

1_2_데이터베이스

DB(1강 & 2강)

정의

- 공유 데이터
- 통합 데이터
- 저장 데이터
- 운영 데이터

즉 DB란 공유와 통합을 통해 데이터들을 저장하는 운영 데이터의 집합이다

정보 시스템 안에서 데이터를 저장하고 있다가 필요할 땐 제공하는 역할을 한다(sql을 사용해 저장 및 추출 등 가능)

필요성

데이터는 단순한 Fact이지만 가공을 통해 유용해지면 정보로 볼 수 있다

(Data → Information → Knowledge → Wisdom)

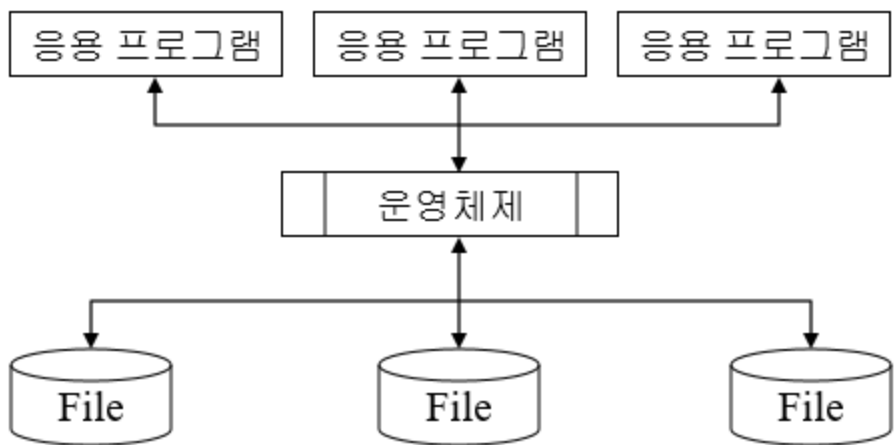
특성

- 실시간 접근(사용자의 요구에 실시간으로 응답)
- 계속 변화(지속적인 C, U, D를 통해 정확한 상태 유지)
- 동시 공유(같은 데이터를 2명 이상이 동시에 사용 가능)
- 내용 기반 참조(데이터가 주소나 위치가 아니라 내용을 기반으로 참조)

DB 관리 시스템 등장 배경

과거의 시스템 관리

과거에는 하나의 파일 시스템으로 각각의 파일들 CRUD를 관리함

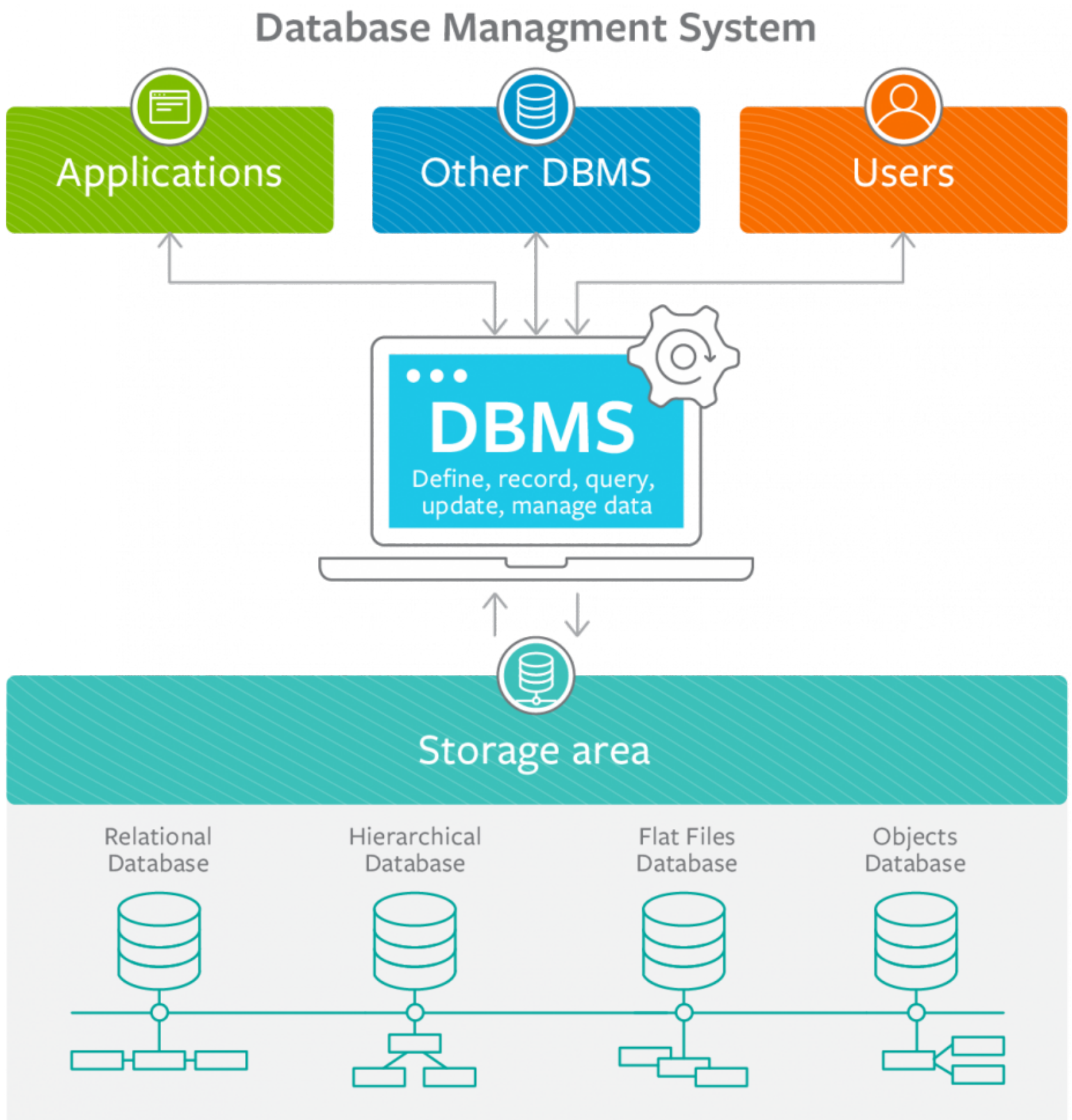


But, 비효율적이고 폐쇄적이고 응용프로그램이 데이터에 종속적이므로 불편함(Ex) 고객접수, 고객상담 등등 각각 구현)

- 데이터 중복성: 데이터의 중복으로 인한 비효율성이 크다(비용 및 수정할때 걸리는 중복성)
- 데이터 종속성: 새로운 row를 추가하려면 응용 프로그램을 새로 만들어야함

현재의 DBMS(Database Management System) 탄생

- DB관리 작업을 하는 시스템
- Data의 무결성 유지
- 통합해서 관리해 파일 시스템의 문제 해결(중복성, 종속성 해결)
- 좋아진 만큼 비용이나 백업등이 복잡해짐



DB 관리 시스템 기능

- 정의(DDL)
- 조작(DML)
- 제어(DCL)

DB 관리 시스템 발전 과정

1세대: 네트워크 DBMS, 계층 DBMS(60~70년대)

2세대: 관계 DBMS(**RDBMS**)

- 테이블 형태로 DB를 구성한 것으로 Oracle, MS SQL, Access 등이 있음
- SQL 언어로 사용

3세대: 객체지향 DBMS, O2, ONTOS / 객체관계 DBMS(객체 DBMS + 관계 DBMS)

3세대보단 2세대가 많지만 2, 3세대는 공존하고 있는 게 현 상황

데이터베이스

03 데이터베이스 시스템

1. 데이터베이스 시스템의 정의

1) 정의

- 데이터베이스에 데이터를 저장, 이를 관리하여 조직에 필요한 정보를 생성해주는 시스템
- 컴퓨터에 있는 데이터베이스, 이를 관리하는 데이터베이스 관리 시스템, 사용자와의 소통을 위한 데이터 언어 까지 일컫는 말

2. 데이터베이스의 구조

1) 정의

- 스키마
 - 데이터베이스에 저장되는 데이터 구조와 제약조건을 정의한 것



고객번호	이름	나이	주소
INT	CHAR(10)	INT	CHAR(20)

그림 3-2 스키마의 예

- 전체적인 뼈대를 정의
 - 동적으로 변하지 X
- 인스턴스
 - 스키마에 따라 데이터베이스에 실제로 저장된 값
 - 스키마와 달리 변동성이 있다(고객 추가와 삭제)

- employee 예시

Employee

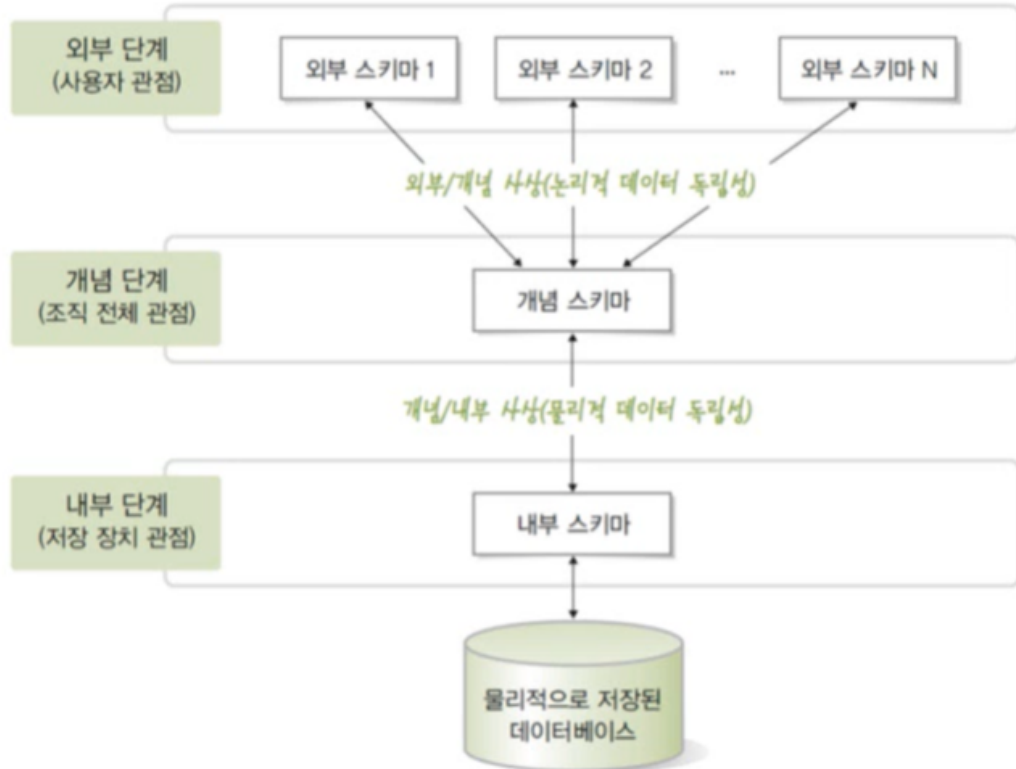
eNo	name	ssn	salary	dept
1	Smith, Joseph	111-11-1111	40000	3
2	Jones, David	222-22-2222	32000	2
3	Olson, Jane	333-33-3333	38000	3
4	Neff, Arnold	444-44-4444	23000	1
5	Homes, Denise	555-55-5555	14000	1
6	Naumi, Susan	666-66-6666	35000	2
7	Young, John	777-77-7777	30000	3

- 스키마: eNo, name, ssn, salary, dept로 구성 + 그에 해당하는 타입
- 인스턴스: 실제 값

2) 3단계 데이터베이스 구조

- 3단계 나누기

- 외부단계: 개별 사용자 관점
- 개념단계: 조직 전체의 관점
- 내부단계: 물리적인 저장 장치의 관점



- 외부단계(사용자 관점)

- 데이터베이스를 개별 사용자 관점에서 이해하고 표현하는 단계
- 추상화가 많이 이루어진 단계 (내부 → 외부 단계로 갈수록 높아짐)
- 데이터베이스 하나에 여러 외부 스키마가 존재 할 수 있다.
- EX) 집주인 관점과 유사
 - 2001호 거주자는 2001호의 데이터만 궁금할 것

- * 외부 스키마(서브 스키마)

- 각 사용자가 생각하는 데이터베이스의 모습
- 외부단계에서 사용자에게 필요한 데이터베이스를 정의한 것

EX) 고객 분석팀 → 성별, 나이, 직업 필드를 가진 외부 스키마

상품 배송팀 → 고객번호, 이름, 주소, 연락처 필드를 가진 외부 스키마

- 개념단계(조직 전체 관점)

- 스키마의 형태를 저장
- 데이터를 어떻게 개념화 해야할지 생각하는 단계
- 조직 전체의 관점에서 이해하고 표현하는 단계
- 데이터베이스 하나당 하나의 개념 스키마 존재
- EX) 아파트 관리인 관점과 유사
 - o 2001호 뿐만 아니라 101동 ~ 109동 모든 데이터에 대한 개념

* 개념 스키마

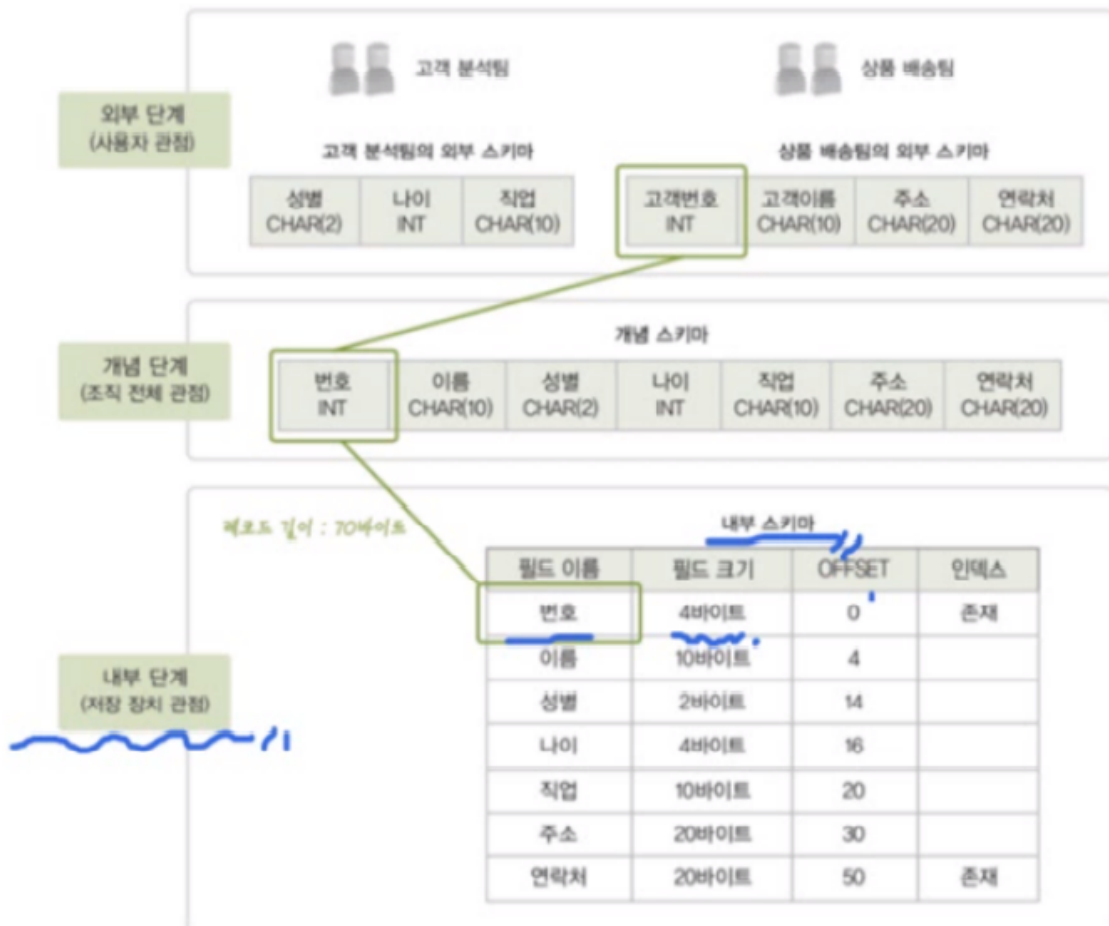
- 개념 단계에서 데이터베이스 전체의 논리적 구조를 정의한 것
- 데이터간 관계, 제약조건, 보안정책, 접근 권한 등에 대한 정의를 포함

● 내부단계(저장 장치 관점)

- 데이터베이스를 저장장치의 관점에서 이해하고 표현하는 단계
- 내부 스키마를 어떤 식으로 가져와야할지
- 데이터베이스 하나당 하나의 내부 스키마 존재
- EX) 건설 업체의 관점과 유사
 - o 공사를 어떻게 할지, 뼈대를 어떻게 구축할지
 - o 물리적으로 어떻게 조작하고 운영할지 고민하는 단계

* 내부 스키마

- 저장 장치에 실제로 저장되는 방법을 정의
- 레코드 구조, 필드 크기, 레코드 접근 등 물리적 저장 구조를 의미

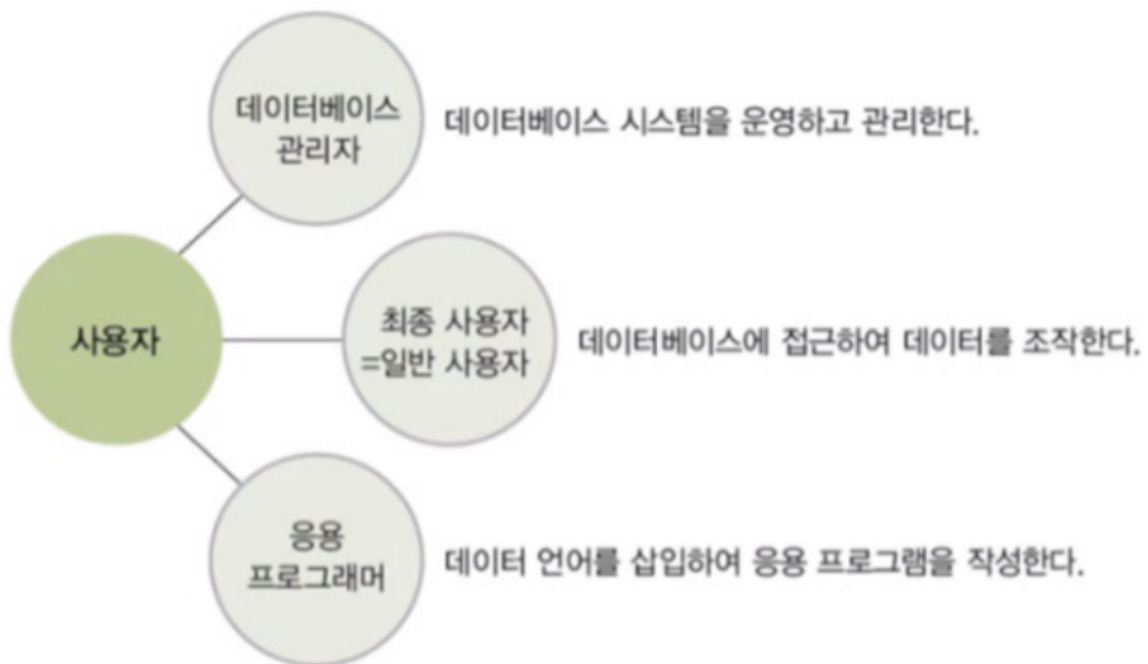


3) 3단계 데이터베이스 구조의 매핑

- 스키마 사이의 대응 관계
 - 외부/개념 사상(응용 인터페이스): 외부 스키마 - 개념 스키마
 - 개념/내부 사상(저장 인터페이스): 개념 스키마 - 내부 스키마
 - 미리 정의된 사상 정보를 이용해 사용자가 원하는 데이터에 접근
 - 데이터 독립성의 실현이 가능
- HOW? 하위 스키마를 변경해도 상위 스키마가 영향을 받지 않으므로
- 논리적 데이터 독립성
 - 개념 스키마가 변경되어도 외부 스키마에 영향 X
 - 개념 스키마 변경시 관련 외부/개념 사상만 수정하면 됨
 - 물리적 데이터 독립성
 - 내부 스키마가 변경되어도 개념 스키마에 영향 X
 - 내부 스키마 변경시 관련 개념/내부 사상만 수정하면 됨

3. 데이터베이스 사용자

이용 목적에 따라 데이터베이스 관리자, 최종 사용자, 응용 프로그래머로 구분



1) 데이터베이스 관리자

- DB 시스템을 운영 및 관리
- 주로 데이터 정의어(테이블 만드는)와 제어어를 사용

- 주요 업무
- 데이터베이스 구성 요소 선정
- 데이터베이스 스키마 정의
- 물리적 저장 구조와 접근 방법 결정
- 무결성 유지를 위한 제약조건 정의
- 보안 및 접근 권한 정책 결정
- 백업 및 회복 기법 정의
- 시스템 데이터베이스 관리
- 시스템 성능 감시 및 성능 분석
- 데이터베이스 재구성

2) 최종 사용자(일반 사용자)

- DB에 접근해 DB를 조작(삽입, 검색, 수정, 삭제 등)
- 주로 데이터 조작어를 사용
- 캐주얼 사용자, 초보 사용자로 구분

3) 응용 프로그래머

- 데이터 언어를 삽입하여 응용 프로그램을 작성
- 주로 데이터 조작어를 사용
- EX) 고객 관련 데이터에 대한 이해도 필요

4. 데이터 언어

데이터 언어 : 사용자와 데이터베이스 관리 시스템간 통신 수단

1) 데이터 정의어(DDL): 스키마를 정의, 수정, 삭제를 위해 사용

2) 데이터 조작어(DML): 데이터 삽입, 삭제, 수정, 검색 등의 처리를 요구하기 위해 사용

- 절차적 데이터 조작어
 - 사용자가 어떤 데이터를 원하고(WHAT) 그 데이터를 얻기 위해 어떻게 처리해야하는지(HOW)도 설명
- 비절차적 데이터 조작어
 - 사용자가 어떤 데이터를 원하는지만(WHAT) 설명
 - SQL은 여기에 해당

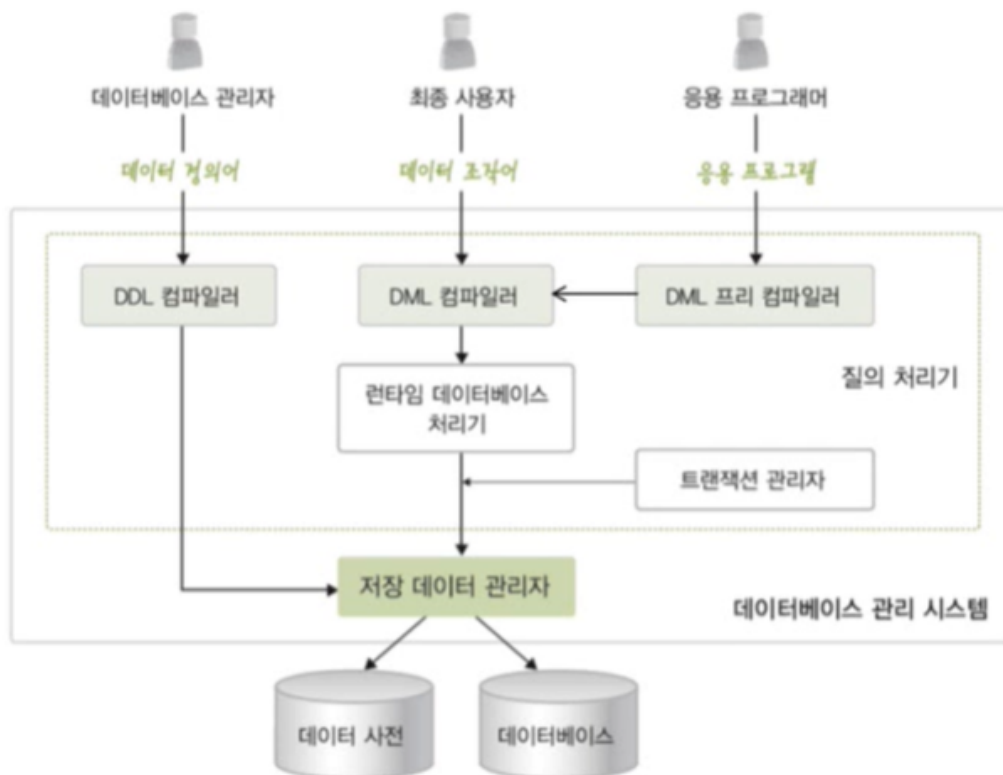
3) 데이터 제어어(DCL): 내부적으로 필요한 규칙 및 기법을 정의하기 위해 사용

- 사용목적
 - 무결성: 정확하고 유효한 데이터만 유지
 - 보안: 허가된 사용자에게 권한 부여(허가받지 X 사용자의 데이터 접근 차단)
 - 회복: 장애가 발생해도 데이터 일관성 유지
 - 동시성 제어: 동시 공유 지원

5. 데이터베이스 관리 시스템

데이터베이스 관리와 사용자의 데이터 처리 요구 수행

- 주요 구성 요소
 - 질의 처리기
 - 사용자의 데이터 처리 요구를 해석하여 처리
 - 저장 데이터 관리자
 - 디스크에 저장된 사용자 데이터베이스와 데이터 사전 관리
 - 실제 접근 역할



데이터베이스 시스템 iv

데이터베이스 설계(1-3단계)



데이터모델링 과 모델

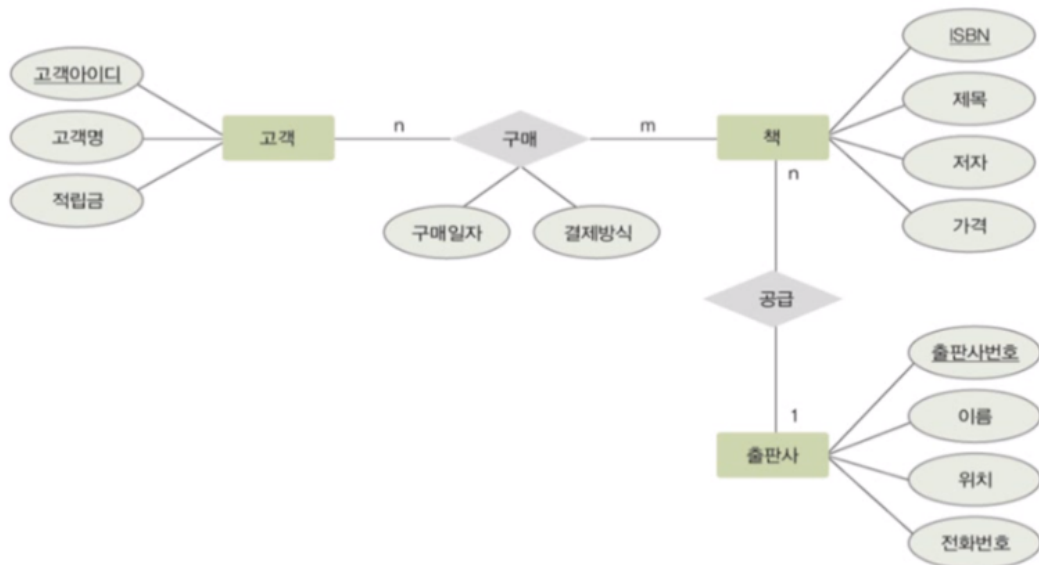
- 모델링
 - 현실세계 데이터를 가상 세계 데이터로 만드는 작업
 - 개념적 데이터 모델링(추상화) : 현실 세계의 특성을 분석하여 개념화하는 작업 / 2단계
 - 논리적 설계 : 개념 세계를 데이터베이스 구조에 맞게 저장하는 작업 / 3단계



- 데이터 모델
 - 모델링의 결과물
 - 개념적 데이터 모델
 - 현실세계를 데이터베이스의 개념적 구조로 표현
 - 개체-관계 모델(Entity-Relationshipmodel)
 - 논리적 데이터 모델
 - 개념적 구조를 논리적 모델링하여 데이터베이스로 표현
 - 관계 데이터 모델(E-R diagram)

개체-관계 모델(Entity-Relationshipmodel)

- 개체- 관계 모델
 - 개체 간 관계를 이용해 현실 세계를 개념적 구조로 표현
 - 핵심 요소 : 개체, 속성, 관계
- 개체- 관계 다이어그램
 - E-R 다이어그램
 - 개체- 관계 모델링을 그림으로 표현



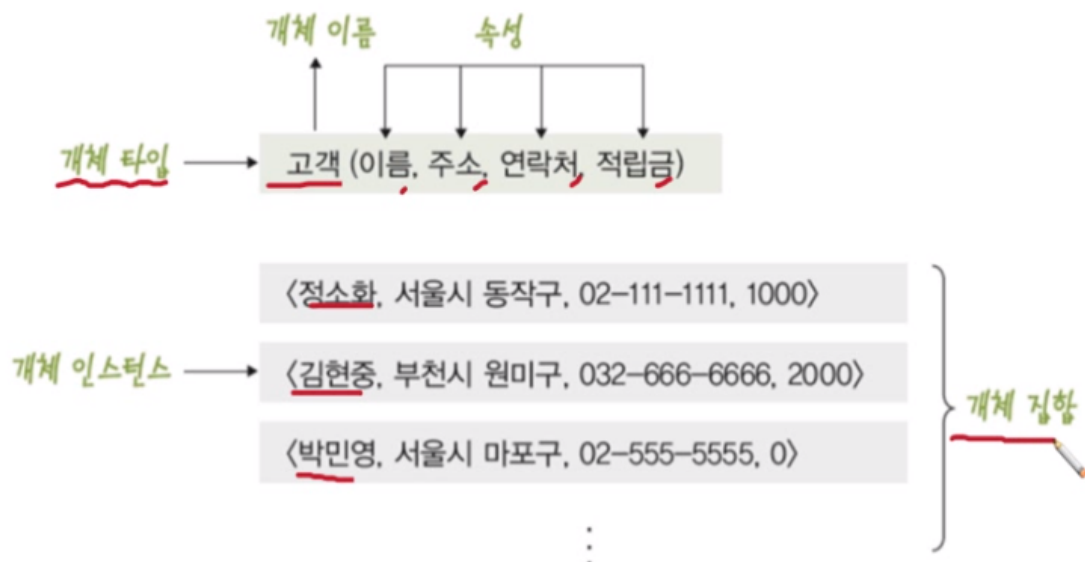
- 개체 (네모) : 고객, 책, 출판사
- 관계 (마름모) : 구매, 공급
- 속성(타원) : 출판사번호(key: 밑줄로 구분), 고객명 등...

개체 관계 모델 용어

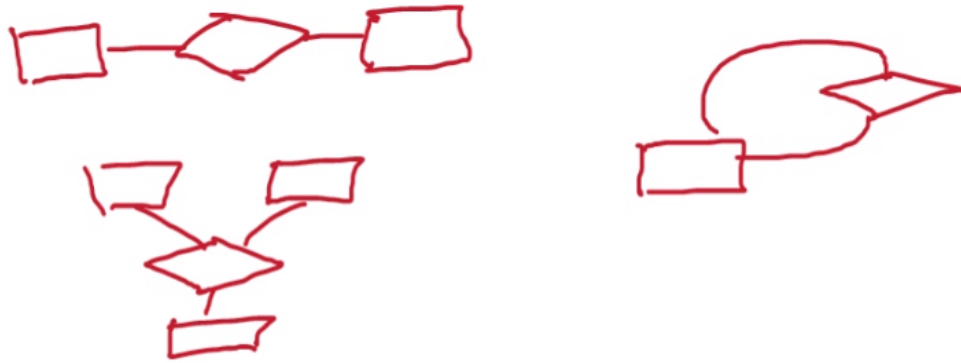
- 개체
 - 사각형으로 표시
 - 현실에서 사람, 사물처럼 구별되는 모든 것
 - 저장 가치와 이름 고유 특성(속성)이 필요하다

- 서점 개체: 고객, 책
- 학교 개체: 학과, 과목
- 병원 개체: 의사, 병실

- 속성
 - 타원으로 표시
- 의미 있는 데이터의 가장 작은 논리적 단위
 - 개체나 관계가 가진 고유 특성
- 개체 타입
 - 개체를 고유 이름과 속성으로 정의
 - 파일 구조의 레코드 타입에 대응
- 개체 인스턴스
 - 개체 속성이 실제 값을 가져서 실체화
 - 개체 어커런스라고도 함
 - 파일 구조 레코드 인스턴스에 대응
- 개체 집합
 - 특정 개체 타입에 대한 개체 인스턴스를 모아놓음



- 관계
 - 개체사이의 대응관계 매핑을 의미
 - 마름모로 표현
 - 이항/ 삼항/ 순환(자기 자신과) 관계로 나눈다.



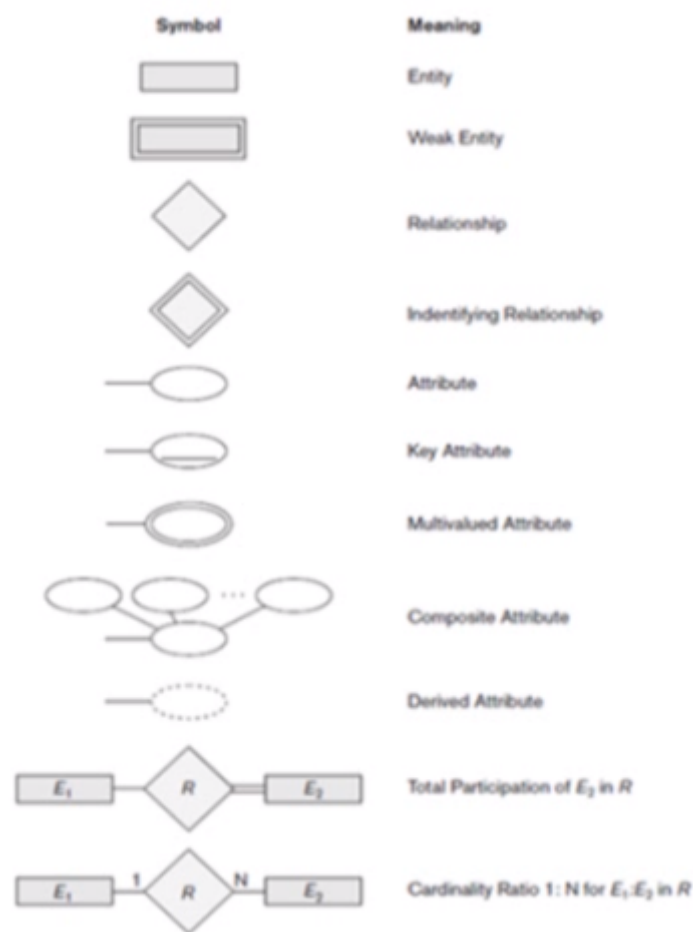
- 관계 유형
 - 매핑 카디널리티(각 개체가 연관성을 맺는 상대 개체 집합의 인스턴스 개수)를 기준으로 결정
 - 1:1
 - 1:n
 - m:n

속성



- 단일 속성: 이름
- 다중 속성: 연락처
 - 일반 타원이 아닌 이중 타원으로 표시!
- 단순 속성: 분해 불가, id
- 복합 속성: 생년월일 => 년 / 월 / 일
 - 속성 밑에 속성이 추가로 존재
- 유도 속성: 하나의 속성으로 부터 유도 되는 속성

- 값이 따로 저장되지 않음
 - 주민번호 => 나이, 가격, 할인율로 부터 실판매가 유도
 - 점성 타원으로 표시
- 널 속성:
 - 아직 모르거나 존재하지 않는 값이 허용되는 속성
 - 키 속성:
 - 인스턴스 객체를 구별 할 수 있다.
 - 둘 이상의 속성을 이용하여 사용하기도 함(이름 + 생일)



개체, 관계, 속성, 키속성, 다중속성, 복합속성, 유도 속성, 절대적 참여, 1:n 관계

관계

- 1:1 관계
 - 하나 당 하나랑만 관계를 맺을 수 있음
 - 부부
- 1:n 관계
 - 하나가 여러개와 관계를 맺을 수 있음

- 부서와 사원
- m:n 관계
 - 여러개와 여러개가 관계를 맺을 수 있음
 - 고객과 책
- 참여
 - 필수적 참여(모든 개체 인스턴스가 참여하여야함, 모든 회사원은 어떤 부서에 포함)
 - 선택 적 참여(일부만 참여해도 괜찮, 책을 구매 안 한 고객이 있어도 됨)
- 종속성
 - 약한 개체: 오너가 필요한 개체
 - 오너 개체: 다른 개체의 존재 여부를 결정하는 개체
 - 일반적으로 오너와 약한 개체는 일대다 관계를 가지고, 약한 개체는 필수참여하는 특징이 있다
 - 약한 개체는 오너개체의 키를 포함하여 키를 구성하는 특징이 있다
 - 관계 = 이중 마름모, 약한 개체 = 이중 사각형으로 표현

