



01.

데이터베이스 시스템

목차

01

데이터베이스와 데이터베이스 시스템

02

데이터베이스 시스템의 발전

03

파일 시스템과 DBMS

04

데이터베이스 시스템의 구성

학습목표

- ❖ 데이터베이스의 유형을 알아보고 개념 및 특징을 이해한다.
- ❖ 데이터베이스 시스템을 중심으로 정보 시스템의 발전 과정을 알아본다.
- ❖ 프로그램과 데이터가 컴퓨터에 어떻게 저장되는지 이해한다.
- ❖ 데이터베이스 시스템의 구성요소를 알아본다.

01 데이터베이스와 데이터베이스 시스템

1. 데이터, 정보, 지식
2. 일상생활의 데이터베이스
3. 데이터베이스의 정의 및 특성
4. 데이터베이스 시스템의 구성



1. 데이터, 정보, 지식

- 데이터 : 관찰/측정을 통해 수집된 사실(Fact) 또는 값(Value)이 특정 기준에 의해 정리된 것.
 - 예) 오늘의 온도, 습도, 강수량 등
- 정보 : 어떤 목적에 의해 데이터를 해석하거나 가공한 형태
 - 예) 예상 온도, 강수 확률 등
- 지식 : 연결된 정보들의 패턴을 바탕으로 경험·학습·이해·추론을 통해 처리된 결과물



그림 1-1 데이터, 정보, 지식

2. 일상생활의 데이터베이스

❖ 데이터베이스란?

조직에 필요한 정보를 얻기 위해 논리적으로 연관된 데이터를 모아 구조적으로 통합해 놓은 것

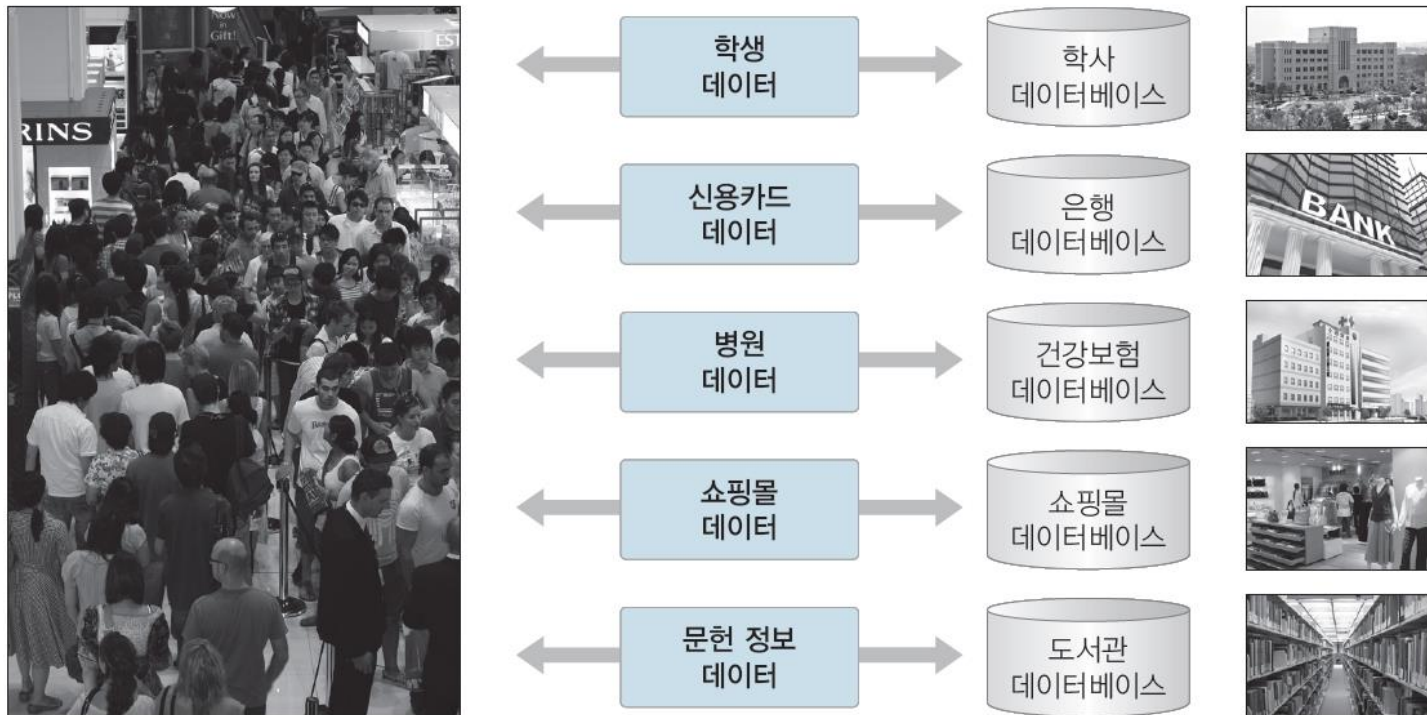


그림 1-2 일상생활에서 생성되는 데이터베이스

2. 일상생활의 데이터베이스

UU Again

***** 최근영수증발행인쇄 *****
 CU 김통점
 사업자등록번호: 1751350895
 강원도 김천시 수촌로 64 (지아동,
 강릉아파트)
 최 회 TEL: 033-643-6434

정부 방침에 의해 12년 7월 1일부터
 현금 결제 취소시, 영수증이 없으면
 교환/환불이 불가능합니다.

33177 2019-05-19(일) POS-01

달콤한대통과자1200	1	1,200
총 구 매 액	1	1,200
과세물품가액		1,091
부 가 세		109
*결 제 금 액		1,200
신 용 카 드		1,200

***** 신 용 카 드 *****
 카드번호: 6161-07**-****-0303
 카드회사: 001 비씨
 할부개월: 00 승인번호: 43030033
 결제금액: 1,200

*표시 상품은 부가세 면세 품목 임.
 환불: 30일내 영수증/카드자참시 가능
 고객센터: 14 담당: 최 회 NO: 6193 13:03

95119051933177093193

KORAIL

열차승차권
 Train Ticket

승차일자 2019년 09월 22일 (금/Fri)
 WWW.CO

출발 From **송주** 도착 To **용산**
 Yongsan Yongsan

16:39 → **18:33**
 ※승차일시와 이용구간을 반드시 확인하시기 바랍니다.

Train No.
4188 열차 **KTX** 신천

타는 곳 번호 Track	호차번호 Car No.	좌석번호 Seat No.
역 전망관 확인	17 호차 일반실	18A (순방향)

영수액 ₩ 42,100원

운임요금₩	46,800원
환급금₩₩	4,700원
(부가세포함)	
신용728720	

발행일자 03/15 15:37

고객센터 전화번호 : 1544-7795
 전화번호 : 1544-6787

그림 1-3 편의점과 철도회사의 데이터베이스 : 간단한 거래도 많은 데이터가 포함된다

2. 일상생활의 데이터베이스

표 1-1 데이터베이스의 활용 분야

분야	활용
생활과 문화	<ul style="list-style-type: none">• 기상정보 : 날씨 관련 정보를 제공• 교통정보 : 교통상황 관련 정보를 제공• 문화예술정보 : 공연이나 인물에 관한 정보를 제공
비즈니스	<ul style="list-style-type: none">• 금융정보 : 금융, 증권, 신용에 관한 정보를 제공• 취업정보 : 노동부와 기업의 채용 관련 정보를 제공• 부동산정보 : 공공기관이나 민간의 토지, 매물, 세금 정보를 제공
학술정보	<ul style="list-style-type: none">• 연구학술정보 : 논문, 서적, 저작물에 관한 정보를 제공• 특허정보 : 특허청의 정보를 기업과 연구자에게 제공• 법률정보 : 법제처와 대법원의 법률에 관한 정보를 제공• 통계정보 : 국가기관의 통계에 관한 정보를 제공

2. 일상생활의 데이터베이스

- 데이터베이스 시스템은 데이터의 검색과 변경 작업을 주로 수행함
- 변경이란 시간에 따라 변하는 데이터 값을 데이터베이스에 반영하기 위해 수행하는 삽입, 삭제, 수정 등의 작업을 말함

표 1-2 검색과 변경 빈도에 따른 데이터베이스 유형

유형	검색 빈도	변경 빈도	데이터베이스 예	특징
유형1	적다	적다	공통 데이터베이스	<ul style="list-style-type: none"> • 검색이 많지 않아 데이터베이스를 구축할 필요 없음 • 보존가치가 있는 경우에 구축
유형2	많다	적다	도서 데이터베이스	<ul style="list-style-type: none"> • 사용자 수 보통 • 검색은 많지만 데이터에 대한 변경은 적음
유형3	적다	많다	비행기 예약 데이터베이스	<ul style="list-style-type: none"> • 예약 변경/취소 등 데이터 변경은 많지만 검색은 적음, 검색은 변경을 위하여 먼저 시도됨 • 실시간 검색 및 변경이 중요함
유형4	많다	많다	증권 데이터베이스	<ul style="list-style-type: none"> • 사용자 수 많음 • 검색도 많고 거래로 인한 변경도 많음

구축이
쉬움

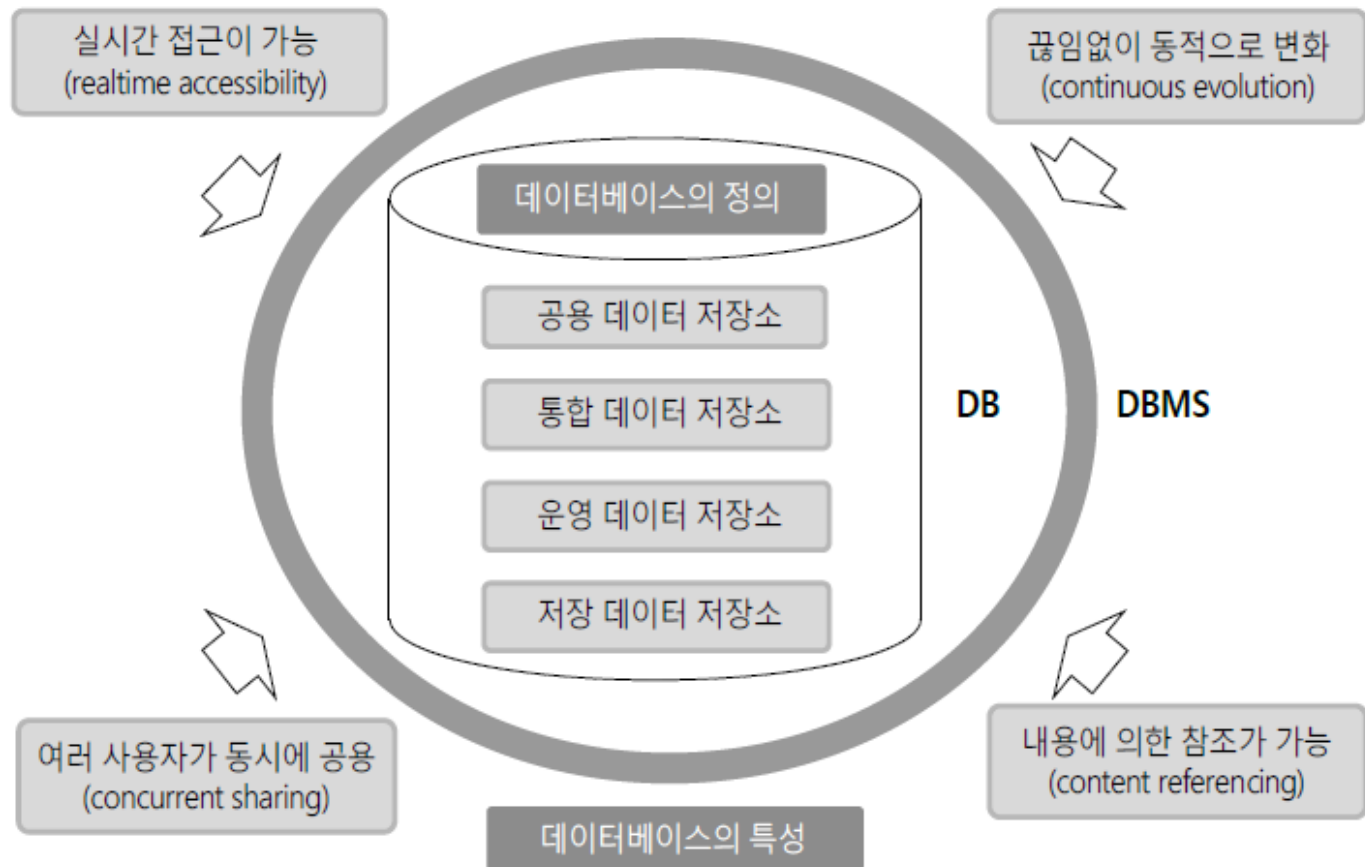


구축이
어려움

3. 데이터베이스의 정의 및 특성

❖ 데이터베이스

- 여러 사용자나 응용 프로그램들이 함께 사용할 목적으로 체계적으로 통합하여 저장한 운영에 꼭 필요한 필수 데이터들의 저장소



3. 데이터베이스의 정의 및 특성

❖ 데이터베이스의 정의

❶ 통합된 데이터(integrated data)

데이터를 통합하는 개념으로, 각자 사용하던 데이터의 중복을 **최소화하여 중복**으로 인한 데이터 불일치 현상을 제거

- 여러 장소에 흩어져 있더라도 상호 연결되어 접근할 수 있는 논리적 통합을 뜻함

❷ 저장된 데이터(stored data)

문서로 보관된 데이터가 아니라 디스크, 테이프 같은 컴퓨터 저장장치에 저장된 데이터를 의미

❸ 운영 데이터(operational data)

조직의 목적을 위해 사용되는 데이터, 즉 업무를 위한 검색을 할 목적으로 저장된 데이터

❹ 공용 데이터(shared data)

한 사람 또는 한 업무를 위해 사용되는 데이터가 아니라 공동으로 사용되는 데이터를 의미

3. 데이터베이스의 정의 및 특성

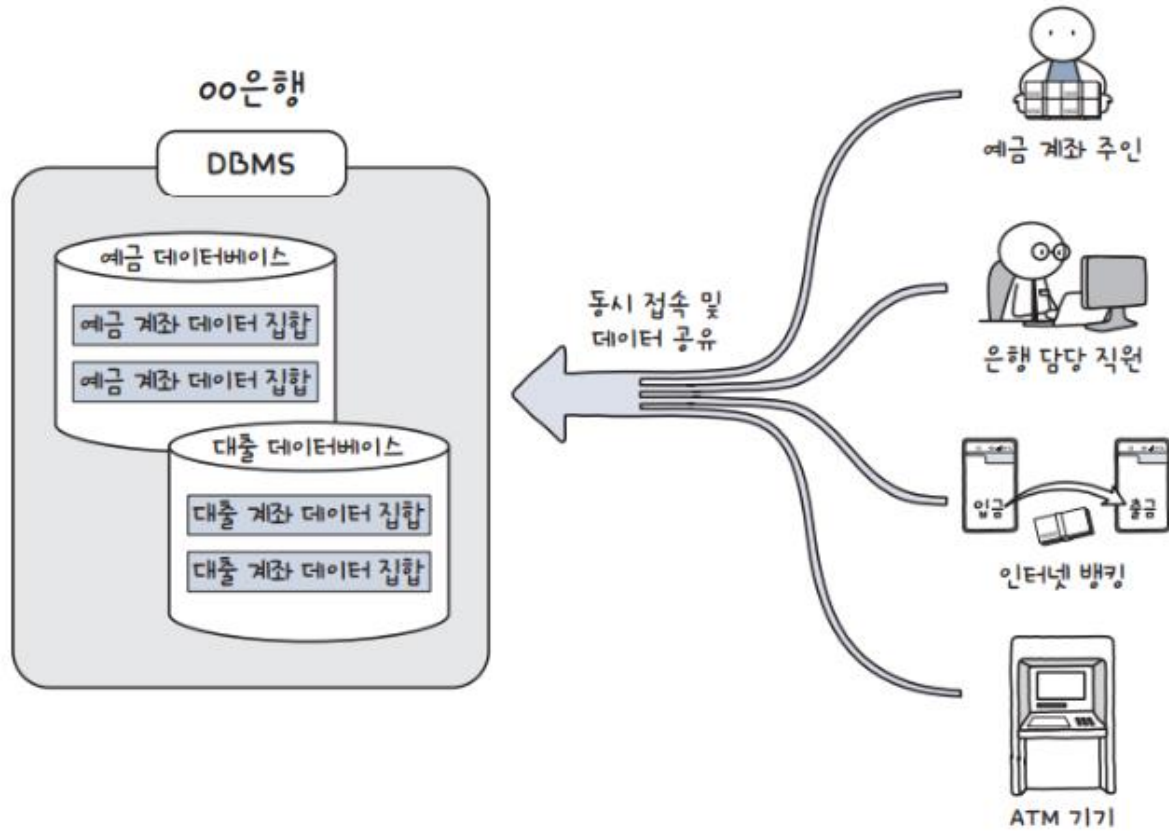


그림 1-4 데이터베이스의 개념 : 데이터베이스는 운영 데이터를 통합하여 저장하며 공용으로 사용된다

3. 데이터베이스의 정의 및 특성

❖ 데이터베이스의 특성

❶ 실시간 접근성(real time accessibility)

데이터베이스는 실시간으로 서비스된다. 사용자가 데이터를 요청하면 몇 시간이나 몇 일 뒤에 결과를 전송하는 것이 아니라 수 초 내에 결과를 서비스한다.

예) 영화 예매를 위해 잔여 좌석수를 검색하는데 수 분 이상이 소요되면 곤란함

❷ 계속적인 변화(continuous change)

데이터베이스에 저장된 내용은 어느 한 순간의 상태를 나타내지만, 데이터 값은 시간에 따라 항상 바뀐다. 데이터베이스는 삽입, 삭제, 수정 등의 작업을 통하여 바뀐 데이터 값을 저장한다.

예) 팔고자 하는 주식 시세가 현재 시세가 아닌 몇 분 전의 과거 시세라면 곤란함

❸ 동시 공유(concurrent sharing)

데이터베이스는 서로 다른 업무 또는 여러 사용자에게 동시에 공유된다. 동시는 병행이라고도 하며, 데이터베이스에 접근하는 프로그램이 여러 개 있다는 의미이다.

예) 사용자 수가 많다고 해서 저장소에 접근이 제한되고 순번을 받고 대기하는 일은 없어야 함

❹ 내용에 따른 참조(reference by content)

데이터베이스에 저장된 데이터는 데이터의 물리적인 위치가 아니라 데이터 값에 따라 참조된다.

예) 검색할 학생 이름만 키워드로 입력하면 '홍길동' 학생의 모든 신상 데이터를 확인할 수 있어야 함

4. 데이터베이스 시스템의 구성

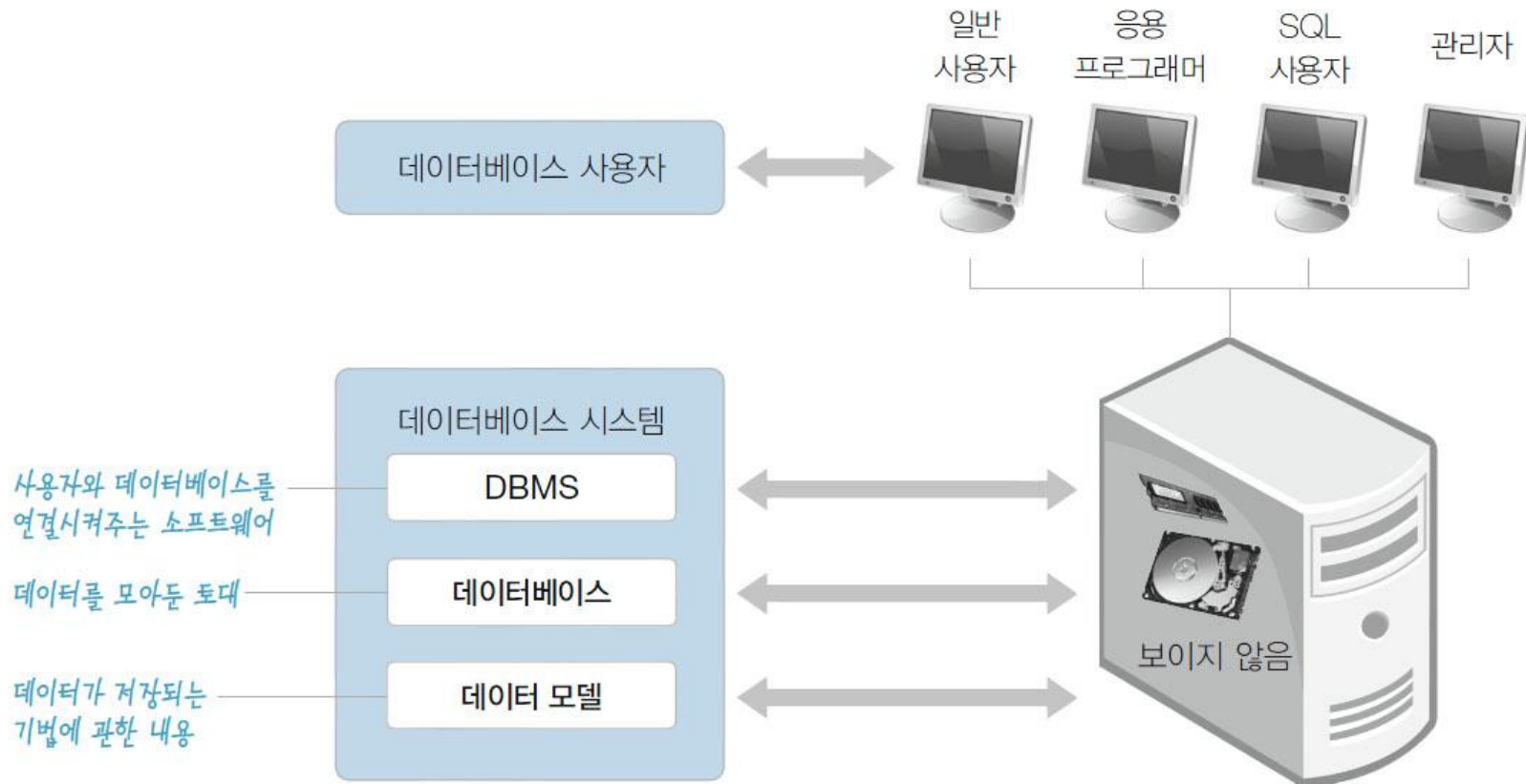


그림 1-5 데이터베이스 시스템의 구성 요소와 물리적인 위치

02 데이터베이스 시스템의 발전

1. 데이터베이스 시스템의 예 : 마당서점
2. 정보 시스템의 발전



1. 데이터베이스 시스템의 예 : 마당서점

❖ [1단계] 마당서점의 시작



-
- 도서 : 100권
 - 고객 : 근처 학교의 학생, 지역 주민
 - 업무 : 회계 업무(계산기 사용), 장부에 기록
 - 고객 서비스 : 사장이 직접 도서 안내
-

그림 1-6 마당서점 초기

1. 데이터베이스 시스템의 예 : 마당서점

❖ [2단계] 컴퓨터의 도입

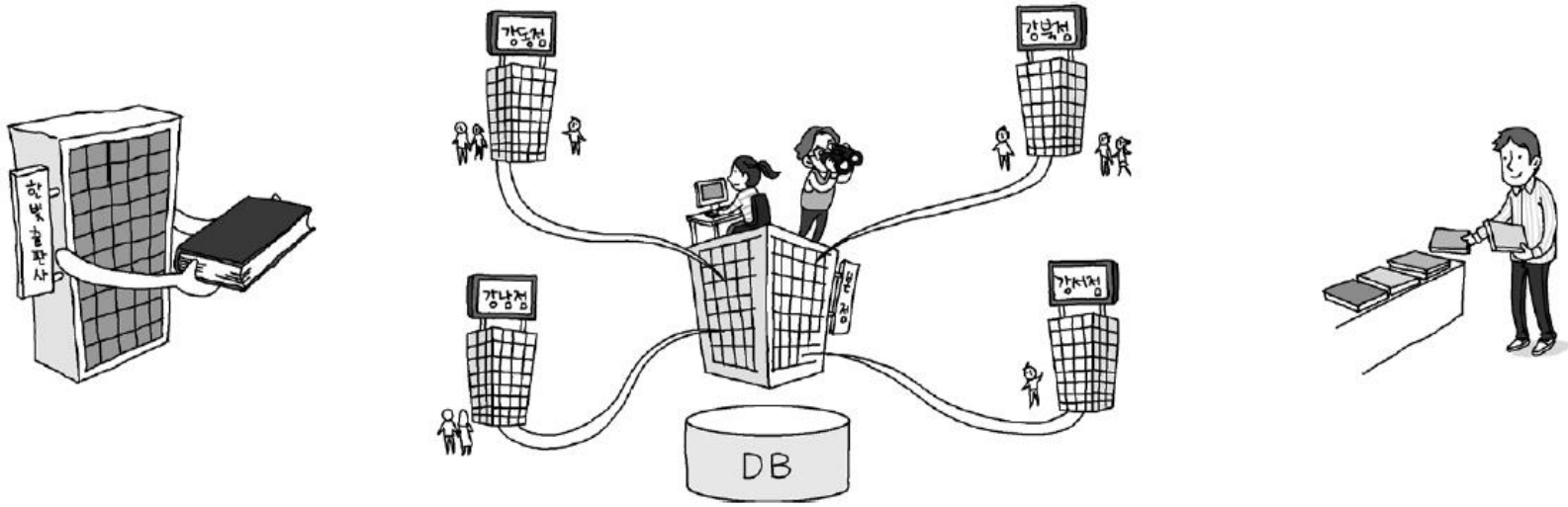


- 도서 : 1,000권
- 고객 : 근처 학교의 학생, 지역 주민
- 업무 : 회계 업무(컴퓨터 사용), 파일 시스템
- 고객 서비스 : 컴퓨터를 이용하여 도서 검색, 직원 고용

그림 1-7 마당서점 전산화

1. 데이터베이스 시스템의 예 : 마당서점

❖ [3단계] 지점 개설 및 데이터베이스 구축

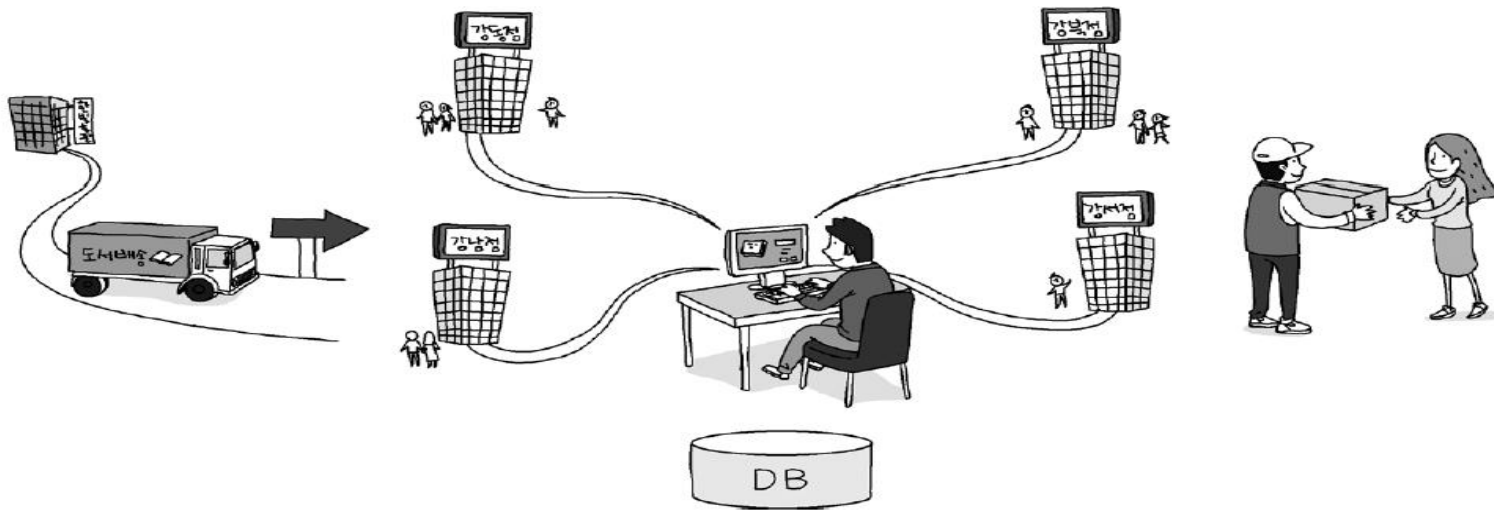


- 도서 : 10,000권
- 고객 : 서울 지역 고객
- 업무 : 회계 업무(컴퓨터 사용), 데이터베이스 시스템
- 고객 서비스 : 클라이언트/서버 시스템으로 지점을 연결하여 도서 검색 서비스 제공

그림 1-8 마당서점 DBMS 도입

1. 데이터베이스 시스템의 예 : 마당서점

❖ [4단계] 홈페이지 구축

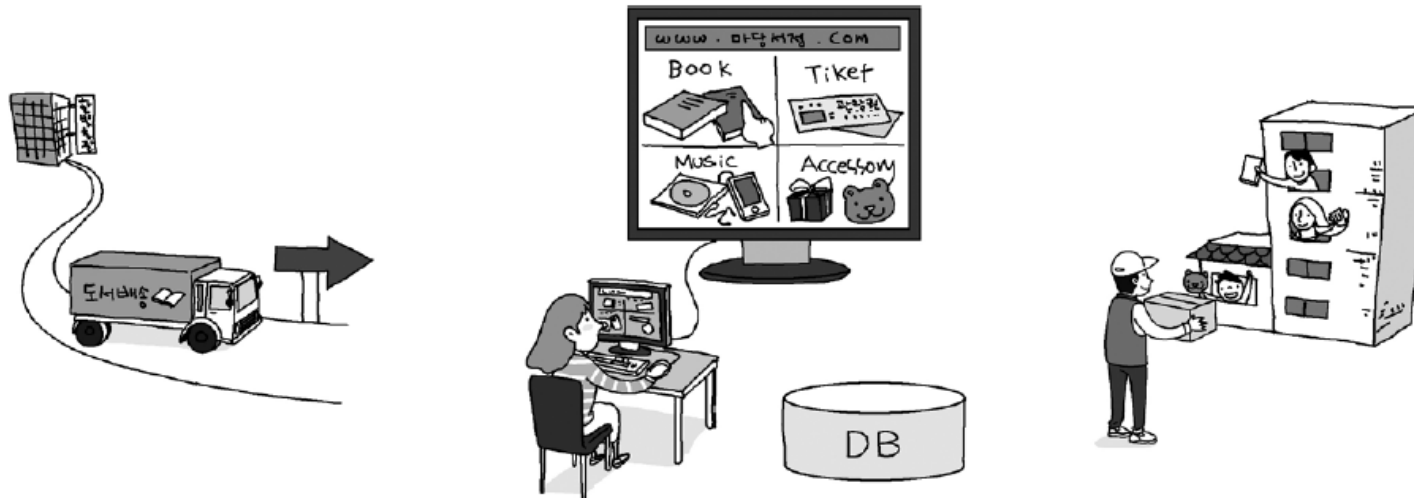


- 도서 : 100,000권
- 고객 : 국민(전국으로 배송)
- 업무 : 회계/인사 업무(컴퓨터와 인터넷 사용), 웹 DB 시스템으로 지점 간 연계
- 고객 서비스 : 인터넷으로 도서 검색 및 주문

그림 1-9 마당서점 인터넷 서비스 실시

1. 데이터베이스 시스템의 예 : 마당서점

❖ [5단계] 인터넷 쇼핑몰 운영



- 도서 : 1,000,000권
- 고객 : 국민(전국으로 배송)
- 업무 : 회계/인사 업무(컴퓨터와 인터넷 사용), DB 서버 여러 개 구축
- 고객 서비스 : 인터넷 종합 쇼핑 서비스 제공

그림 1-10 마당서점 인터넷 쇼핑몰 운영

1. 데이터베이스 시스템의 예 : 마당서점

표 1-3 정보통신기술의 발전과 마당서점의 성장

단계	시기	주요 특징
	정보기술	
1단계 마당서점	1970년대	<ul style="list-style-type: none">• 사장이 모든 도서의 제목과 가격을 기억• 매출과 판매가 컴퓨터 없이 관리됨• 매출에 대한 내용이 정확하지 않음
	컴퓨터 없음	
2단계 초기전산화	1980년대	<ul style="list-style-type: none">• 컴퓨터를 이용한 초기 응용 프로그램으로 업무 처리• 파일 시스템 사용• 한 대의 컴퓨터에서만 판매 및 매출 관리
	컴퓨터	
3단계 데이터베이스	1990년대	<ul style="list-style-type: none">• 지점 간 클라이언트/서버 시스템을 도입하여 업무 처리• 데이터베이스 관리 시스템(DBMS)을 도입
	컴퓨터+원격통신	
4단계 홈페이지 구축	2000년대	<ul style="list-style-type: none">• 인터넷을 이용하여 도서 검색 및 주문• 웹 DB 시스템으로 불특정 다수 고객 유치• 고객이 지리적으로 넓게 분산됨
	컴퓨터+인터넷	
5단계 인터넷 쇼핑몰	2010년대	<ul style="list-style-type: none">• 도서뿐만 아니라 음반, 액세서리, 문구, 공연 티켓까지 판매하는 인터넷 쇼핑몰로 확대• 도서 외 상품의 매출 비중이 50% 이상으로 늘어남
	컴퓨터+인터넷	

2. 정보 시스템의 발전

❶ 파일 시스템

- 데이터를 파일 단위로 파일 서버에 저장
- 각 컴퓨터는 LAN을 통해 파일 서버에 연결, 파일 서버에 저장된 데이터를 사용하기 위해 각 컴퓨터의 응용 프로그램에서 열기/닫기(open/close)를 요청
- 각 응용 프로그램이 독립적으로 파일을 다루기 때문에 데이터가 중복 저장될 가능성이 있음
- 동시에 파일을 다루기 때문에 데이터의 일관성이 훼손될 수 있음

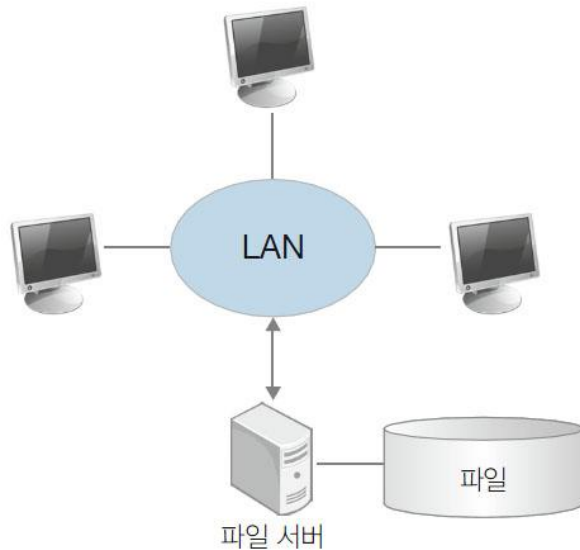


그림 1-11 파일 시스템

2. 정보 시스템의 발전

② 데이터베이스 시스템

- DBMS를 도입하여 데이터를 통합 관리하는 시스템
- DBMS가 설치되어 데이터를 가진 쪽을 서버(server), 외부에서 데이터 요청하는 쪽을 클라이언트(client)라고 함
- DBMS 서버가 파일을 다루며 데이터의 일관성 유지, 복구, 동시 접근 제어 등의 기능을 수행
- 데이터의 중복을 줄이고 데이터를 표준화하며 무결성을 유지함

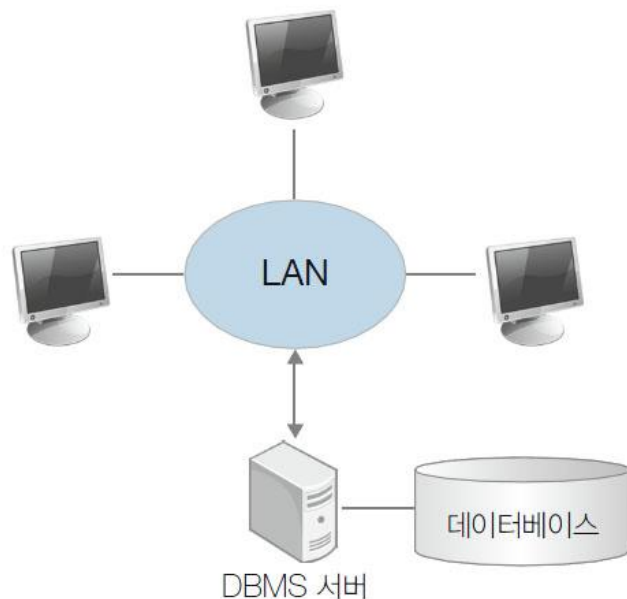


그림 1-12 데이터베이스 시스템

2. 정보 시스템의 발전

㉓ 웹 데이터베이스 시스템

- 데이터베이스를 웹 브라우저에서 사용할 수 있도록 서비스하는 시스템
- 불특정 다수 고객을 상대로 하는 온라인 상거래나 공공 민원 서비스 등에 사용됨

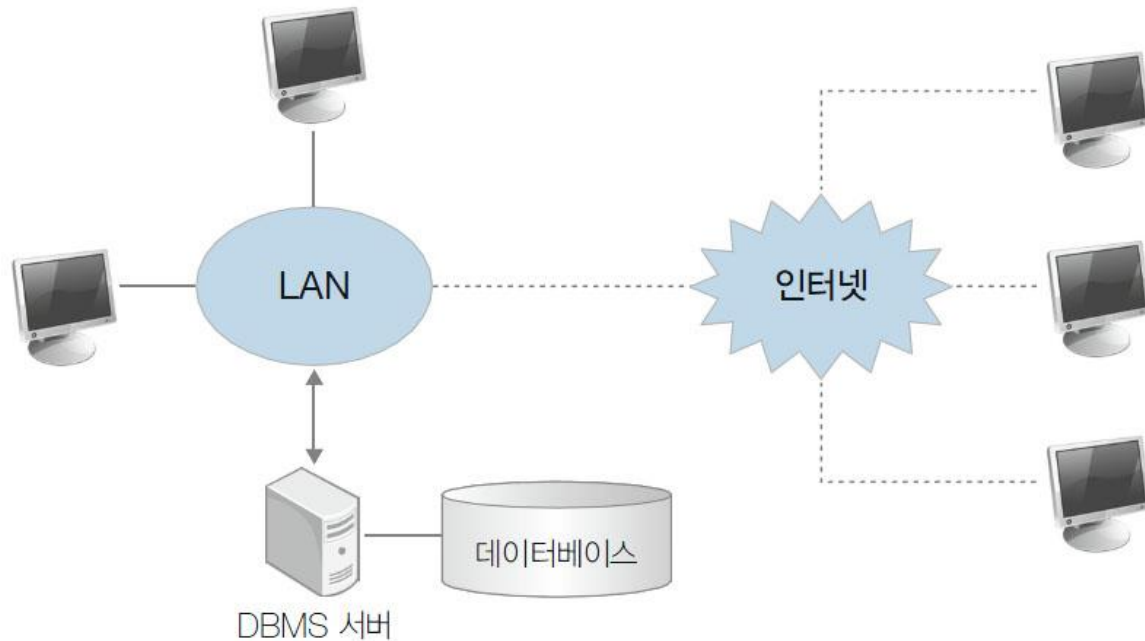


그림 1-13 웹 데이터베이스 시스템

2. 정보 시스템의 발전

④ 분산 데이터베이스 시스템

- 여러 곳에 분산된 DBMS 서버를 연결하여 운영하는 시스템
- 대규모의 응용 시스템에 사용됨

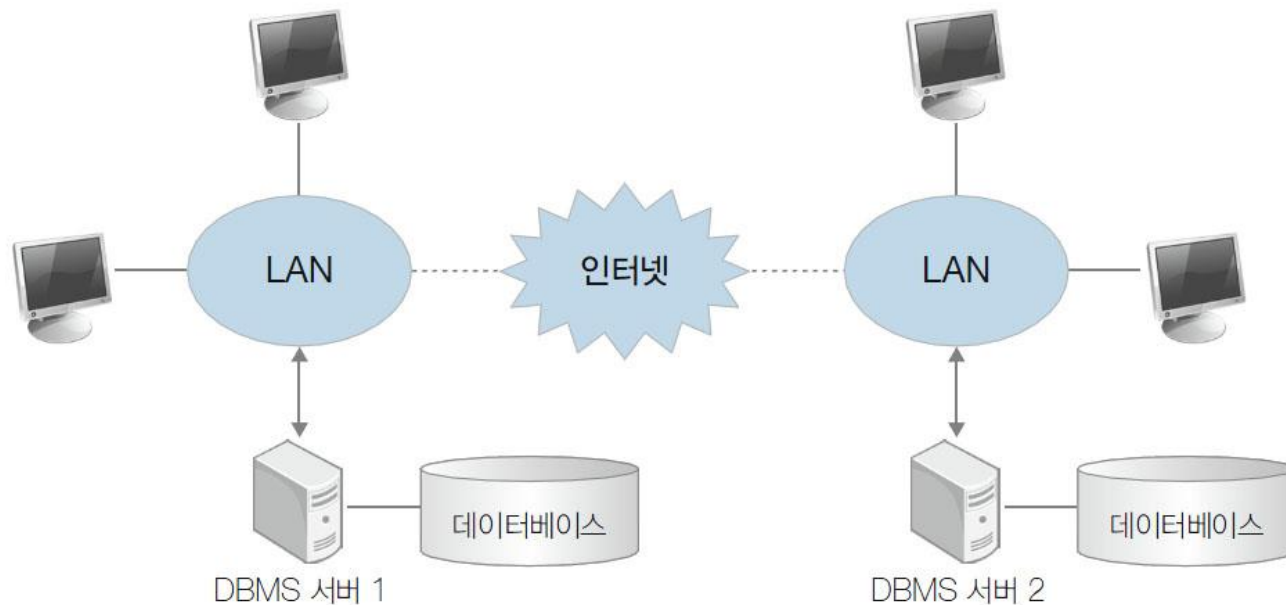


그림 1-14 분산 데이터베이스 시스템

2. 정보 시스템의 발전

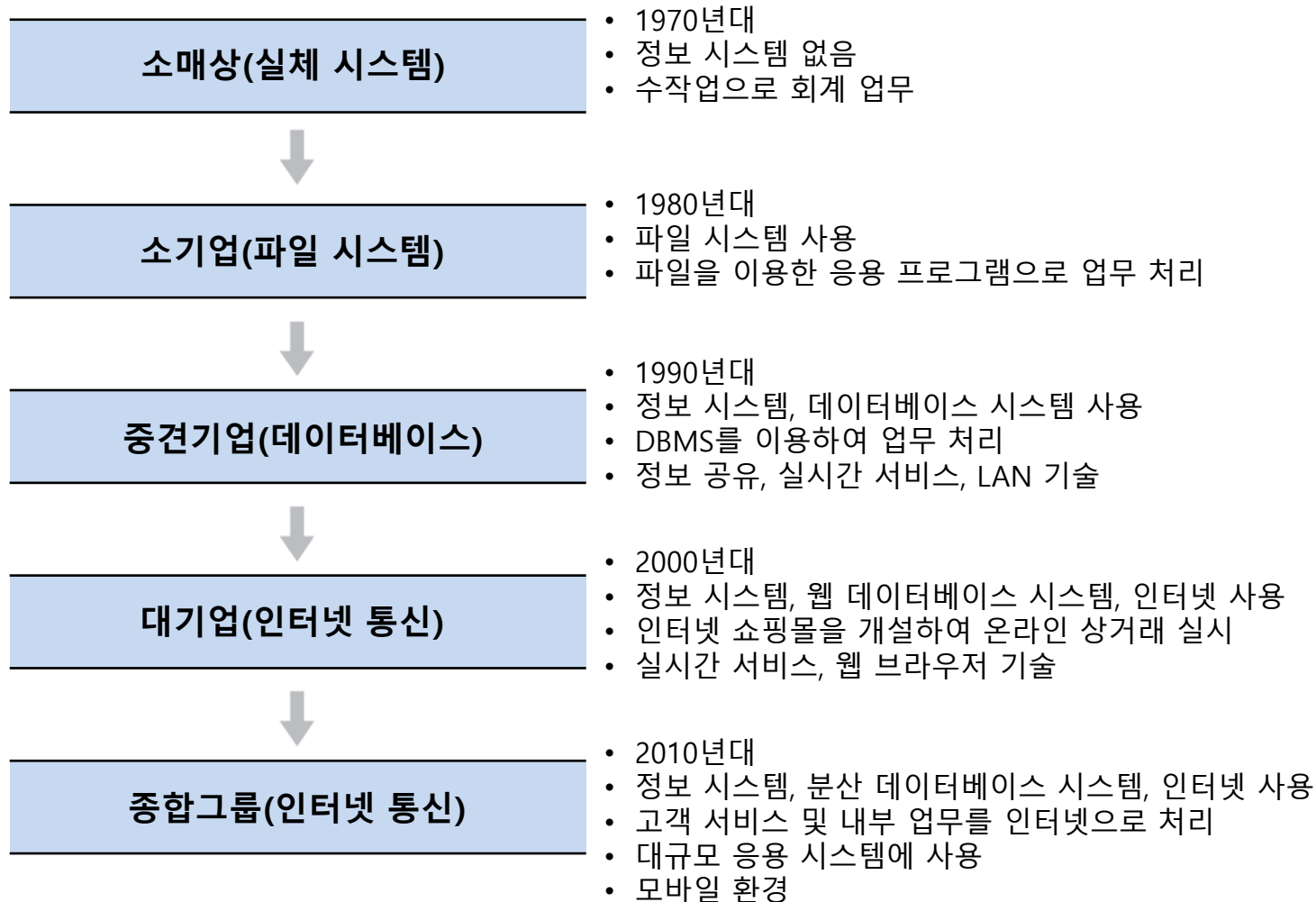


그림 1-15 정보 시스템의 발전과 기업의 업무 환경 변화

03 파일 시스템과 DBMS

1. 데이터를 저장하는 방법
2. 데이터의 저장 방법 비교
3. 파일 시스템과 DBMS의 비교



1. 데이터를 저장하는 방법

- ❶ 데이터를 프로그램 내부에 저장하는 방법
- ❷ 파일 시스템을 사용하는 방법
- ❸ DBMS를 사용하는 방법



그림 1-16 고객 서비스를 온라인 정보 서비스로 전환

1. 데이터를 저장하는 방법

❖ 데이터를 프로그램 내부에 저장하는 방법

[프로그램 1]

- C 언어의 구조체 BOOK을 먼저 선언하고 main() 프로그램에서 구조체 배열 변수 BOOKS[]에 데이터를 저장
- 도서 데이터는 프로그램 내 구조체 변수에 저장됨
- 문제점 : 새로운 데이터가 생길 때마다 프로그램을 수정한 후 다시 컴파일해야 함

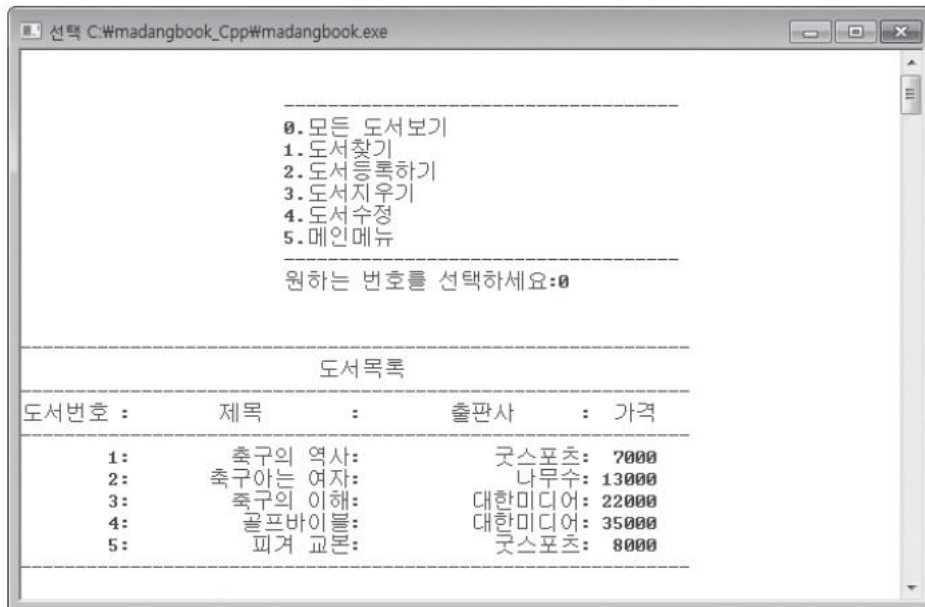


그림 1-17 도서 검색 프로그램

1. 데이터를 저장하는 방법

[프로그램 1] 소스코드

```
/* BOOK 데이터 구조 정의 */
typedef struct
{
    int  bookid[5];
    char bookname[20];
    char publisher[20];
    int  price;
} BOOK;

int main() {
    BOOK BOOKS[10];

    /* 구조체 배열 변수에 데이터 저장 */
    /* 첫 번째 도서 저장 */
    BOOKS[1].bookid=1;
    strcpy(BOOKS[1].bookname, "축구의 역사");
    strcpy(BOOKS[1].publisher, "굿스포츠");
    BOOKS[1].price=7000;
    /* 두 번째 도서 저장 */
    BOOKS[2].bookid=2;
    strcpy(BOOKS[2].bookname, "축구 아는 여자");
    strcpy(BOOKS[2].publisher, "나무수");
    BOOKS[2].price=13000;

    /* 나머지 다른 도서 저장(생략) */
    ....

    /* 모든 도서보기 프로그램 호출 */
    search_all();
    ....
}
```

1. 데이터를 저장하는 방법

❖ 파일 시스템을 사용하는 방법

[프로그램 2]

- BOOK 데이터 구조를 먼저 선언하고 main() 프로그램에서 파일로부터 데이터를 불러와 구조체 배열 변수 BOOKS[]에 저장
- 새로운 데이터가 추가되어도 프로그램을 수정할 필요 없음
- 문제점 : 같은 파일을 두 개의 프로그램이 공유하는 것이 운영체제의 도움 없이 불가능

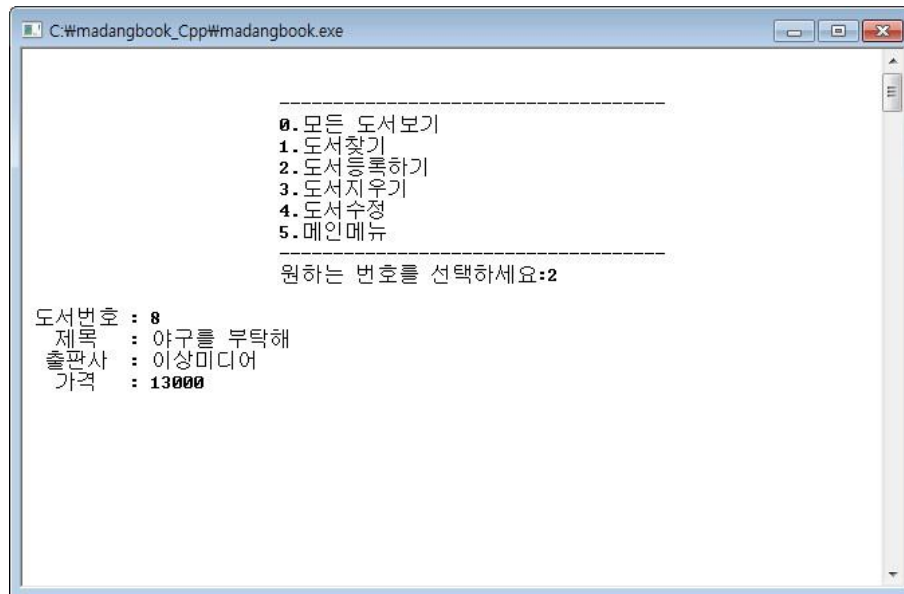


그림 1-18 도서 검색 프로그램에서 도서를 등록하는 화면

1. 데이터를 저장하는 방법

[프로그램 2] 소스코드

```
/* BOOK 데이터 구조 정의 */
typedef struct
{
    int  bookid[5];
    char bookname[20];
    char publisher[20];
    int  price;
} BOOK;

int main( )
{
    BOOK BOOKS[10];
    int i=1;
    /* 도서 입력 함수 */
    insert( );
    /* 파일에 저장된 데이터를 배열 BOOKS[ ]에 저장 */
    fp=fopen("book.dat","rb");
    bp=(BOOK *)calloc(1,sizeof(BOOK));
    /* 파일에서 책을 읽는다 */
    while(fread(bp,sizeof(BOOK),1,fp) != 0)
    {
        BOOKS[i].bookid =bp->bookid;
        strcpy(BOOKS[i].bookname, bp ->bookname);
        strcpy(BOOKS[i].publisher, bp ->publisher);
        BOOKS[i].price =bp ->price;
        i++;
    }
    /* 모든 도서보기 프로그램 호출 */
    search_all( );

    .....
}
```


1. 데이터를 저장하는 방법

❖ DBMS를 사용하는 방법

[프로그램 3]

- 데이터 정의와 데이터 값을 DBMS가 관리
- DBMS는 데이터 정의, 데이터 변경 등의 작업을 할 수 있는 별도의 프로그램을 갖고 있음
- 프로그램에 데이터 정의나 데이터 값을 포함하지 않기 때문에 데이터 구조가 바뀌어도 다시 컴파일할 필요가 없음

1. 데이터를 저장하는 방법

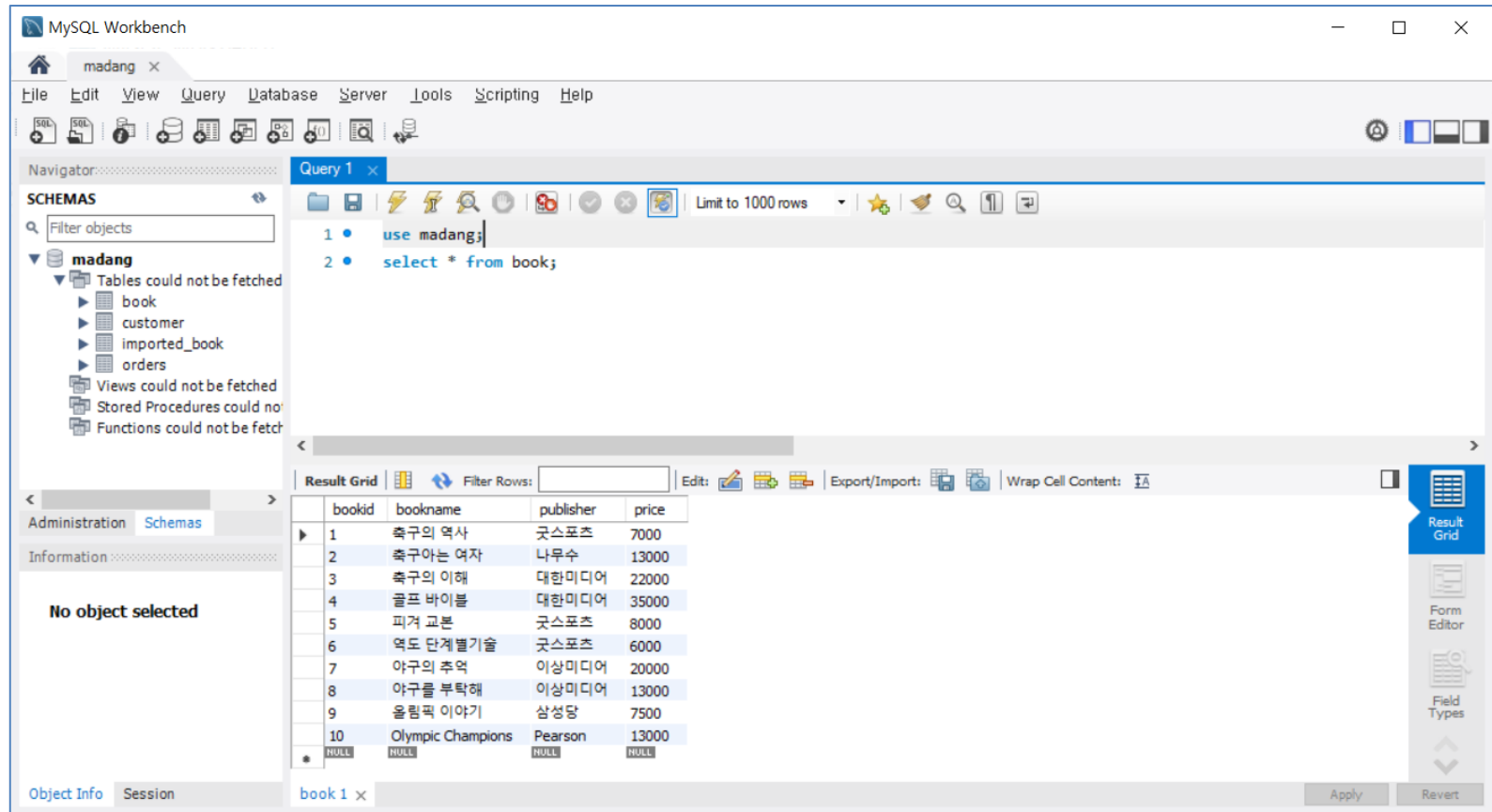


그림 1-19 MySQL Workbench의 데이터베이스 관리 화면

1. 데이터를 저장하는 방법

[프로그램 3] 소스코드

```
int main( )
{
    /* 반환된 행의 수 */
    int num_ret;

    /* DBMS에 접속 */
    EXEC SQL CONNECT :username IDENTIFIED BY :password;

    /* SQL 문 실행 */
    EXEC SQL DECLARE c1 CURSOR FOR
        SELECT bookname, publisher, price FROM BOOK;
    EXEC SQL OPEN c1;



    /* 모든 도서보기 프로그램 호출 */
    search_all( );

    /* SQL 문 실행 결과 출력 */
    for (;;) {
        EXEC SQL FETCH c1 INTO :BOOK_rec;
        print_rows(num_ret);
    }
    EXEC SQL CLOSE c1;

    /* 접속 해제 */
    EXEC SQL COMMIT WORK RELEASE;
}
```

2. 데이터의 저장 방법 비교

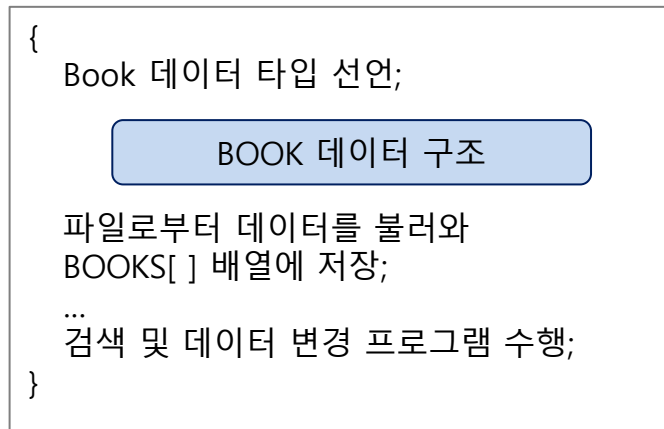
[프로그램 1] 구조

```
{  
    Book 데이터 타입 선언;  
      
  
    프로그램 내에서  
        BOOKS[ ] 배열에 데이터 저장;  
      
  
    ...  
    검색 및 데이터 변경 프로그램 수행;  
}
```

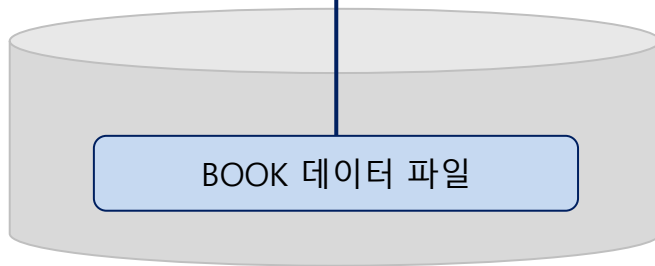
- 프로그램에 데이터 정의와 데이터 값을 모두 포함하는 방식
- 프로그램에 BOOK 데이터 구조를 정의하고 데이터 값도 직접 변수에 저장함
- 데이터 구조 혹은 데이터 값이 바뀌면 프로그램을 다시 컴파일해야 함

2. 데이터의 저장 방법 비교

[프로그램 2] 구조



- 파일에 데이터 값, 프로그램에 데이터 정의를 포함하는 방식
- 프로그램에 BOOK 데이터 구조만 정의하고, 데이터 값은 book.dat라는 파일에 저장됨
- 데이터 값이 바뀌면 프로그램에 변경이 없지만, 데이터 구조가 바뀌면 프로그램을 다시 컴파일해야 함

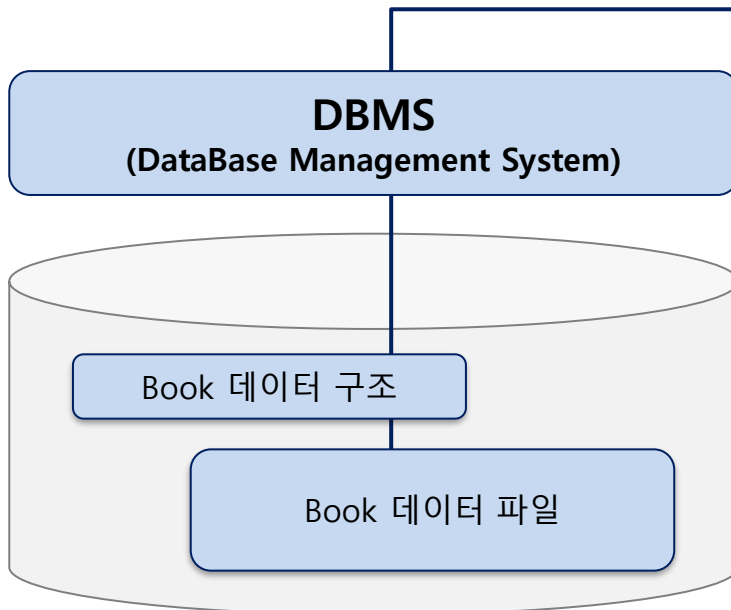


2. 데이터의 저장 방법 비교

[프로그램 3] 구조

```
{  
  /* BOOK 데이터 타입 선언 필요 없음 */  
  
  SQL 문을 실행하여 결과를 가져옴;  
  ...  
  SQL 문으로 데이터 변경 ;  
}
```

- DBMS가 데이터 정의와 데이터 값을 관리하는 방식
- BOOK 데이터 구조는 DBMS가 관리하고, 데이터 값은 데이터베이스에 저장됨
- 데이터 값이 바뀌거나 데이터 값이 바뀌어도 프로그램을 다시 컴파일할 필요 없음



3. 파일 시스템과 DBMS의 비교

표 1-5 파일 시스템과 DBMS의 비교

구분	파일 시스템	DBMS
데이터 정의 및 저장	데이터 정의 : 응용 프로그램 데이터 저장 : 파일 시스템	데이터 정의 : DBMS 데이터 저장 : 데이터베이스
데이터 접근 방법	응용 프로그램이 파일에 직접 접근	응용 프로그램이 DBMS에 파일 접근을 요청
사용 언어	자바, C++, C 등	자바, C++, C 등과 SQL
CPU/주기억장치 사용	적음	많음

3. 파일 시스템과 DBMS의 비교

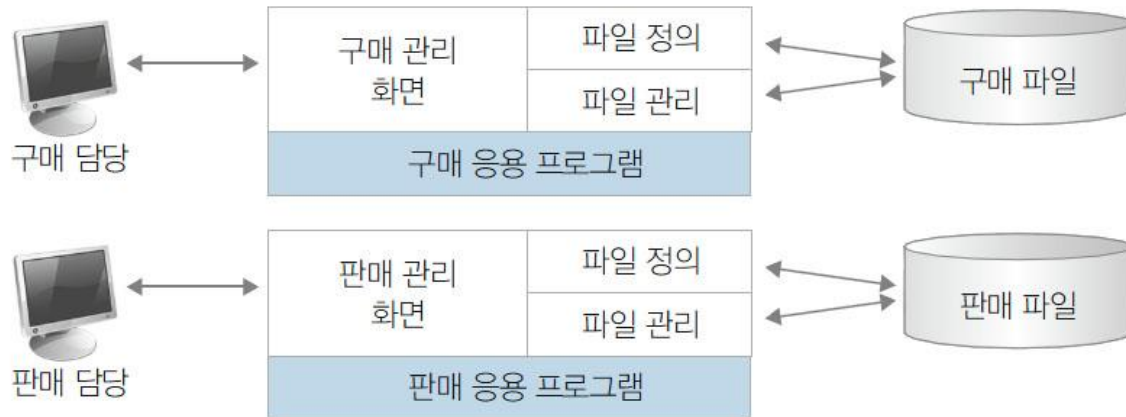


그림 1-20 파일 시스템으로 구축된 구매 및 판매 응용 프로그램

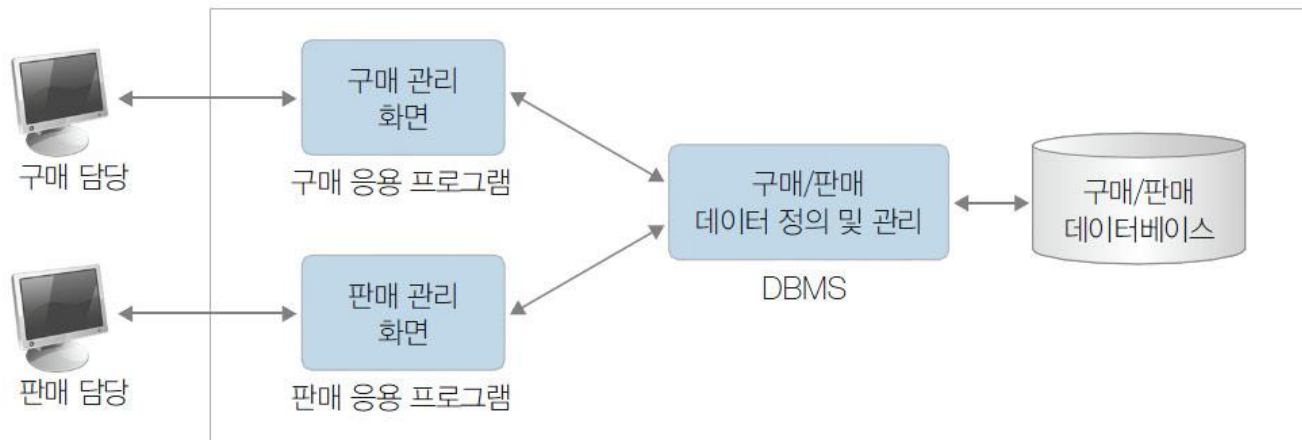


그림 1-21 DBMS로 구축된 구매 및 판매 응용 프로그램

3. 파일 시스템과 DBMS의 비교

표 1-6 DBMS의 장점

구분	파일 시스템	DBMS
데이터 중복	데이터를 파일 단위로 저장하므로 중복 가능	DBMS를 이용하여 데이터를 공유하기 때문에 중복 가능성 낮음
데이터 일관성	데이터의 중복 저장으로 일관성이 결여됨	중복 제거로 데이터의 일관성이 유지됨
데이터 독립성	데이터 정의와 프로그램의 독립성 유지 불가능	데이터 정의와 프로그램의 독립성 유지 가능
관리 기능	보통	데이터 복구, 보안, 동시성 제어, 데이터 관리 기능 등을 수행
프로그램 개발 생산성	나쁨	짧은 시간에 큰 프로그램을 개발할 수 있음
기타 장점	별도의 소프트웨어 설치가 필요없음 (운영체제가 지원)	데이터 무결성 유지, 데이터 표준 준수 용이

04. 데이터베이스 시스템의 구성

1. 데이터베이스 언어
2. 데이터베이스 사용자
3. DBMS
4. 데이터 모델
5. 데이터베이스의 개념적 구조



<데이터베이스 시스템의 구성>

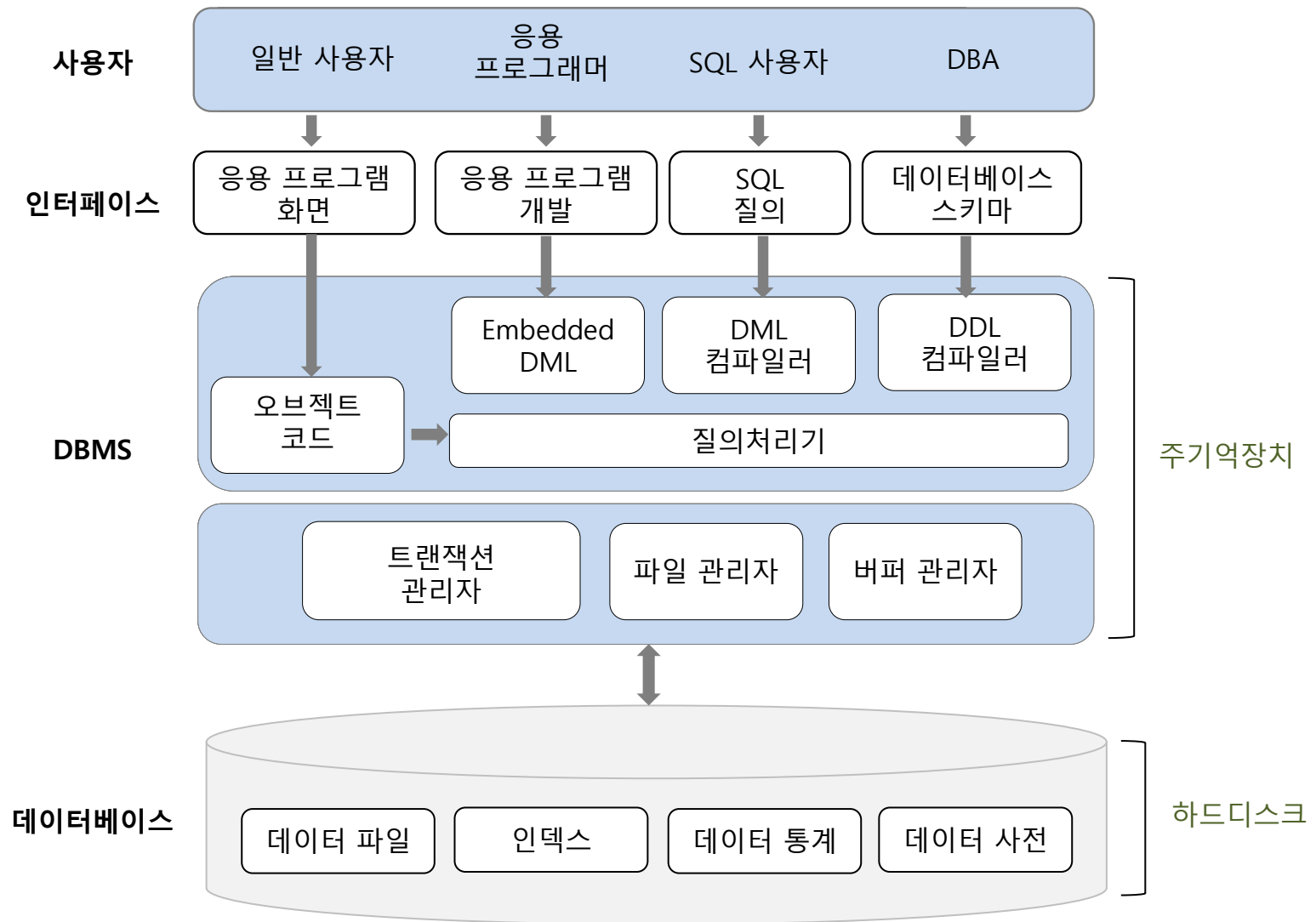


그림 1-22 데이터베이스 시스템의 구성

1. 데이터베이스 언어

■ SQL(Structured Query Language)

- 데이터베이스 생성, 변경, 삭제와 레코드 삽입, 조회, 삭제, 수정 그리고 데이터베이스의 백업, 롤백, 보안, 관리 등에 의한 처리에 사용되는 표준 언어
 - 새로운 데이터베이스를 생성할 수 있고, 삭제할 수 있다.
 - 데이터베이스에 자료를 삽입, 삭제할 수 있다.
 - 데이터베이스 안에 새로운 테이블을 생성하고 삭제할 수 있다.
 - 데이터베이스에서 자료를 검색, 추출할 수 있다.
- SQL은 관계형 데이터베이스와 비관계형 데이터베이스(NoSQL)로 나뉘는데, 주로 관계형 데이터베이스를 다룬다
- 종류
 - 데이터 정의어(**DDL**, Data Definition Language)
 - 데이터 조작어(**DML**, Data Manipulation Language)
 - 데이터 제어어(**DCL**, Data Control Language)

1. 데이터베이스 언어

질의 1-1 Book 테이블에서 모든 도서이름(bookname)과 출판사(publisher)를 검색하시오.

```
SELECT bookname, publisher  
FROM Book;
```

Book 테이블

bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000

bookname	publisher
축구의 역사	굿스포츠
축구아는 여자	나무수
축구의 이해	대한미디어
골프 바이블	대한미디어
피겨 교본	굿스포츠

1. 데이터베이스 언어

질의 1-2 가격(price)이 10,000원 이상인 도서이름(bookname)과 출판사(publisher)를 검색하시오.

```
SELECT  bookname, publisher
FROM    Book
Where   price >= 10000;
```

Book 테이블

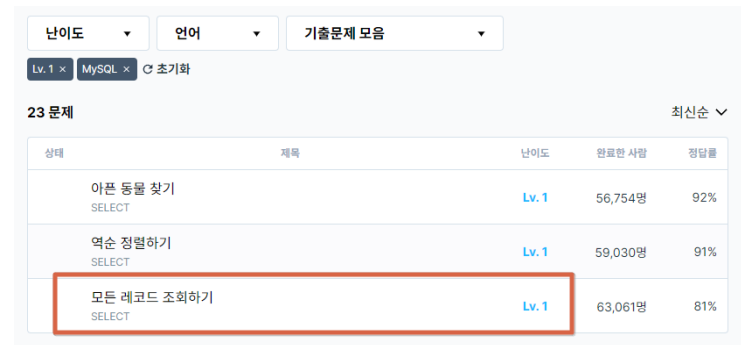
bookid	bookname	publisher	price
1	축구의 역사	굿스포츠	7000
2	축구아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000

bookname	publisher
축구아는 여자	나무수
축구의 이해	대한미디어
골프 바이블	대한미디어

1. 데이터베이스 언어

■ 프로그래머스

- 프로그래머 사이트(<http://programmers.co.kr>) 회원 가입 후 아래 문제를 해결해봅시다
- Level 1
 - 모든 레코드 조회하기
 - 역순 정렬하기
 - 동물의 아이디와 이름



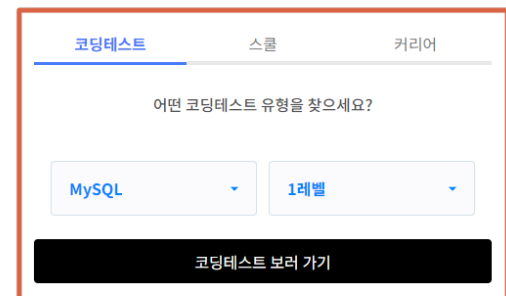
The screenshot shows the 'All records' problem highlighted in a red box. The table lists problems with their status, titles, difficulty levels, number of users, and completion rates.

상태	제목	난이도	완료한 사람	정답률
아픈 동물 찾기 SELECT		Lv. 1	56,754명	92%
역순 정렬하기 SELECT		Lv. 1	59,030명	91%
모든 레코드 조회하기 SELECT		Lv. 1	63,061명	81%

[스쿨](#)[커리어](#)[커뮤니티](#)[스킬체크](#)[코딩테스트 연습](#)[코딩역량인증시험](#)[로그인](#) · [회원가입](#)[기업서비스](#)

성장하는 개발자를 위한 베이스캠프

개발자 맞춤
큐레이션을 찾아보세요.



The screenshot shows the '코딩테스트' (Coding Test) section with filters for 'MySQL' and '1레벨' (Level 1). The text '어떤 코딩테스트 유형을 찾으세요?' (Which coding test type are you looking for?) is visible above the filters.

코딩테스트 스쿨 커리어

어떤 코딩테스트 유형을 찾으세요?

MySQL 1레벨

코딩테스트 보러 가기

2. 데이터베이스 사용자

■ 일반사용자

- 은행의 창구 혹은 관공서의 민원 접수처 등에서 데이터를 다루는 업무를 하는 사람
- 프로그래머가 개발한 프로그램을 이용하여 데이터베이스에 접근 일반인

■ 응용프로그래머

- 일반 사용자가 사용할 수 있도록 프로그램을 만드는 사람
- 자바, C, JSP 등 프로그래밍 언어와 SQL을 사용하여 일반 사용자를 위한 사용자 인터페이스와 데이터를 관리하는 응용 로직을 개발

■ SQL 사용자

- SQL을 사용하여 업무를 처리하는 IT 부서의 담당자
- 응용 프로그램으로 구현되어 있지 않은 업무를 SQL을 사용하여 처리

■ 데이터베이스 관리자(DBA, Database Administrator)

- 데이터베이스 운영 조직의 데이터베이스 시스템을 총괄하는 사람
- 데이터 설계, 구현, 유지보수의 전 과정을 담당
- 데이터베이스 사용자 통제, 보안, 성능 모니터링, 데이터 전체 파악 및 관리, 데이터 이동 및 복사 등 제반 업무를 함

2. 데이터베이스 사용자

표 1-7 데이터베이스 사용자 별로 갖추어야 할 지식 수준(× : 없음, ○ : 보통, ◎ : 높음)

	SQL 언어	프로그래밍 능력	DBMS 지식	데이터 구성
일반 사용자	×	×	×	×
SQL 사용자	◎	×	○	○
응용 프로그래머	◎	◎	○	○
데이터베이스 관리자	◎	○	◎	◎

3. DBMS

표 1-8 DBMS의 기능

데이터 정의(Definition)	데이터의 구조를 정의하고 데이터 구조에 대한 삭제 및 변경 기능을 수행함
데이터 조작(manipulation)	데이터를 조작하는 소프트웨어(응용 프로그램)가 요청하는 데이터의 삽입, 수정, 삭제 작업을 지원함
데이터 추출(Retrieval)	사용자가 조회하는 데이터 혹은 응용 프로그램의 데이터를 추출함
데이터 제어(Control)	데이터베이스 사용자를 생성하고 모니터링하며 접근을 제어함. 백업과 회복, 동시성 제어 등의 기능을 지원함

3. DBMS

표 1-8 DBMS의 제품

DBMS	특징
MySQL	오픈소스와 상용으로 구분
MariaDB	오픈소스, MySQL과 유사, 초기 개발자가 만듦
PostgreSQL	오픈소스, Ubuntu Linux에 기본 탑재
Oracle	주로 상용, 시장 점유율 1위
SQL Server	MS-Windows서버용, 중대형
Access	MS-Windows PC용
SQLite	메모리 기반, 모바일, 오픈 소스(무료)

4. 데이터 모델

- 계층 데이터 모델(hierarchical data model)
- 네트워크 데이터 모델(network data model)
- 객체 데이터 모델(object data model)
- 관계 데이터 모델(relational data model) → 가장 많이 쓰인다
- 객체-관계 데이터 모델(object-relational data model) → 관계 데이터 모델과 객체 데이터 모델의 장점을 결합한 모델

4. 데이터 모델

❶ 포인터 사용 : 계층 데이터 모델, 네트워크 데이터 모델

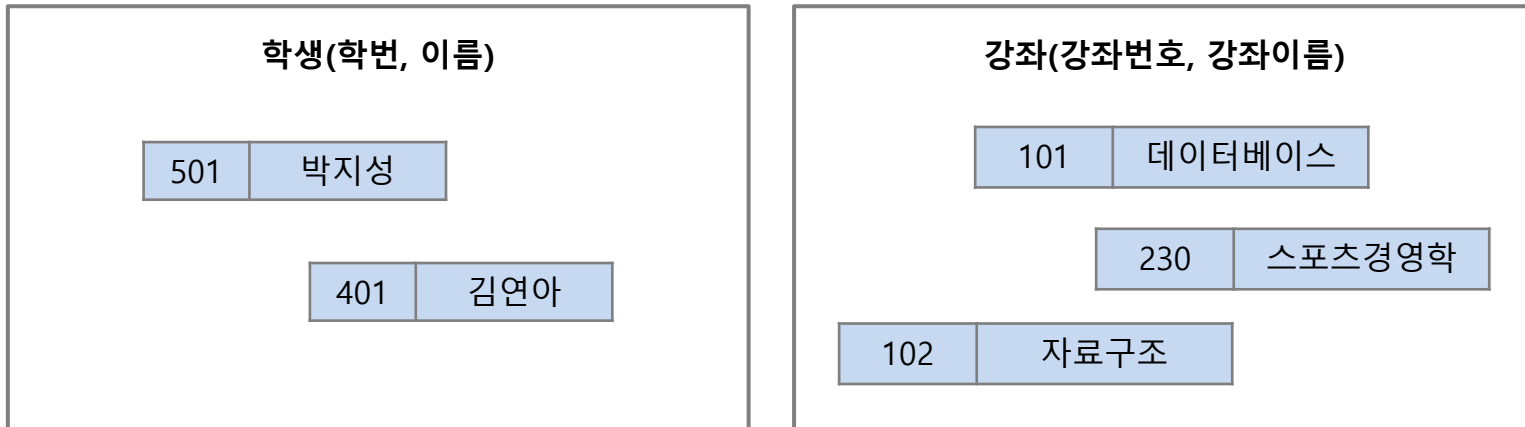


그림 1-23 관계 표현을 위한 예시

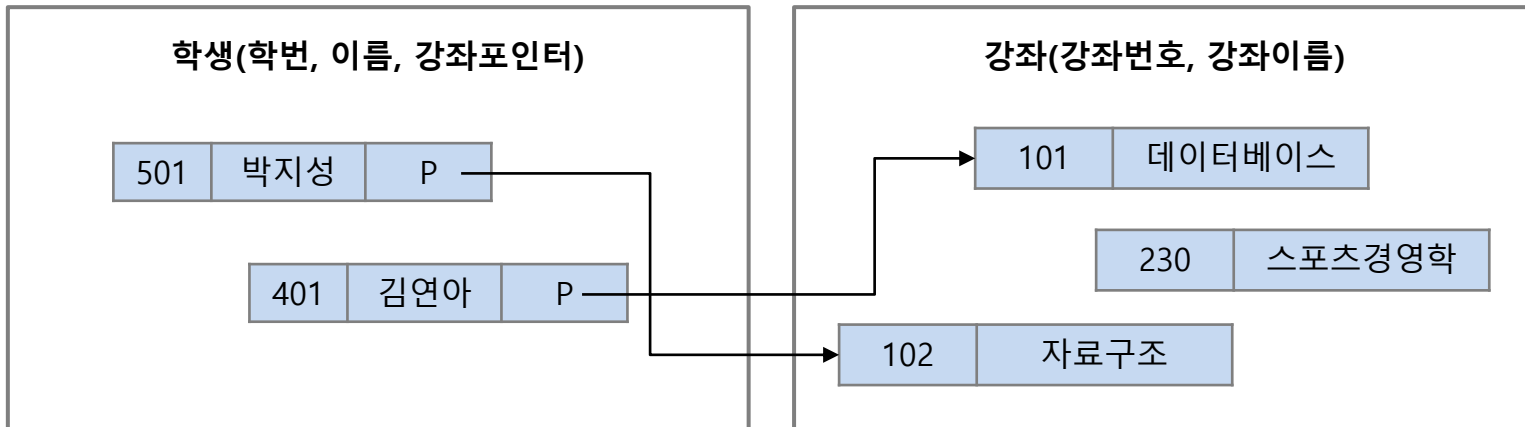


그림 1-24 포인터를 사용하여 관계 표현

4. 데이터 모델

㉓ 속성 값 사용 : 관계 데이터 모델

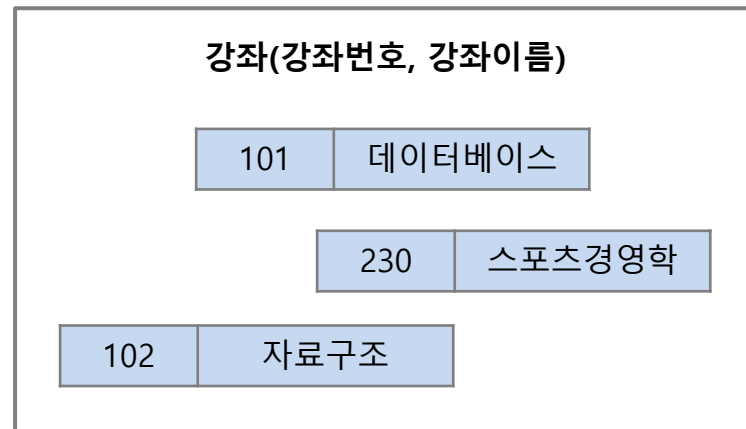
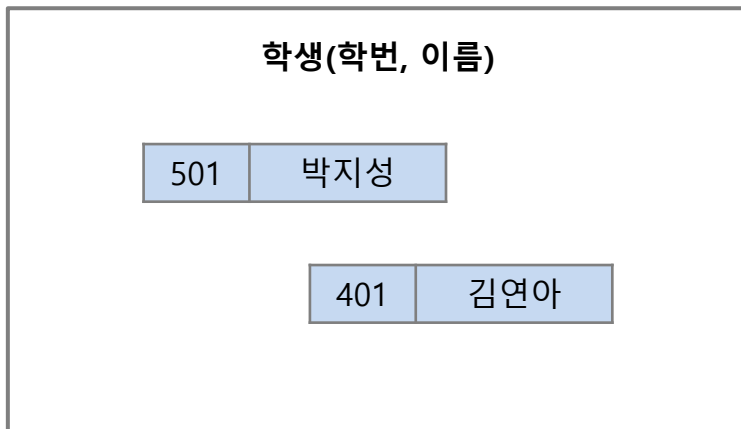


그림 1-23 관계 표현을 위한 예시

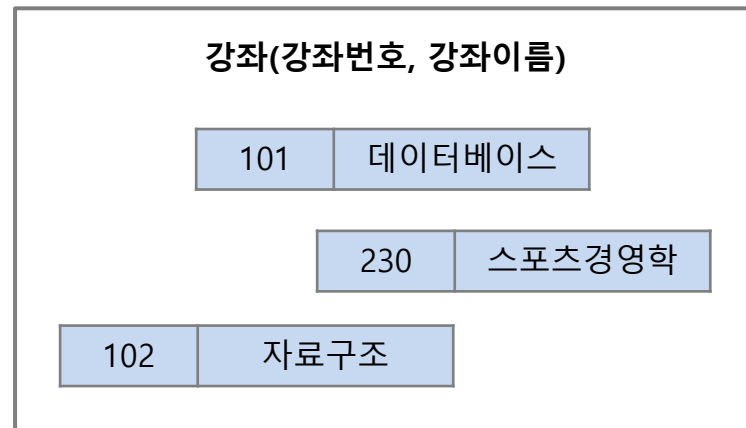
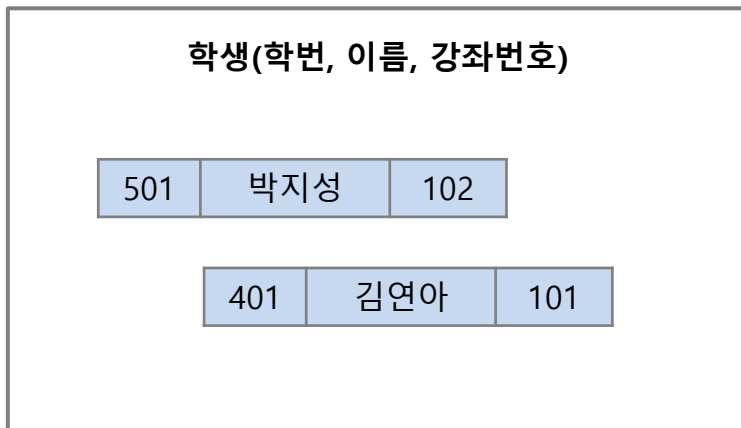


그림 1-25 속성 값을 사용하여 관계 표현

4. 데이터 모델

㉓ 객체식별자 사용 : 객체 데이터 모델

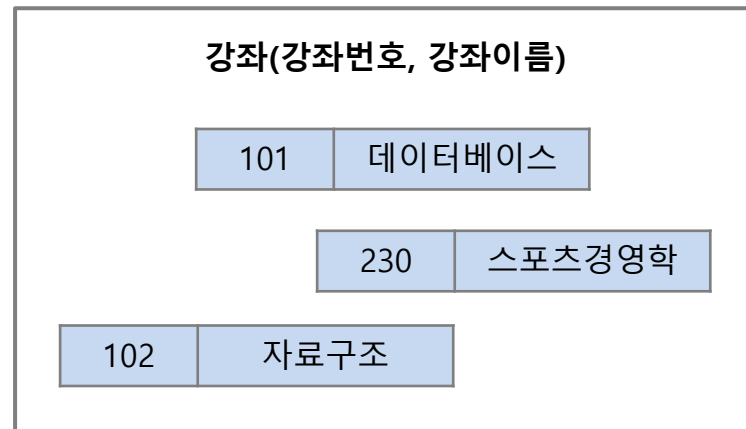
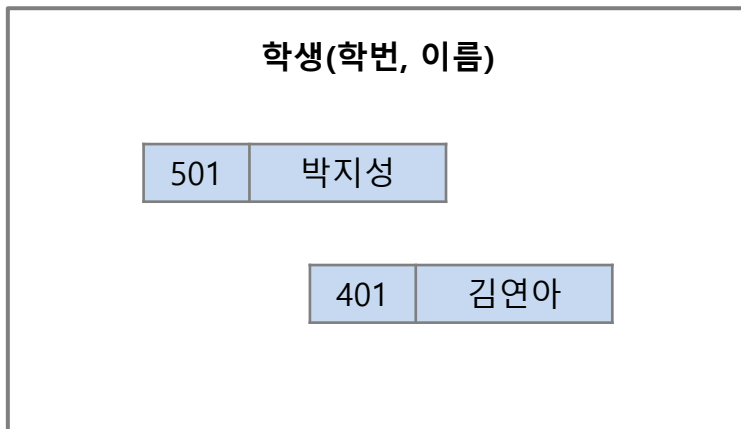


그림 1-23 관계 표현을 위한 예시

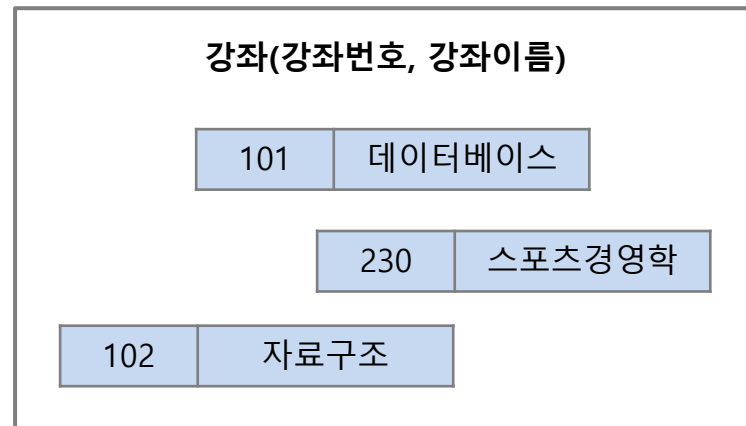
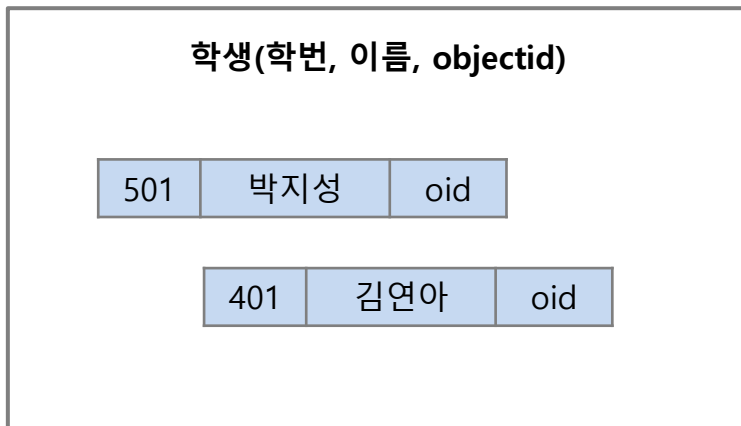


그림 1-26 객체식별자를 사용하여 관계 표현


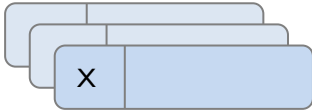
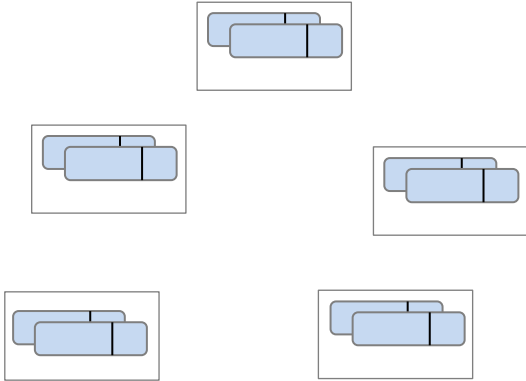
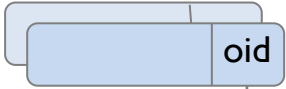

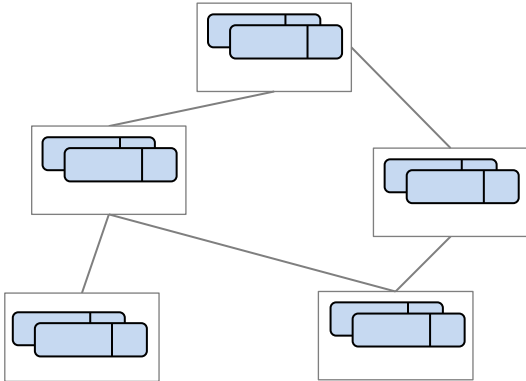
4. 데이터 모델

표 1-9 데이터 모델과 각 모델에서 관계의 표현 방법

데이터 모델	관계의 표현	데이터 구성
계층 데이터 모델 (포인터 사용)	<div> <div>학생</div> <div> <div></div> <div></div> <div>P</div> </div> <div>강좌</div> <div> <div></div> <div></div> <div></div> <div></div> </div> </div>	
네트워크 데이터 모델 (포인터 사용)	<div> <div>학생</div> <div> <div></div> <div></div> <div>P</div> </div> <div>강좌</div> <div> <div></div> <div></div> <div></div> <div></div> </div> </div>	

4. 데이터 모델

표 1-9 데이터 모델과 각 모델에서 관계의 표현 방법

데이터 모델	관계의 표현	데이터 구성
관계 데이터 모델 (속성 값 사용)	<p>학생</p>  <p>강좌</p> 	
객체 데이터 모델 (객체식별자 사용)	<p>학생</p>  <p>강좌</p>  <p>객체 번호 oid</p>	

4. 데이터 모델

표 1-10 데이터 모델의 역사

데이터 모델	1960 년대	1970 년대	1980 년대	1990 년대	2000 년대	2010 년대
제품 종류						
계층 데이터 모델 IMS(IBM)	⇒	⇒	⇒			
네트워크 데이터 모델 IDS(GE)	⇒	⇒	⇒			
관계 데이터 모델 Oracle(Oracle), System R(IBM)			⇒	⇒	⇒	⇒
객체 데이터 모델 GemStone, ObejectStore				⇒	⇒	
객체-관계 데이터 모델 UniSQL					⇒	⇒

- 위 표에는 해당 데이터 모델이 주로 사용되던 시기를 표시한 것이다.
- 계층 데이터 모델과 네트워크 데이터 모델은 1960년대에, 관계 데이터 모델은 1970년대에 처음 사용되기 시작하였다.

5. 데이터베이스의 개념적 구조

❖ 3단계 데이터베이스 구조

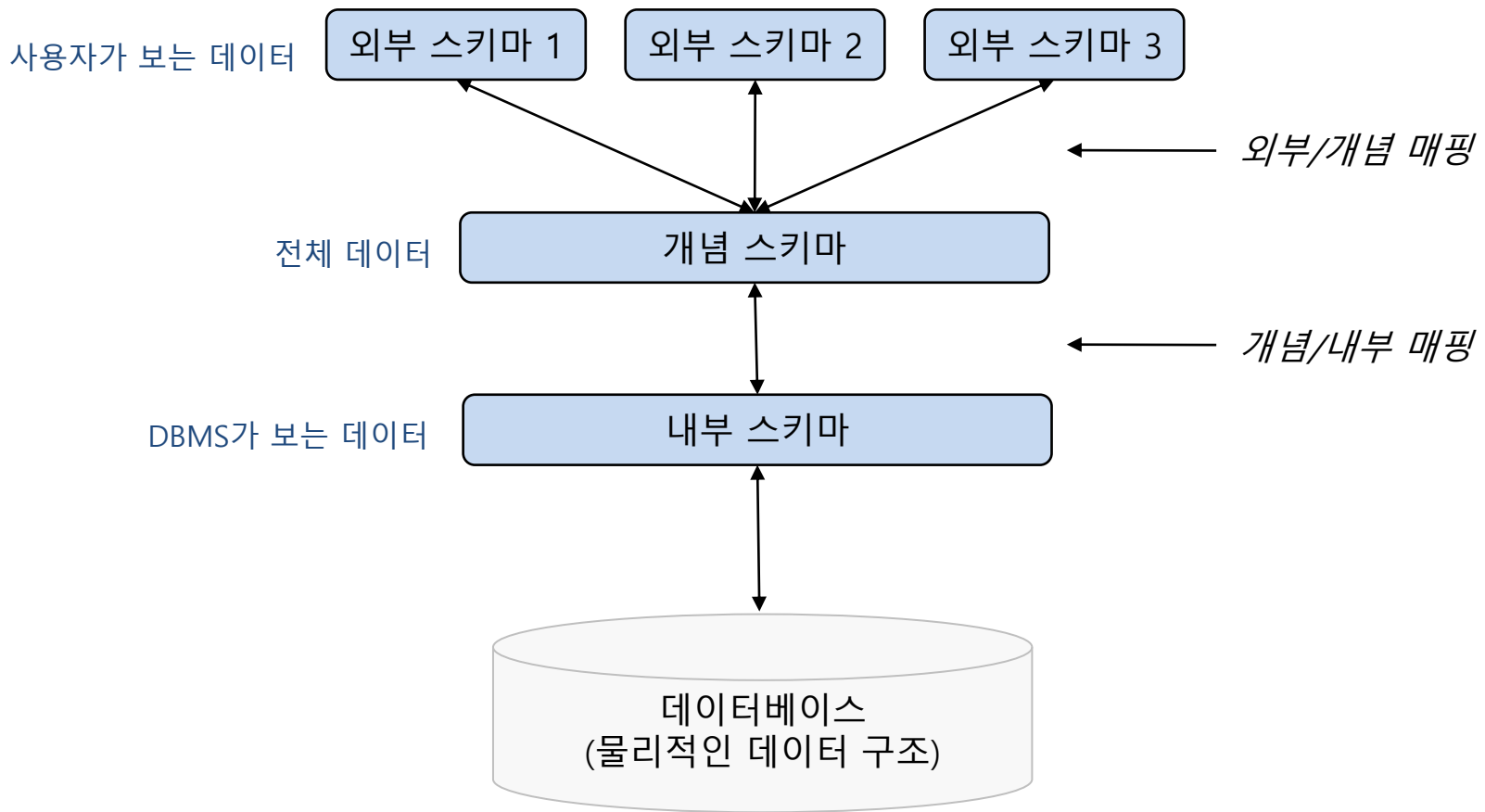


그림 1-27 ANSI의 3단계 데이터베이스 구조

5. 데이터베이스의 개념적 구조

■ 외부 단계

- 일반 사용자나 응용 프로그래머가 접근하는 계층으로 전체 데이터베이스 중에서 하나의 논리적인 부분을 의미
- 여러 개의 외부 스키마(external schema)가 있을 수 있음
- 서브 스키마(sub schema)라고도 하며, 뷰(view)의 개념임

■ 개념 단계

- 전체 데이터베이스의 정의를 의미
- 통합 조직별로 하나만 존재하며 DBA가 관리함
- 하나의 데이터베이스에는 하나의 개념 스키마(conceptual schema)가 있음

■ 내부 스키마

- 물리적 저장 장치에 데이터베이스가 실제로 저장되는 방법의 표현
- 내부 스키마(intenal schema)는 하나
- 인덱스, 데이터 레코드의 배치 방법, 데이터 압축 등에 관한 사항이 포함됨

5. 데이터베이스의 개념적 구조

■ 외부/개념 매핑

- 사용자의 외부 스키마와 개념 스키마 간의 매핑(사상)
- 외부 스키마의 데이터가 개념 스키마의 어느 부분에 해당되는지 대응시킴

■ 개념/내부 매핑

- 개념 스키마의 데이터가 내부 스키마의 물리적 장치 어디에 어떤 방법으로 저장되는지 대응시킴

5. 데이터베이스의 개념적 구조

❖ 데이터베이스 구조의 예 : 수강신청 데이터베이스 구조

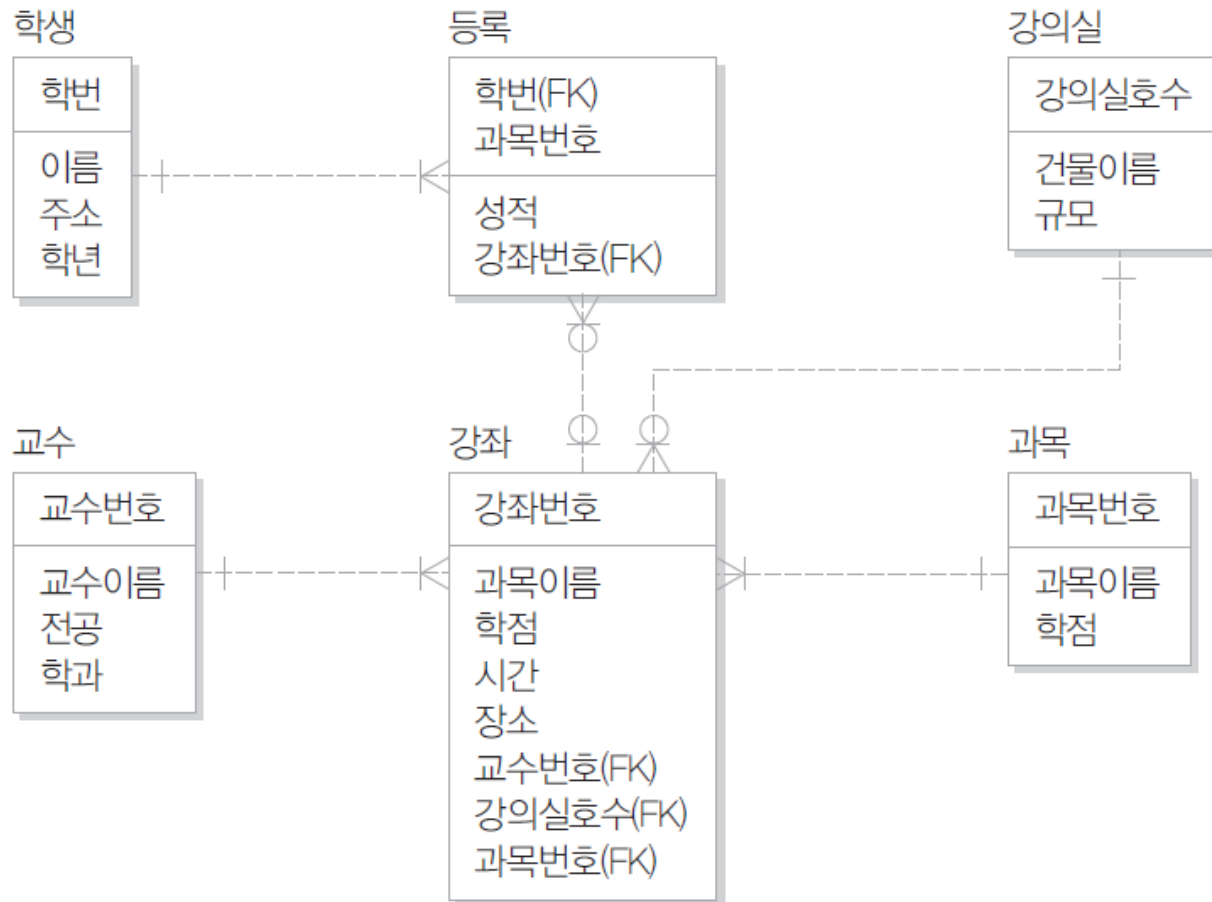


그림 1-28 수강신청 데이터베이스의 개념 스키마

5. 데이터베이스의 개념적 구조



그림 1-29 수강등록 담당 부서에서 필요한 데이터베이스(외부 스키마1)

5. 데이터베이스의 개념적 구조

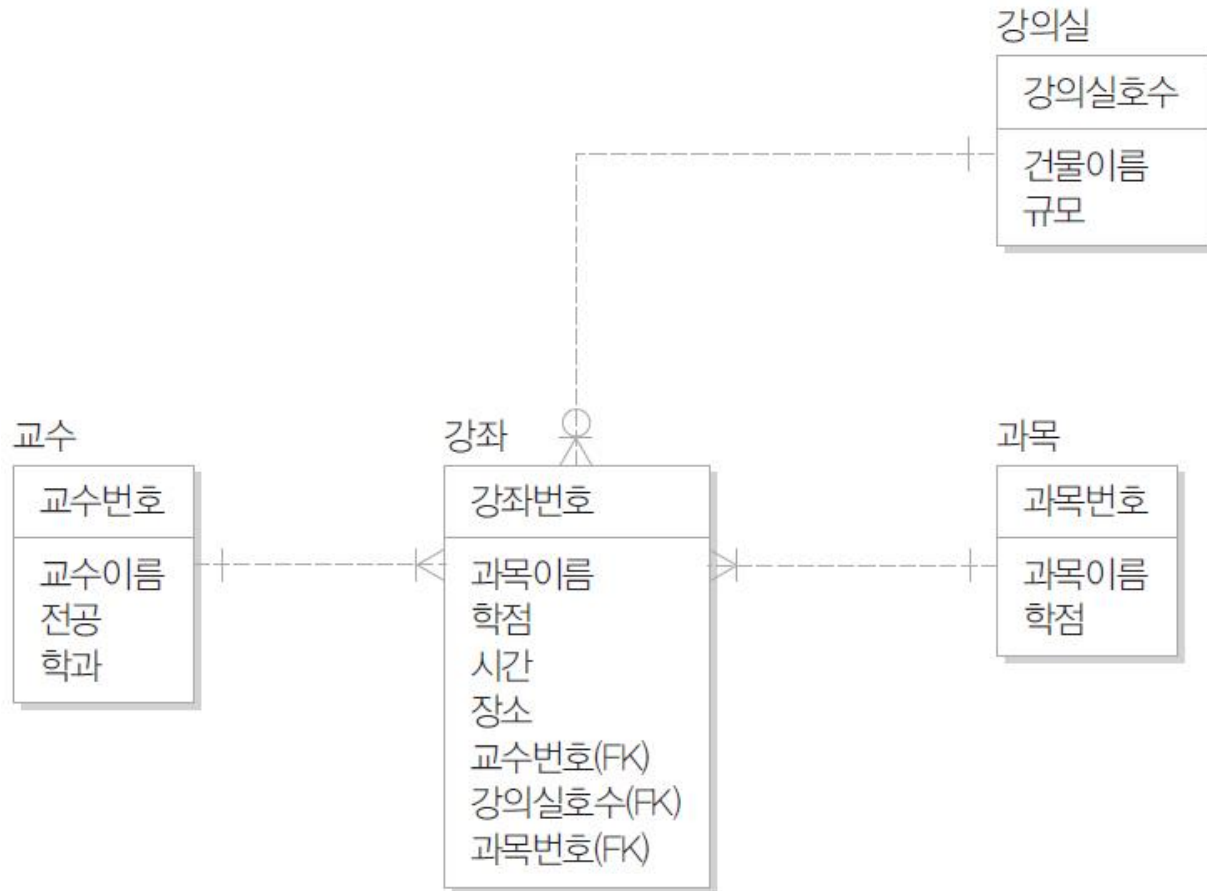


그림 1-30 시간표 담당 부서에서 필요한 데이터베이스(외부 스키마2)

5. 데이터베이스의 개념적 구조

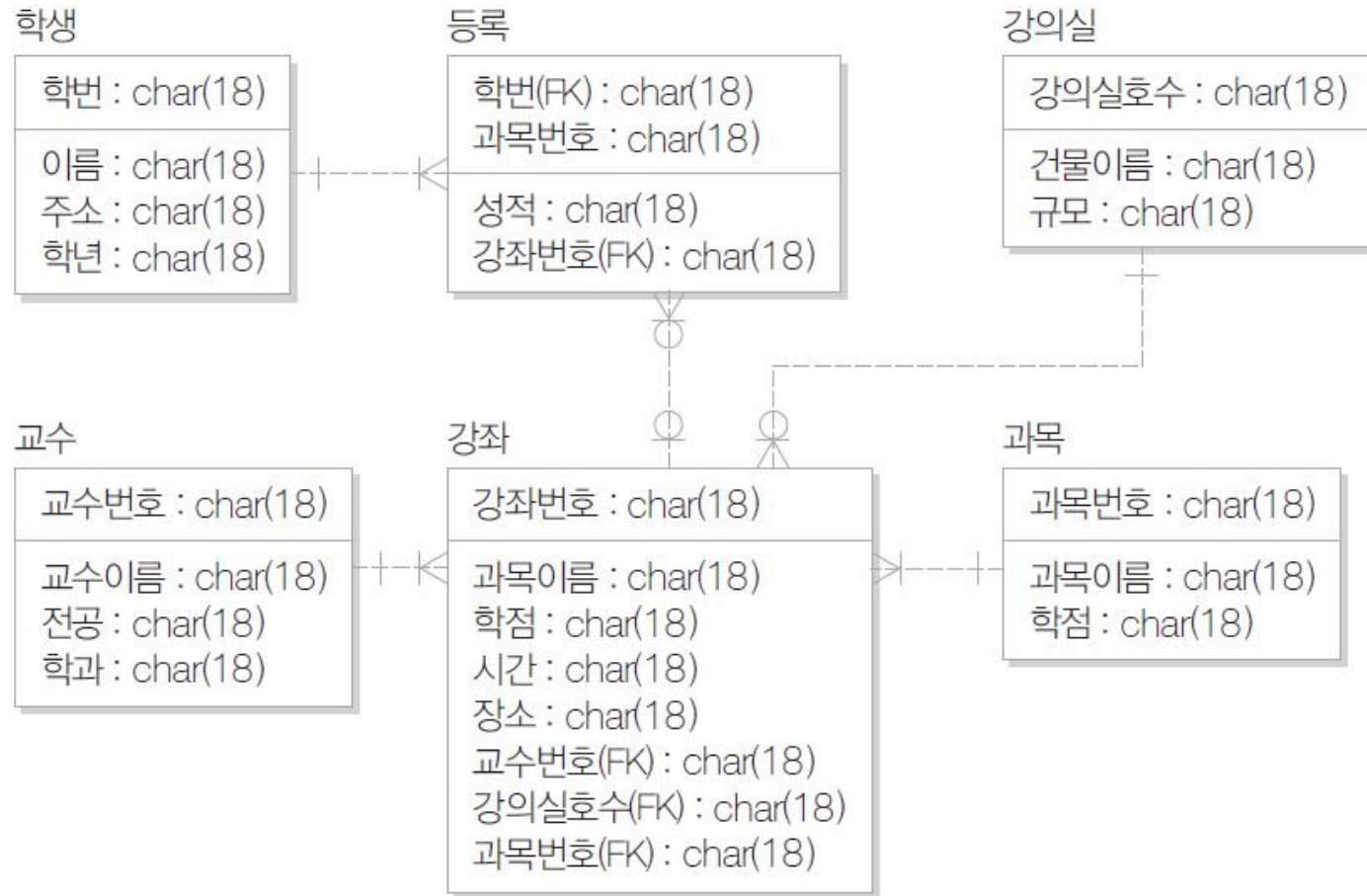
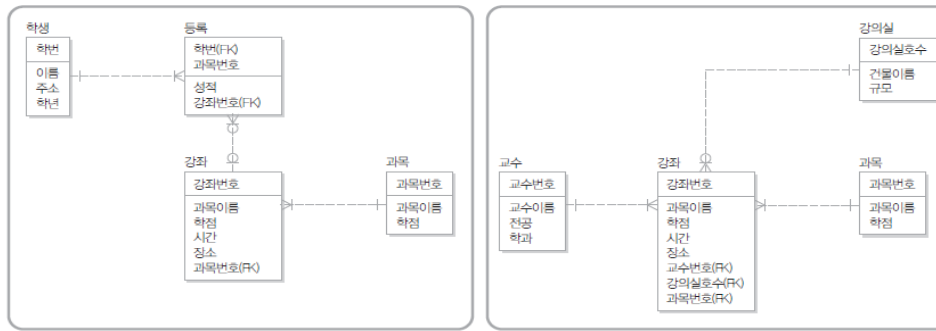
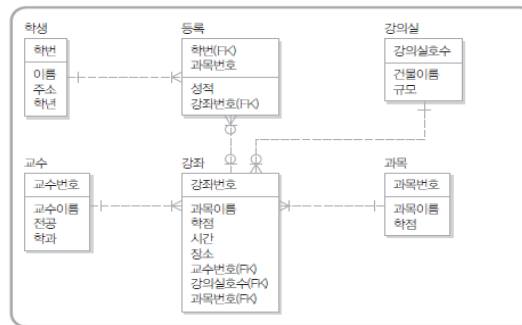


그림 1-31 수강신청 데이터베이스의 내부 스키마

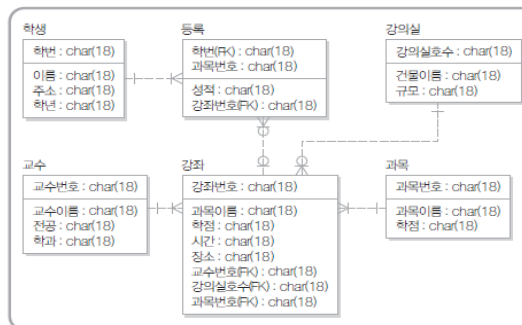
5. 데이터베이스의 개념적 구조



외부 단계
(사용자 관점)



개념 단계
(조직 전체 관점)



내부 단계
(저장 장치 관점)

그림 1-32 수강신청 데이터베이스의 3단계 구조

5. 데이터베이스의 개념적 구조

❖ 데이터 독립성

■ 논리적 데이터 독립성(logical data independence)

- 외부 단계(외부 스키마)와 개념 단계(개념 스키마) 사이의 독립성
- 개념 스키마가 변경되어도 외부 스키마에는 영향을 미치지 않도록 지원
- 논리적 구조가 변경되어도 응용 프로그램에는 영향이 없도록 하는 개념
- 개념 스키마의 테이블을 생성하거나 변경하여도 외부 스키마가 직접 다루는 테이블이 아니면 영향이 없음

■ 물리적 데이터 독립성(physical data independence)

- 개념 단계(개념 스키마)와 내부 단계(내부 스키마) 사이의 독립성
- 저장장치 구조 변경과 같이 내부 스키마가 변경되어도 개념 스키마에 영향을 미치지 않도록 지원
- 성능 개선을 위하여 물리적 저장 장치를 재구성할 경우 개념 스키마나 응용 프로그램 같은 외부 스키마에 영향이 없음
- 물리적 독립성은 논리적 독립성보다 구현하기 쉬움

5. 데이터베이스의 개념적 구조

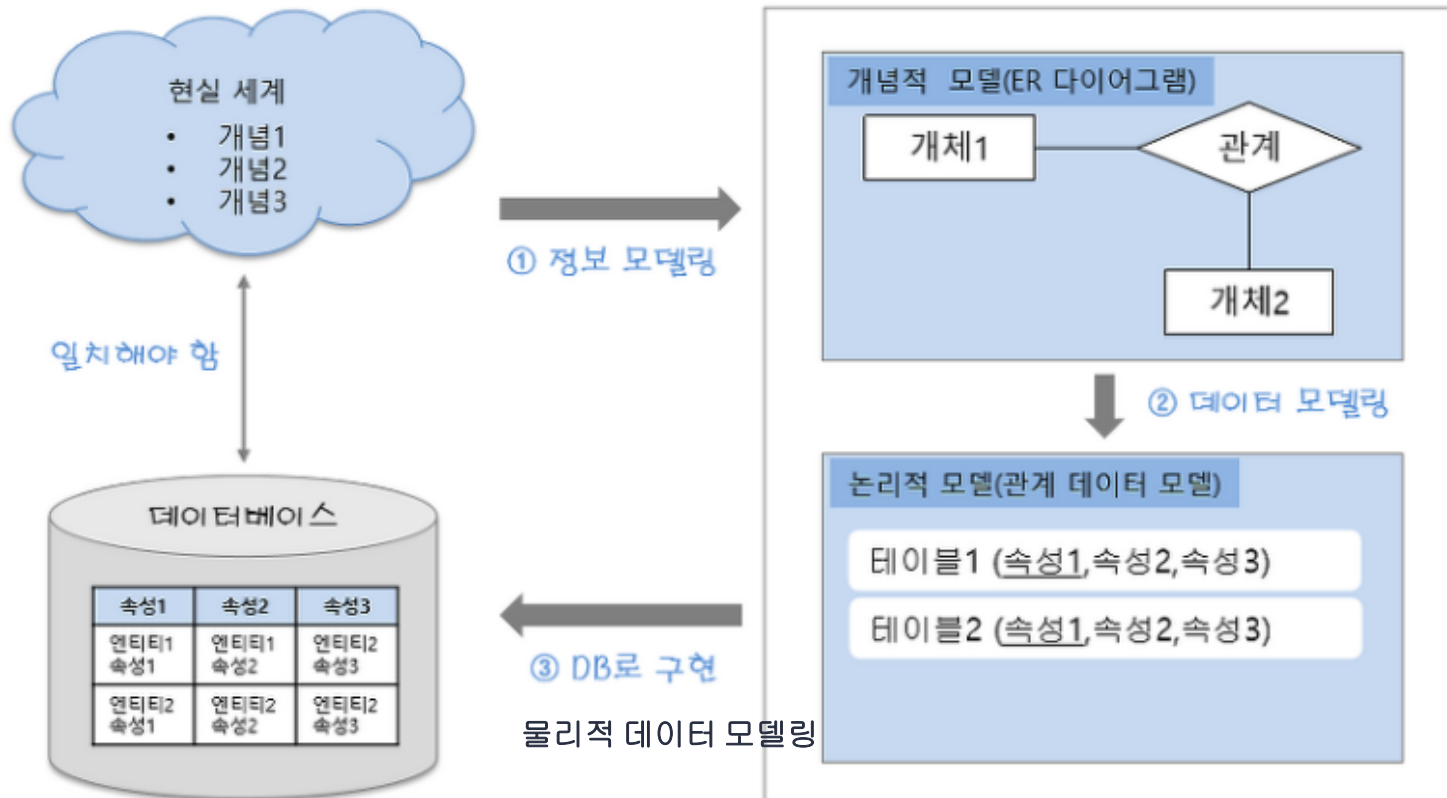
■ ERD 작성

- 2022년 Top 8 무료 ERD 다이어그램 툴 리뷰 (<https://gitmind.com/kr/er-diagram-tool.html>)
- Draw.io ERD (<https://app.diagrams.net/>)
 - 다운로드 (<https://github.com/jgraph/drawio-desktop/releases/tag/v20.8.16>)

5. 데이터베이스의 개념적 구조

■ 데이터베이스 구축

- 데이터 모델링이란 정보시스템 구축의 대상이 되는 업무 내용을 분석하여 이해하고 **약속된 표기법**에 의해 표현하는걸 의미한다. 그리고 이렇게 분석된 **모델**을 가지고 실제 데이터베이스를 생성하여 개발 및 데이터 관리에 사용된다.



데이터베이스 모델링

- 데이터베이스 모델링 데이터 모델링은 현실 세계에서 사용되는 작업이나 사물들을 데이터베이스 개체로 변환하는 과정이라고 말할 수 있다.
- 다른 표현으로는 현실에서 사용되는 것들을 테이블로 변환하는 작업이라고 말할 수 있다.
- 테이블(Table)은 데이터베이스 내에서 데이터를 저장하는 개체
 - 우리가 흔히 사용하는 파일이나 엑셀에서 하나의 시트와 같은 개념
- 제품의 거래를 모델링하고, 모델링의결과를 MySQL을 이용하여 데이터베이스로 구축하는 과정

주문내용 기록 예

주문번호	주문일자	주문자ID	성명	주소	상품코드1	상품명1	단가1	수량1	상품코드2	상품명2	단가2	수량2
202201001	22.01.04	hong1	홍길동	인천송림동	GDS01	연필	1000	10	GDS02	볼펜	1500	20
202201002	22.01.05	hong1	홍길동	인천송림동	GDS01	연필	1000	10	GDS03	노트	2500	10

반복을 제거한 주문내용 기록 예

주문번호	주문일자	주문자ID	성명	주소	상품코드	상품명	단가	수량
202201001	2022.01.04	hong1	홍길동	인천송림동	GDS01	연필	1000	10
202201001	2022.01.04	hong1	홍길동	인천송림동	GDS02	볼펜	1500	20
202201002	2022.01.05	hong1	홍길동	인천송림동	GDS01	연필	1000	10
202201002	2022.01.05	hong1	홍길동	인천송림동	GDS03	노트	2500	10

데이터베이스 모델링

반복을 제거한 주문내용 기록 예

주문번호	주문일자	주문자ID	성명	주소	상품코드	상품명	단가	수량
202201001	2022.01.04	hong1	홍길동	인천송림동	GDS01	연필	1000	10
202201001	2022.01.04	hong1	홍길동	인천송림동	GDS02	볼펜	1500	20
202201002	2022.01.05	hong1	홍길동	인천송림동	GDS01	연필	1000	10
202201002	2022.01.05	hong1	홍길동	인천송림동	GDS03	노트	2500	10

주문의 기본정보

주문번호	주문일자	주문자ID	성명	주소
202201001	2022.01.04	hong1	홍길동	인천송림동
202201002	2022.01.05	hong1	홍길동	인천송림동

주문의 상세정보

주문번호	상품코드	상품명	단가	수량
202201001	GDS01	연필	1000	10
202201001	GDS02	볼펜	1500	20
202201002	GDS01	연필	1000	10
202201002	GDS03	노트	2500	10

네모 안의 내용은 중복이다. 중복을 제거해 보자

주문번호	주문일자	주문자ID
202201001	2022.01.04	hong1
202201002	2022.01.05	hong1

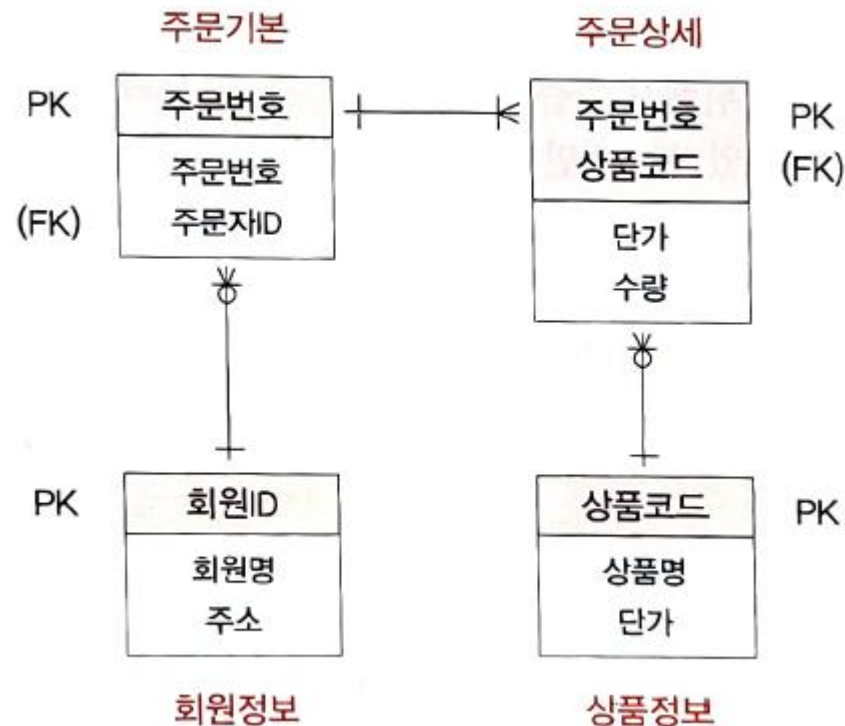
주문자ID	성명	주소
hong1	홍길동	인천송림동

주문번호	상품코드	단가	수량
202201001	GDS01	1000	10
202201001	GDS02	1500	20
202201002	GDS01	1000	10
202201002	GDS03	2500	10

상품코드	상품명
GDS01	연필
GDS02	볼펜
GDS03	노트

정규화 Normalization 및 개체 관계도 ERD; Entity Relationship Diagram

- 앞서 모델링을 통하여 데이터베이스 내의 불필요한 데이터 반복과 중복이 제거하였고, 개체 Entity 간의 관계에 대한 데이터의 삽입·수정·삭제 연산을 수행할 때 부작용, 즉 데이터의 무결성(Integrity) 문제가 발생하지 않도록 데이터베이스를 올바르게 설계해 나가는 과정이 정규화 Normalization이다.



1. 데이터베이스
2. 데이터베이스의 개념
3. 데이터베이스의 특징
4. 데이터베이스 시스템의 구성
5. 정보 시스템의 발전
6. DBMS의 장점
7. SQL
8. 데이터베이스 관리자(DBA)
9. 데이터 모델
10. 3단계 데이터베이스 구조
11. 데이터 독립성