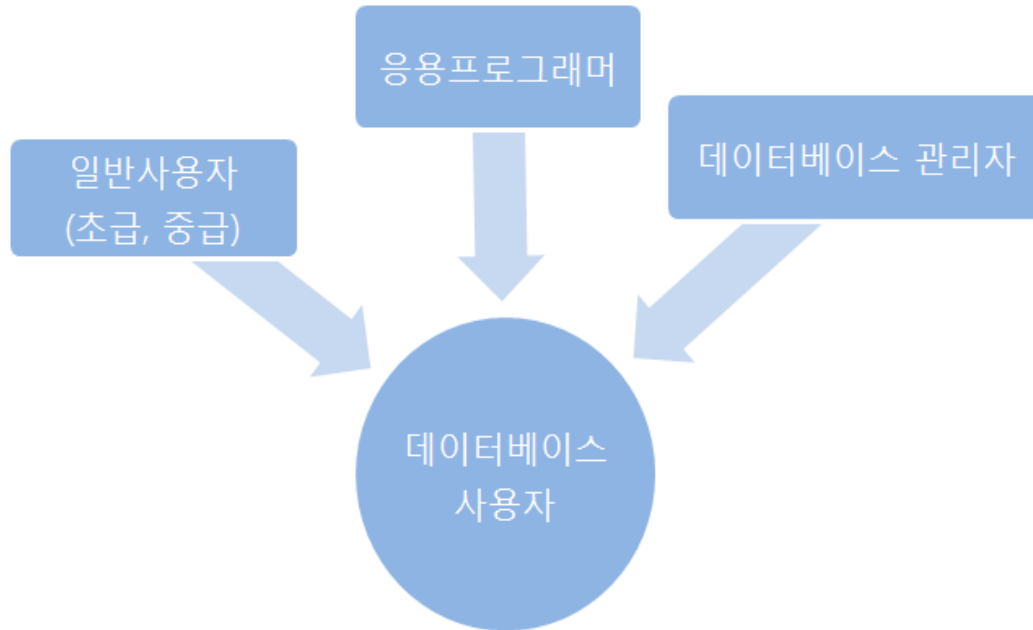
1.1 데이터베이스 시스템의 개념

- 데이터베이스(DB: DataBase)
 - : 저장소
- 데이터베이스 관리 시스템(DBMS: DataBase Management System)
 - : 데이터베이스를 관리하는 소프트웨어
- 데이터베이스 시스템(DBS: DataBase System)
 - : 데이터베이스와 데이터베이스 관리 시스템들을 모두 포함하는 개념

● 데이터베이스 시스템의 구성 요소



데이터베이스 사용자의 분류



1.2 데이터베이스 사용자

- 일반 사용자 = 최종 사용자(end user)
 - 컴퓨터나 데이터베이스에 대한 전문 지식 없이 데이터베이스를 접근하는 사용자 유형
 - 초급 사용자: 데이터베이스나 DBMS의 존재를 알지 못하는 관련 지식이 없는 사용자
 - 중급 사용자: 데이터베이스 기본 지식을 갖추고 필요할 경우, 데이터 언어를 사용하여 데이터에 대한 처리를 직접 DBMS에 요청
- 응용 프로그래머
 - 데이터베이스 전문 지식을 가지고 응용 프로그램(application)을 개발할 목적으로 데이터베이스를 접근하는 사용자 유형
 - 프로그램 전문 개발자로 C, Java, JSP, PHP와 같은 프로그래밍 언어를 구사할 수 있고 작성한 응용 프로그램 안에 데이터 언어나 DBMS 실행 명령어 작성 능력을 갖춘 사용자
- 데이터베이스 관리자(DBA: DataBase Administrator)
 - 데이터베이스를 구축하고 데이터베이스 시스템을 자체적으로 운영·통제하는 특별한 소수의 사용자 유형
 - DBMS의 슈퍼 사용자(super user)
 - 데이터베이스 시스템의 관리를 총괄(데이터베이스의 설계, 구축, 서비스, 운영 등을 위해 DBMS의 다양한 기능들을 활용함으로써 데이터베이스 시스템을 제어)

데이터베이스 관리자

- 데이터베이스에 대한 모든 권한과 최종 책임을 갖는 전문가
- DBA의 주요 역할
 - 데이터베이스 시스템의 구성 요소 선정
 - 데이터베이스의 구조 정의
 - 물리적 저장 구조와 접근 방법 결정
 - 무결성 유지를 위한 제약 조건 정의
 - 보안 및 접근 권한 정책 결정
 - 백업 및 회복 기법 정의
 - 시스템 데이터베이스 관리
 - 시스템 성능 감시 및 성능 분석
 - 데이터베이스의 재구성
 - 데이터베이스 관련 의견 조정과 분쟁 해결

데이터베이스 사용자별 요구 능력

(○: 높음, △: 중간, ×: 없음)

구분	일반 사용자		응용 프로그래머	데이터베이스 관리자
	초급	중급		
SQL 활용 능력	×	△	○	○
프로그래밍 능력	×	×	○	△
모델링 능력	×	×	△	○
DBMS 활용 능력	×	△	△	○

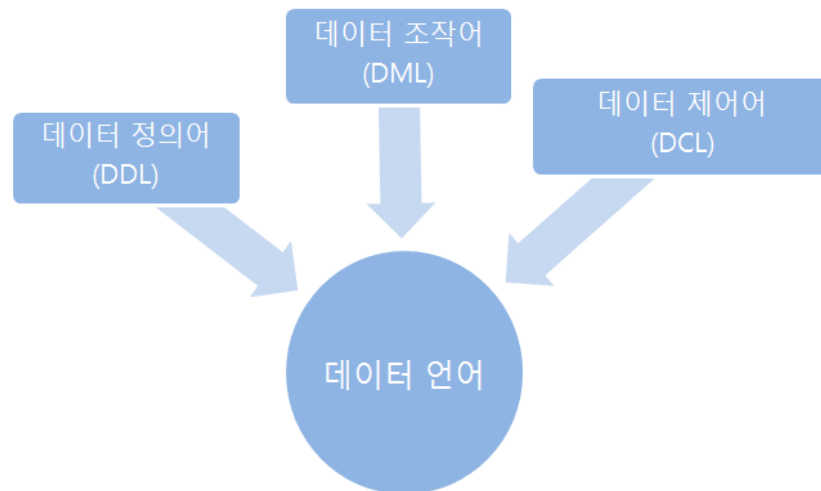
1.3 데이터 언어

- 데이터 언어(data language)

- 데이터베이스 사용자와 응용 프로그램은 모두 DBMS를 통해서만 데이터베이스에 접근할 수 있음
- DBMS에 요청 내용을 전달하기 위한 도구
- 보통 데이터 언어는 표준 데이터베이스 언어인 SQL(Structured Query Language)을 의미함

- 데이터 언어의 분류

- 사용 목적에 3가지 명령어 그룹으로 분류



데이터 언어의 분류

- 데이터 정의어(DDL: Data Definition Language)

- 새로운 데이터베이스 구조를 정의하고 기존 데이터베이스 구조를 변경하는 명령어 집합
- 데이터베이스 구조를 표현하는 데이터베이스 스키마를 명세하기 위해 사용

• DDL 명령어의 예 : CREATE, ALTER, DROP

- 데이터 조작어(DML: Data Manipulation Language)

- 데이터베이스 안의 데이터를 실제 조작하는 명령어 집합
- DBMS에게 데이터의 입력, 수정, 삭제 및 검색을 요청하기 위해 사용

• DML 명령어의 예 : INSERT, UPDATE, DELETE, SELECT

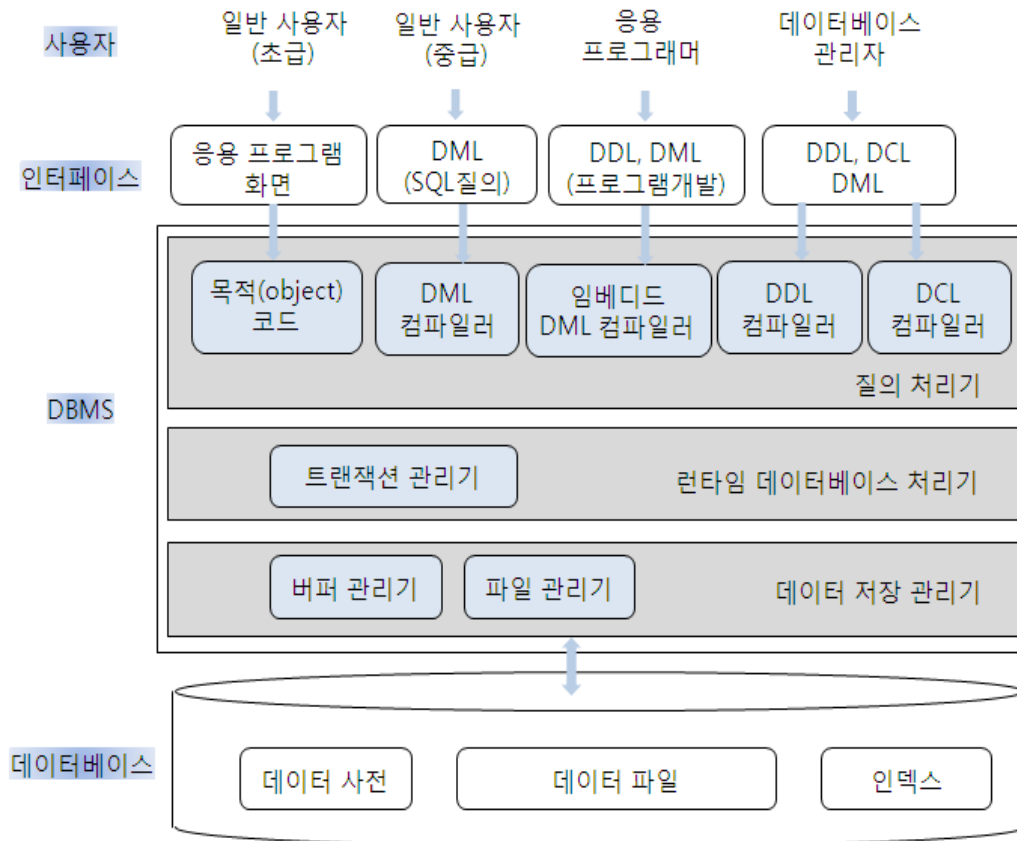
- 데이터 제어어(DCL: Data Control Language)

- 데이터베이스를 제어하고 통제하기 위해 사용하는 명령어 집합
- 데이터베이스가 안전하게 오류 없이 동작하고 성능을 유지하도록 각종 제약이나 옵션을 설정함으로써 DBMS가 데이터베이스를 올바르게 관리하도록 함

• DCL 명령어의 예 : GRANT, REVOKE, CREATE USER, COMMIT, ROLLBACK

1.4 데이터베이스 관리 시스템

- 데이터베이스 관리 시스템(DBMS: Database Management System)
 - 데이터베이스를 효율적으로 관리하고 데이터베이스에 대한 데이터 요청을 처리하는 소프트웨어 시스템
- DBMS의 구성 요소



DBMS의 필수 기능

1) 정의 기능

- DBMS는 필요한 모든 데이터를 저장하는 통합 데이터베이스 구조를 생성하거나 이미 생성된 구조를 삭제 또는 변경할 수 있도록 함

2) 조작 기능

- DBMS는 데이터베이스 안에 저장된 데이터에 접근하여 원하는 데이터 조작을 할 수 있도록 함
- 사용자의 다양한 입력, 수정, 삭제 및 검색 요청을 효율적으로 처리

3) 제어 기능

- DBMS는 여러 사용자가 동시에 다양한 목적으로 접근하더라도 항상 데이터를 정확하고 안전하게 유지하도록 통제함
- 사용자별 보안과 권한을 설정하고 데이터 조작 과정 중에 높은 동시성과 무결성을 유지하면서도 백업을 통해 장애 발생 시 필요한 회복 조치가 가능하도록 제어

사용자 유형별 활용 SQL 및 DBMS 기능

- 사용자 유형에 따라서 사용하는 SQL 명령어와 이를 해석하고 처리하는 DBMS의 필수 기능이 밀접하게 관련됨

사용자 유형		SQL 명령어	DBMS 기능
일반 사용자	초급	-	조작 기능
	중급	DML	조작 기능
응용 프로그래머		DDL, DML	정의·조작 기능
데이터베이스 관리자		DDL, DML, DCL	정의·조작·제어 기능

1.5 데이터베이스 서버

- 데이터베이스 서버
 - 데이터베이스가 구동되는 서버(server)의 역할을 하는 컴퓨터
 - 클라이언트/서버 컴퓨팅 환경
 - 사용자는 클라이언트 컴퓨터를 통해 접근
 - 데이터베이스 처리는 시스템 부하가 매우 크므로 보통 데이터베이스 시스템을 독립된 컴퓨터 안에 서버 형태로 운영
 - 데이터베이스 서버 안에는 데이터베이스가 물리적으로 저장되며 DBMS가 설치되어 다양한 요청이 처리됨
- 구축하고자 하는 정보시스템의 규모와 용도에 맞게 적절한 사양의 컴퓨터를 선택
 - 전통적으로 데이터베이스 컴퓨터는 고성능의 사양을 요구
 - 점차 컴퓨터의 처리 능력과 성능이 크게 발전됨에 따라 현재는 데이터베이스 전용 컴퓨터가 없어도 일반 컴퓨터를 데이터베이스 서버로 활용하는 것이 가능한 환경이 됨

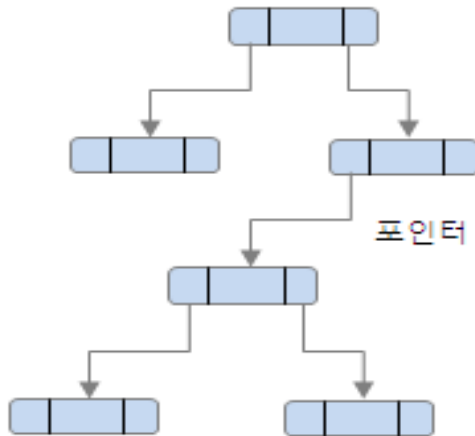
1.6 데이터 모델

- 데이터 모델(data model)
 - 데이터베이스 종류에 따라서 다양한 저장 구조를 갖게 되는데 이러한 데이터베이스 구조를 명세하기 위한 개념 틀(concept framework)
- 데이터 모델의 종류
 - 계층형 데이터 모델
 - 네트워크형 데이터 모델
 - 관계형 데이터 모델
 - 객체 지향형 데이터 모델
 - 객체 관계형 데이터 모델

데이터 모델의 종류

1) 계층형 데이터 모델(HDM: Hierarchical Data Model)

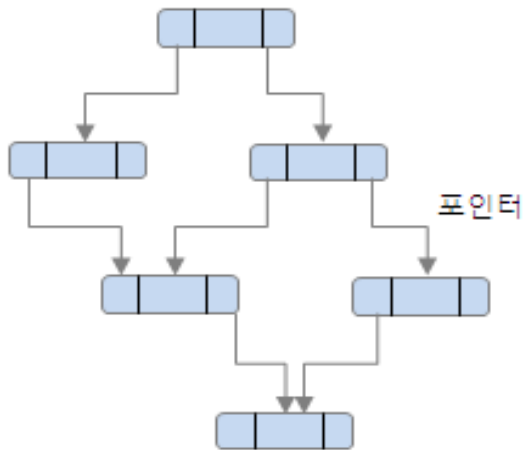
- 데이터 사이의 연관 관계를 물리적 위치 정보인 포인터로 표현하는 저장 방식
- 트리 형태의 계층 구조
 - 각 데이터들이 하나의 상위 부모와 여러 개의 하위 자식들을 갖는 구조
- 장점
 - 검색 속도가 빠르기 때문에 일대다(1:n) 관계를 갖는 대용량 데이터베이스 처리 유리



데이터 모델의 종류

2) 네트워크형 데이터 모델(NDM: Network Data Model)

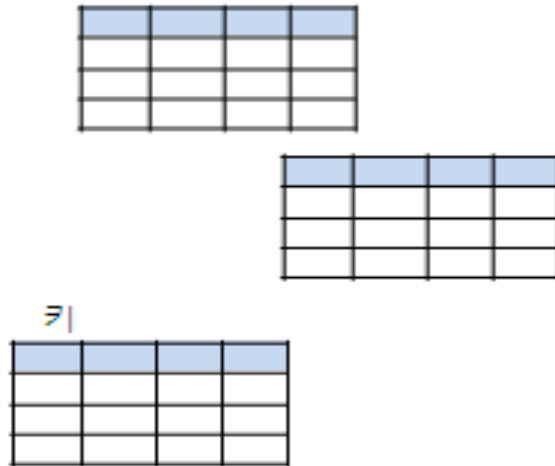
- 데이터 사이의 연관 관계를 포인터로 표현하는 저장 방식
- 계층형 모델의 단점을 보완하여 트리 구조를 네트워크 형태로 변환한 모델
- 장점
 - 검색 속도가 빠르고 다대다(m:n) 표현이 용이함



데이터 모델의 종류

3) 관계형 데이터 모델(RDM: Relational Data Model)

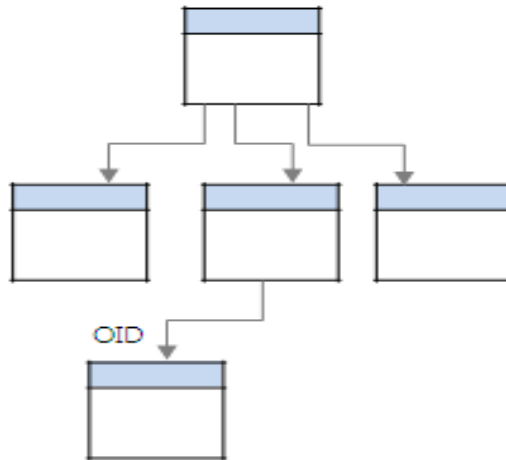
- 테이블 형태의 저장 구조를 가지며 데이터 사이의 연관 관계를 테이블의 키(key) 열(column)을 통해 표현하는 저장 방식
- 장점
 - 연관된 데이터 사이에 기본키와 외래키를 통해 논리적 관련성을 표현함으로써 유연하고 이해하기 쉬움



데이터 모델의 종류

4) 객체 지향형 데이터 모델(OODM: Object Oriented Data Model)

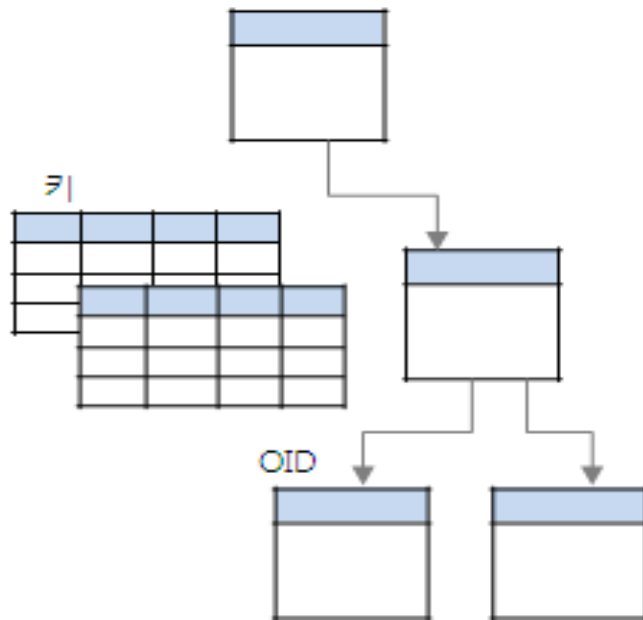
- 객체 단위의 저장 구조를 가지며 연관된 데이터 사이의 관계를 객체 식별자(OID: Object Identifier)로 표현하는 저장 방식
- 클래스, 상속, 자료추상화 등의 객체 지향 개념과 데이터베이스 개념을 통합하여 실세계와 유사하게 모형화하는 데이터 모델
 - 구조적 측면에서는 계층형 또는 네트워크형 데이터 모델과 비슷함
 - 객체 지향 언어의 상속, 캡슐화 등의 개념을 저장 구조에 도입



데이터 모델의 종류

5) 객체 관계형 데이터 모델(ORDM: Object Relational Data Model)

- 관계형 데이터 모델을 기본 개념으로 하고 객체 데이터 모델의 객체 특성을 일부 반영한 절충형 모델
 - 객체 지향 데이터 모델과 관계형 데이터 모델의 장점을 취해 하나의 시스템으로 통합
 - 범용적인 관계형 데이터 모델을 기반으로 하고 필요할 경우, 특정한 분야를 위해 객체 개념을 추가로 지원하는 방식



데이터 모델과 DBMS의 관계

- 데이터베이스 또는 DBMS를 분류하는 가장 중요한 기준은 데이터 모델
- 데이터 모델에 따른 DBMS의 역사

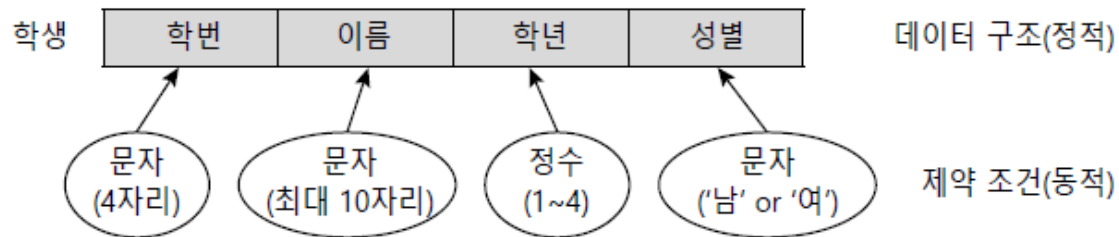
세대	데이터모델	데이터 베이스	DBMS	DBMS의 예
1세대	계층형(Hierarchical)	HDB	HDBMS	IMS
	네트워크형(Network)	NDB	NDBMS	IDS, 코다실(CODASYL)
2세대	관계형(Relational)	RDB	RDBMS	System R, 잉그레스(Ingres), 포스트그레스(POSTGRES), 사이베이스(Sybase), 액세스(Access), 인포믹스(Informix), DB2, 오라클(Oracle), MS SQL 서버, MySQL
3세대	객체 지향형(Object)	OODB	OODBMS	O2, 온토스(ONTOS), 젬스톤(GemStone), 오브젝트스토어(ObjectStore)
	객체 관계형(Object-Relational)	ORDB	ORDBMS	인포믹스 유니버설 서버(Informix Universal Server), 유니SQL(UniSQL), 오라클(Oracle)

2. 3단계 데이터베이스

2.1 스키마(schema)

- 데이터베이스 안에 저장되는 데이터 구조와 제약 조건 등을 정의한 것
- 데이터베이스의 정적인 구성뿐만 아니라 동적인 유지 조건까지를 포함
- 동일한 데이터베이스라도 접근 관점에 따라 스키마는 다를 수 있음

● 스키마의 예

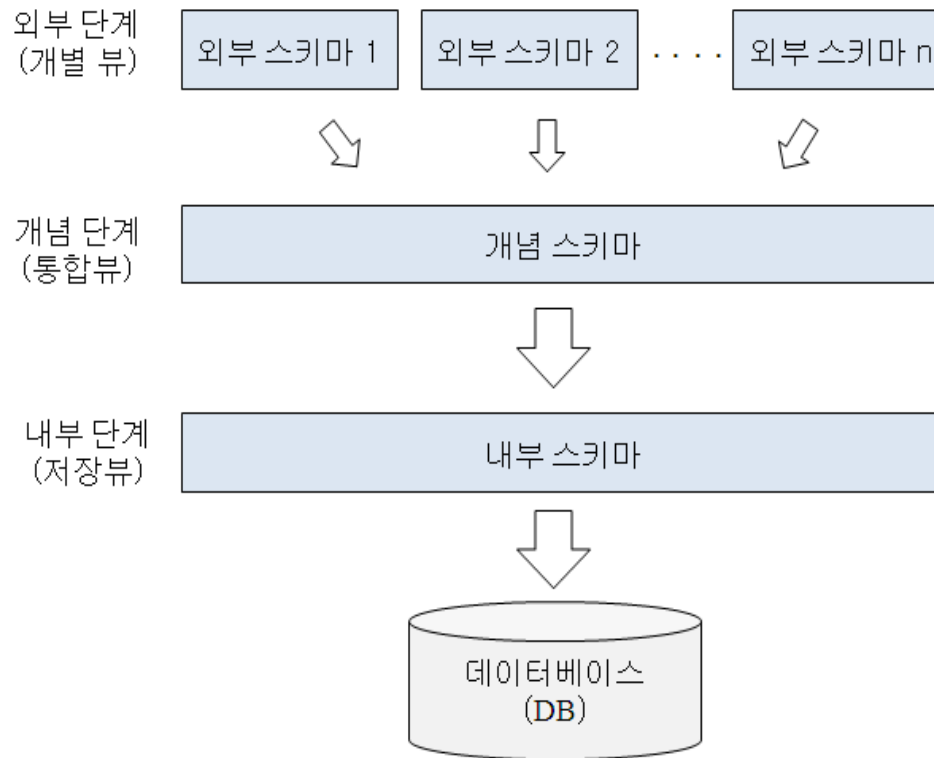


- 주로 데이터 구조에 관심을 가지므로 대부분의 경우, 스키마의 세부 제약 조건보다는 릴레이션 구조만을 간략히 명세함
- 다음과 같이 테이블 이름과 열 이름으로만 간단히 스키마를 명세

학생(학번, 이름, 학년, 성별)

2.2 3단계 데이터베이스의 구조 (3-level database architecture)

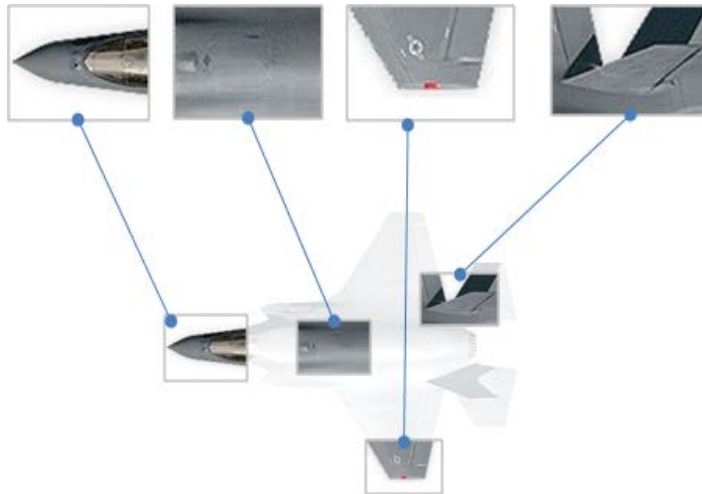
- ANSI에서 제시한 방법(ANSI/SPARC 3단계 아키텍처, 1978년)으로 데이터베이스를 보는 관점을 3개의 단계로 분리



스키마의 종류

1) 외부 스키마(external schema)

- 사용자가 외부에서 바라보는 관점에서의 개인적 데이터베이스 구조를 정의한 것
- 일반 사용자나 응용 프로그래머 차원에서 접근하는 일부 데이터베이스의 논리적 부분을 표현
- 데이터베이스 전체가 아닌 데이터베이스 일부만을 대상으로 한정하여 명세한 구조이므로 서브 스키마(sub schema)라고도 부름
- 각 사용자별로 외부에서 바라보는 관점은 다양하므로 여러 외부 스키마가 존재
 - 예) '강의정보' 서브 스키마(교수), '수강정보' 서브 스키마(학생)



스키마의 종류

2) 개념 스키마(conceptual schema)

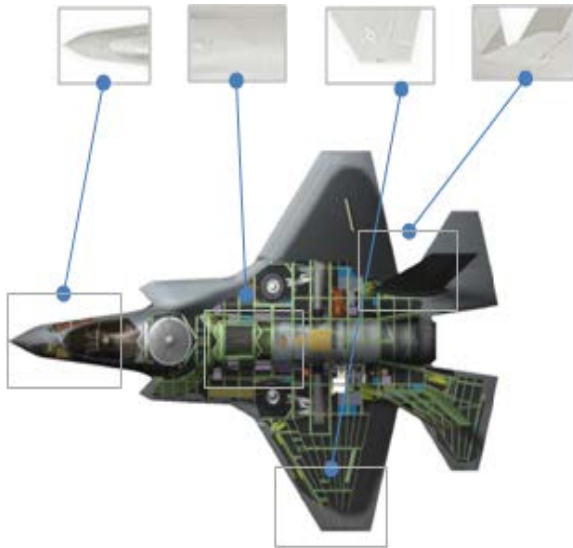
- 모든 사용자들의 관점을 통합하여 전체 조직 관점에서 데이터베이스 구조를 정의한 것
- 데이터베이스 관리 차원에서 접근하는 통합된 데이터베이스의 논리적 부분을 표현
- 조직이나 기관의 데이터베이스 전체를 명세한 구조로서 간단히 스키마(schema)라고 부름
- 데이터베이스에는 하나의 개념 스키마만 존재
- 보통 스키마라고 하면 개념 스키마를 의미함
 - 예) '대학정보' 개념 스키마(대학)



스키마의 종류

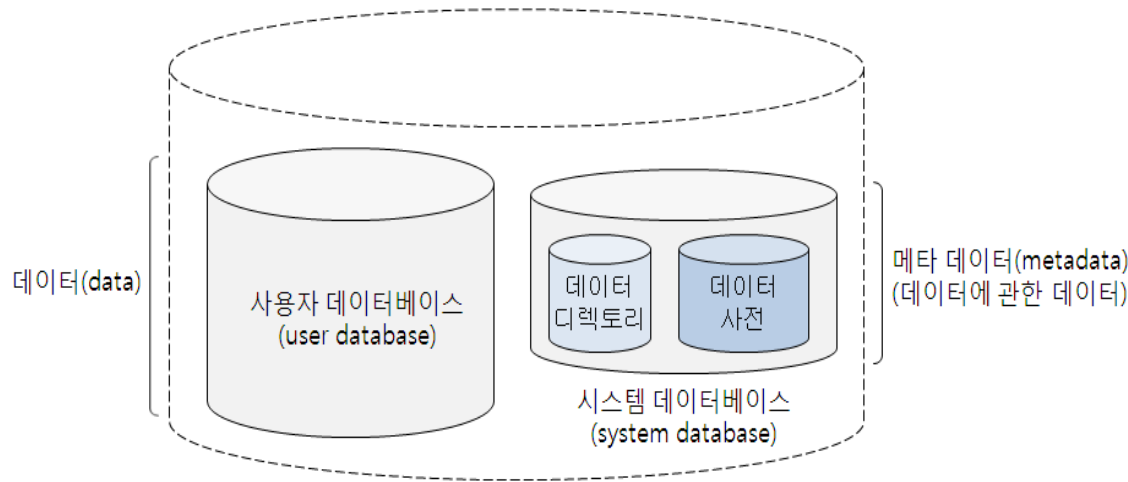
3) 내부 스키마(internal schema)

- 저장 장치의 관점에서 전체 데이터베이스의 내부 구조를 정의
- 개념 스키마에 대한 시스템 내부의 저장 방식을 표현
- 내부 레코드의 형식이나 배치 방법, 인덱스 등에 대한 명세를 포함
- 실제 장치의 물리적 저장 방식이나 구조를 명세한 것은 아니며 그보다는 추상화된 상위 표현
- 내부 스키마도 데이터베이스 당 하나만 존재
 - 예) '대학정보' 내부 스키마(대학)



2.3 데이터 사전

- DBMS는 스키마와 스키마 사상 정보를 데이터 사전이라는 별도의 저장소에 관리
- 데이터 사전(data dictionary)
 - 다양한 데이터베이스 객체(테이블과 열, 뷰, 인덱스, 사용자 권한 등)에 관한 모든 데이터를 포함
 - DBMS가 스스로 생성하고 유지
 - 사용자뿐만 아니라 시스템 관리를 위해서도 필요한 저장 공간으로 그 자체가 하나의 데이터베이스



데이터 사전

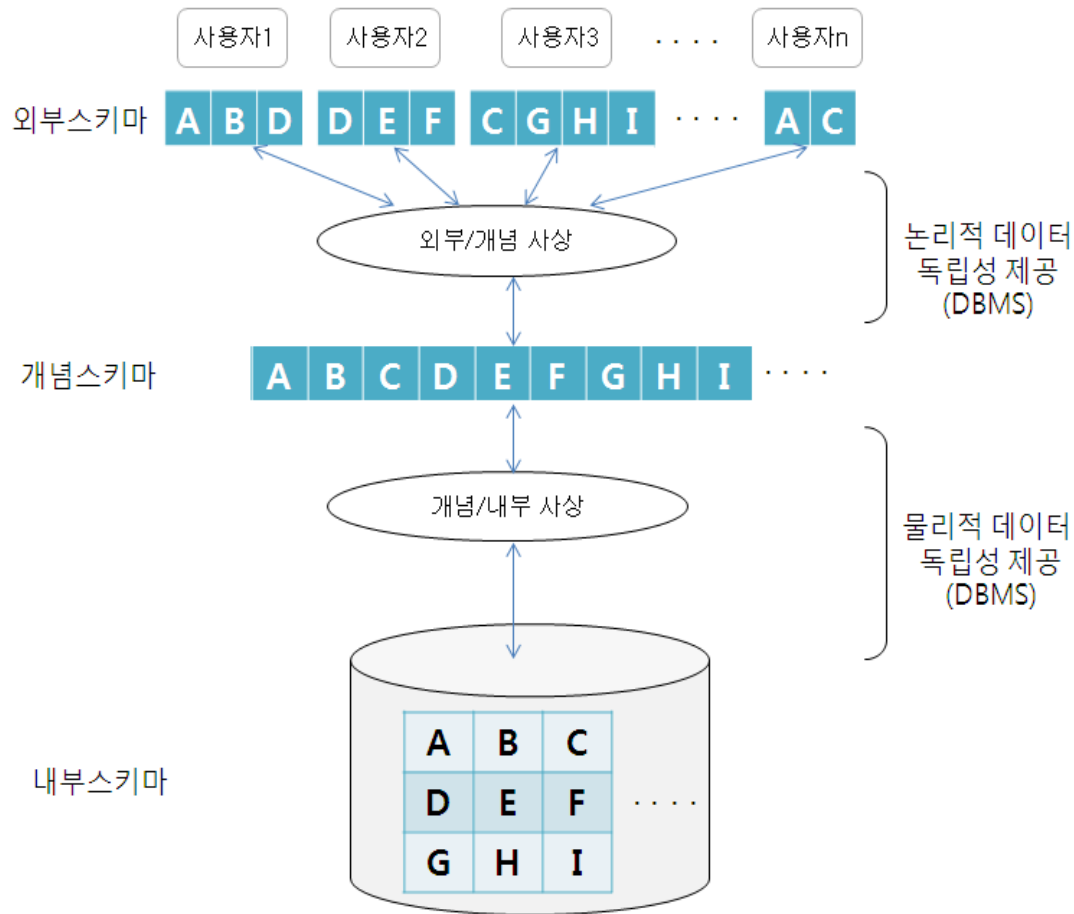
- 사용자 데이터베이스(user database)
 - 사용자나 응용 프로그램이 실제로 이용하는 데이터를 저장하는 일반 데이터베이스
- 시스템 데이터베이스(system database) 또는 시스템 카탈로그(system catalog)
 - 데이터 디렉토리(data directory)
 - 사용자가 접근할 수 없음
 - 데이터 접근에 필요한 위치 정보를 저장하는 저장소
 - 데이터 사전(data dictionary)
 - 데이터베이스에 저장된 모든 부가 정보 즉, 데이터베이스 정의나 명세뿐만 아니라 스키마와 이들 간의 사상 정보, 제약 조건 등을 저장하는 저장소
- 메타 데이터
 - 데이터 사전에 저장된 데이터는 ‘데이터에 관한 데이터(data about data)’임
 - 데이터베이스 자체에 관한 데이터
 - 한 차원 높은 데이터라는 의미에서 메타 데이터(meta data)라고 함
 - DBMS뿐만 아니라 일반 사용자도 검색할 수 있지만 변경은 DBMS만 가능

2.4 데이터 독립성

- 3단계 데이터베이스 구조의 특징
 - 응용 프로그램과 데이터 사이의 독립성을 제공
- 데이터 독립성(data independency)
 - 데이터의 논리적 구조나 물리적 구조가 변경되더라도 응용 프로그램이 영향을 직접 받지 않는 특성
 - 기존 응용 프로그램에 영향을 주지 않으면서 데이터베이스 구조를 변경할 수 있고 반대로 데이터베이스에 영향을 미치지 않으면서 응용 프로그램을 수정할 수 있도록 함
 - 각 단계의 스키마 사이에 적절한 내부 사상을 한다면 하위 스키마가 변경되더라도 상위 스키마에 영향을 주지 않도록 변경 내용을 숨길 수 있음

DBMS의 데이터 독립성 제공

- DBMS의 핵심 역할 중 하나
- DBMS는 3단계 데이터베이스 구조의 스키마 간의 적절한 사상을 통하여 데이터 독립성을 지원



데이터 독립성 제공

1) 외부/개념 사상

- 외부 스키마를 변경하더라도 전체 개념적 스키마는 변경되지 않거나 변경되더라도 변경 내용을 최소화함으로써 다른 응용 프로그램에 주는 영향을 최소화
- 논리적 데이터 독립성(logical data independency) 제공
 - 응용 프로그램에 영향을 최소화하면서 데이터베이스의 논리적 구조를 변경
 - DBMS에 의해 수행되는 외부/개념 사상 즉, 하나의 개념 스키마를 여러 외부 스키마 형태로 사상시킴으로써 가능

2) 개념/내부 사상

- 물리적 구조의 변경에 따른 내부 스키마가 수정되더라도 연관된 개념/내부 사상 정보만 수정하면 상위 스키마에 대한 영향을 최소화
- 물리적 데이터 독립성(physical data independency) 제공
 - 개념적 스키마에 영향을 최소화하면서 데이터베이스의 물리적 구조를 변경
 - DBMS에 의해 수행되는 개념/내부 사상 즉, 하나의 개념 스키마와 내부 스키마 사이의 사상을 통하여 가능