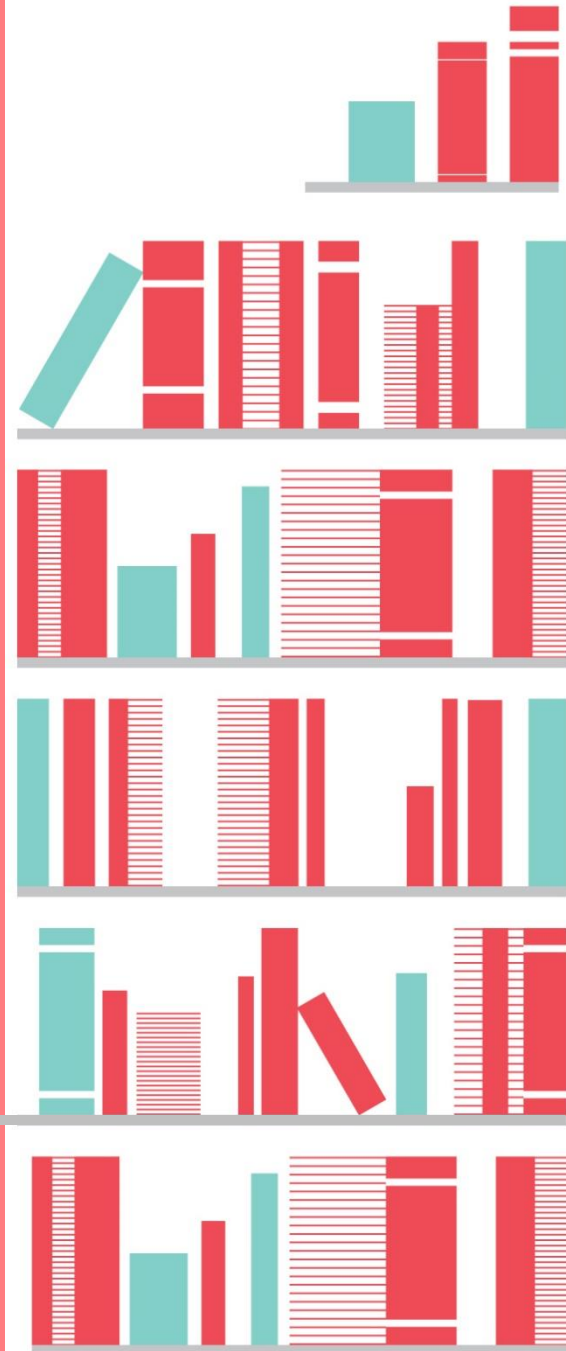


Chapter 01

데이터베이스 시스템

오라클로 배우는 데이터베이스 개론과 실습



목차

1. 데이터베이스와 데이터베이스 시스템
2. 데이터베이스 시스템의 발전
3. 파일 시스템과 DBMS
4. 데이터베이스 시스템의 구성

학습목표

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

01. 데이터베이스와 데이터베이스 시스템

- 데이터, 정보, 지식
- 일상생활의 데이터베이스
- 데이터베이스의 개념 및 특징
- 데이터베이스 시스템의 구성

1. 데이터, 정보, 지식

- 데이터 : 관찰의 결과로 나타난 정량적 혹은 정성적인 실제 값
- 정보 : 데이터에 의미를 부여한 것
- 지식 : 사물이나 현상에 대한 이해



그림 1-1 데이터, 정보, 지식

2. 일상생활의 데이터베이스

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

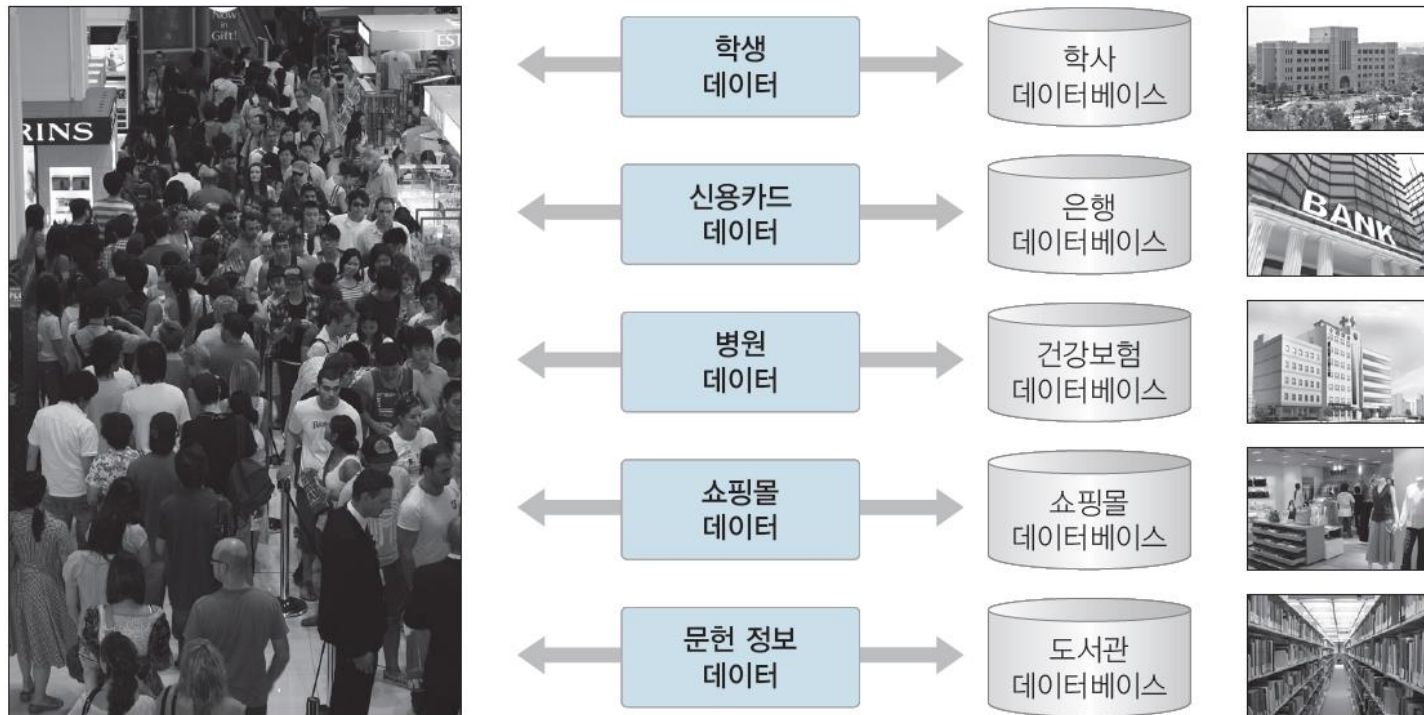


그림 1-2 일상생활에서 생성되는 데이터베이스

2. 일상생활의 데이터베이스

신대방삼거리역/
 대표 : 김양숙외1(1081897274) T:02-812-3235
 담당 : 점주 POS NO : 01
 일자 : 2013-03-24 18:12:38

	QTY	D/C	AMT
소프트 아이스크림	1	0	500
과세금액			273
부가세			227
총매출액			500
합계			500
받은 돈			500
[카드]			500

소비자 중심 경영 인증기업(CCM)

« 카드 승인 »

[카드번호] 940951*****271
 [승인금액] 13,500
 [승인번호] 33517480 [일시불]

이용안내

- 미 홈-티켓(Home-Ticket)은 철도승차권이며 해당열차에 지정 승차자가 승차하지 않습니다.
- 코레일직원이 본인 확인을 요구할 경우 신분증을 제시하셔야 합니다.
- 열차출발시각 이전에는 인터넷, 역 및 대리점에서 반환이 가능하나 출발시각 이후에는 역과 대리점에서만 반환이 가능하며 도착역 도착시각 이후에는 반환이 불가능합니다.
- 코레일 멤버십회원은 열차출발 24시간 전부터 출발시각 전까지 전화로 승차권 반환신청이 가능합니다.

- 문의전화 ☎ 1588-7788 / 1544-7788

승차일 2013년 03월 10일(일)

부산 ▶ **서울**

Busan ▶ Seoul

14:45 ▶ 17:47

KTX 제310 열차 일반실 11호차 10C석

운임요금 52,900 환입금액 2,600 영수액 50,300

어른 신용 940040

승차자 변소현 예약자 변소현

인쇄일시 2013/03/08 12:17

반환번호 80003-0308-12248-10

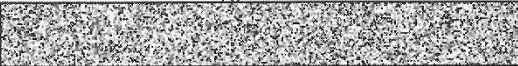


그림 1-3 패스트푸드 체인점과 철도청의 데이터베이스 - 간단한 거래도 많은 데이터가 포함

2. 일상생활의 데이터베이스

표 1-1 데이터베이스의 활용 분야

종류	특징
생활과 문화	<ul style="list-style-type: none">• 기상정보 : 날씨 정보를 제공• 교통정보 : 교통상황 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 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 법률 정보를 제공• 통계정보 : 국가기관의 통계 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 개념

① 통합된 데이터(integrated data)

데이터를 통합하는 개념으로, 각자 사용하던 데이터의 중복을 최소화하여 중복으로 인한 데이터 불일치 현상을 제거

② 저장된 데이터(stored data)

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

③ 운영 데이터(operational data)

조직의 목적을 위해 사용되는 데이터를 의미한다. 즉 업무를 위한 검색을 할 목적으로 저장된 데이터

④ 공용 데이터(shared data)

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

3. 데이터베이스의 개념

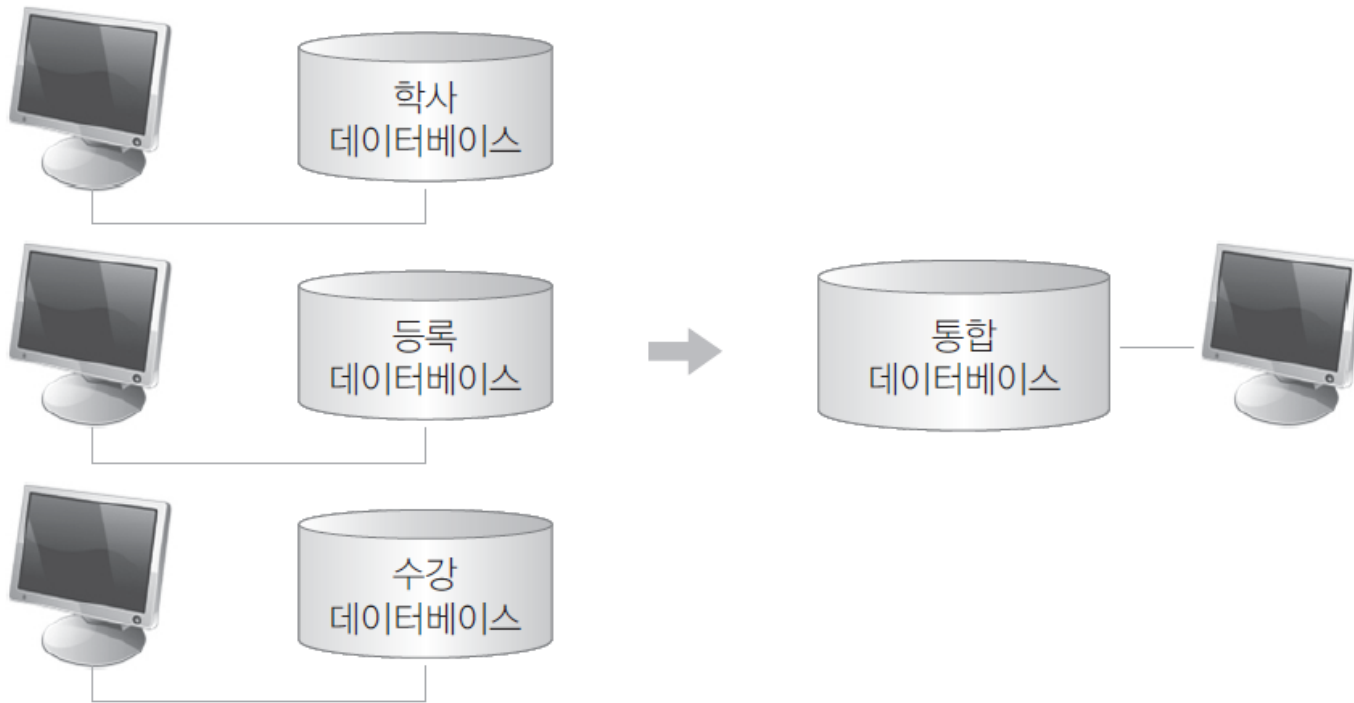


그림 1-4 데이터베이스의 개념 : 데이터베이스는 운영 데이터를 통합하여 저장하며 공용으로 사용된다

3. 데이터베이스의 특징

① 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.

② 지속적인 변화(continuous change)

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

③ 동시 공유(concurrent sharing)

데이터베이스는 서로 다른 업무 또는 여러 사용자에게 동시에 공유된다. 동시(concurrent)는 병행이라고도 하며, 데이터베이스에 접근하는 프로그램이 여러 개 있다는 의미다.

④ 내용에 따른 참조(reference by content)

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

4. 데이터베이스 시스템의 구성

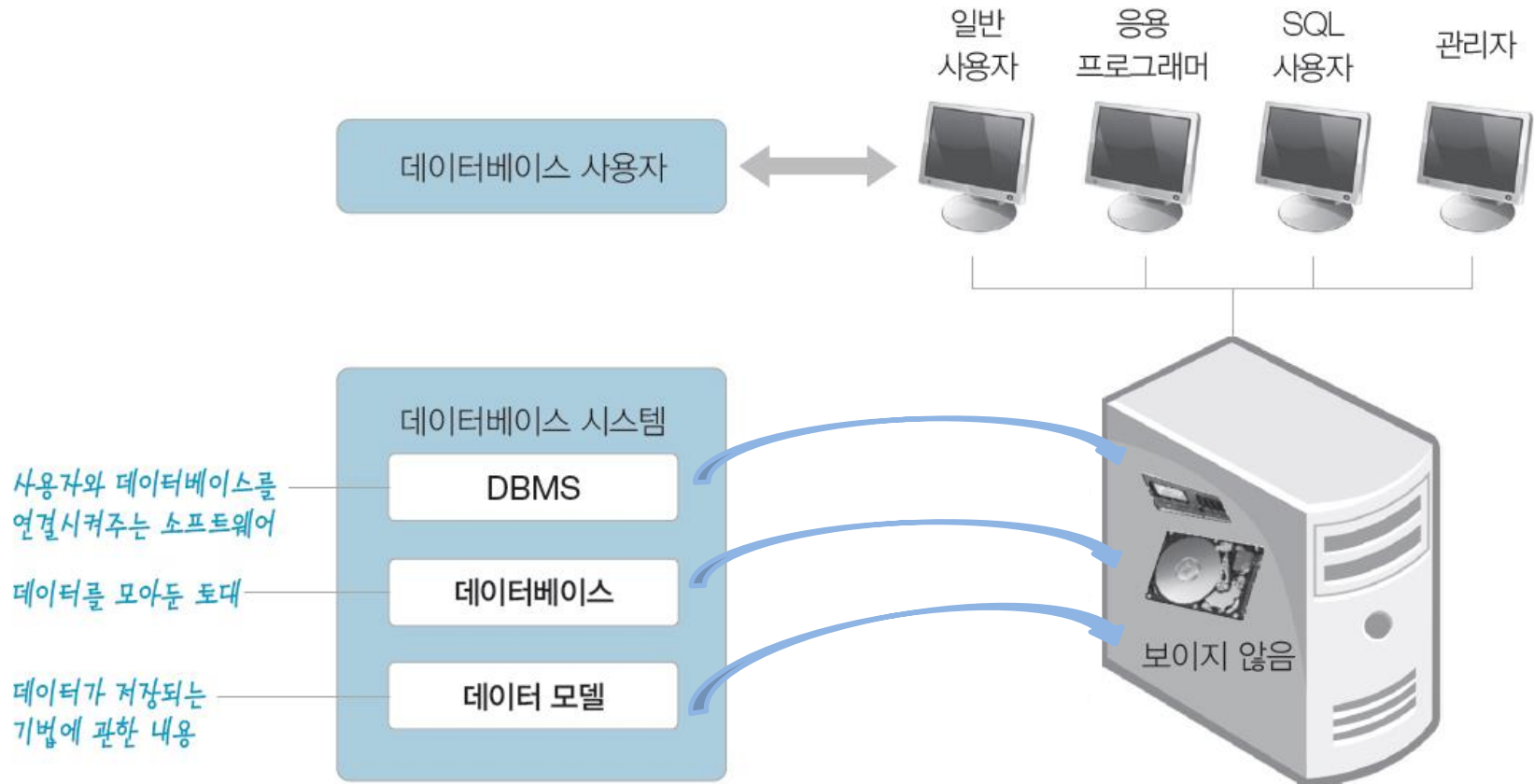


그림 1-5 데이터베이스 시스템의 구성 요소와 물리적인 위치

02. 데이터베이스 시스템의 발전

- 마당서점과 데이터베이스 시스템
- 정보 시스템의 발전

[1단계] 마당서점의 시작



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

그림 1-6 마당서점 초기

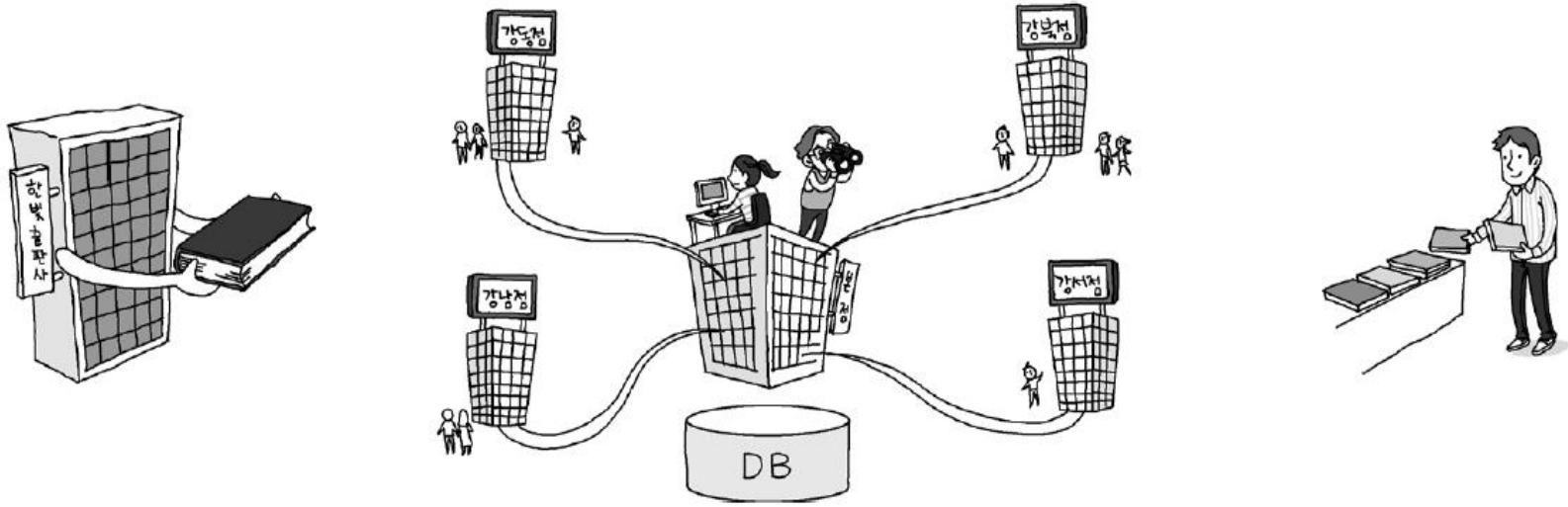
[2단계] 컴퓨터의 도입



- 도서 : 1000권
- 고객 : 근처 학교의 학생, 지역 주민
- 고객 서비스 : 컴퓨터를 이용하여 도서 검색, 직원 고용
- 업무 : 회계업무(컴퓨터 사용), 파일 시스템

그림 1-7 마당서점 전산화

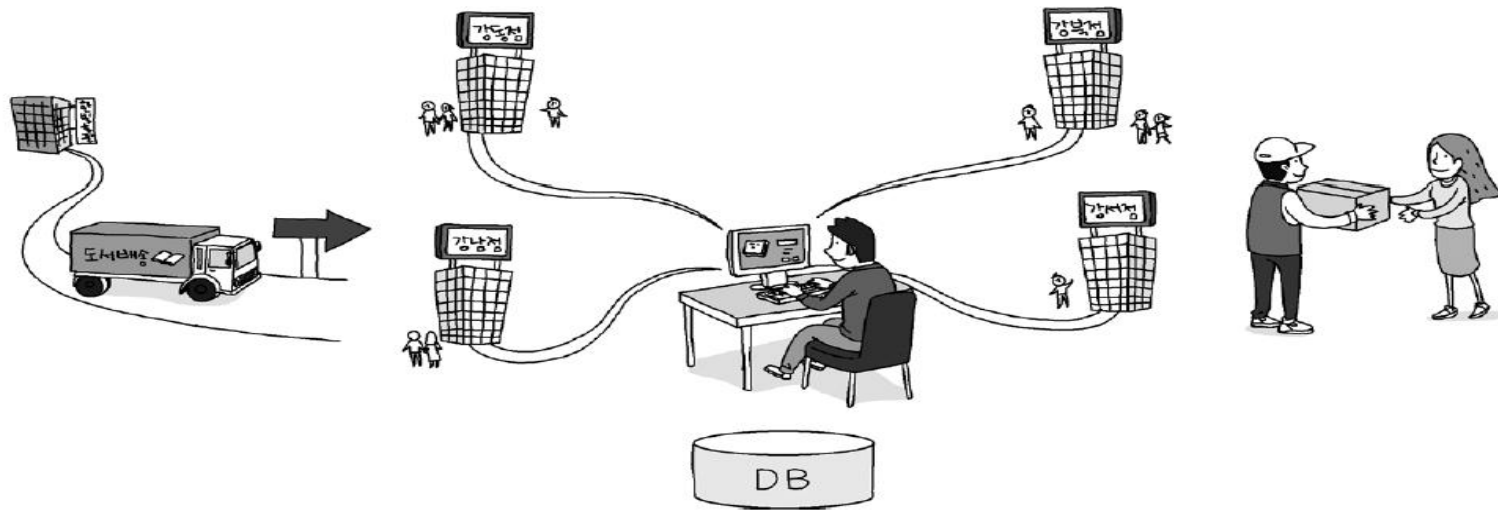
[3단계] 지점 개설 및 데이터베이스 구축



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

그림 1-8 마당서점 DBMS 도입

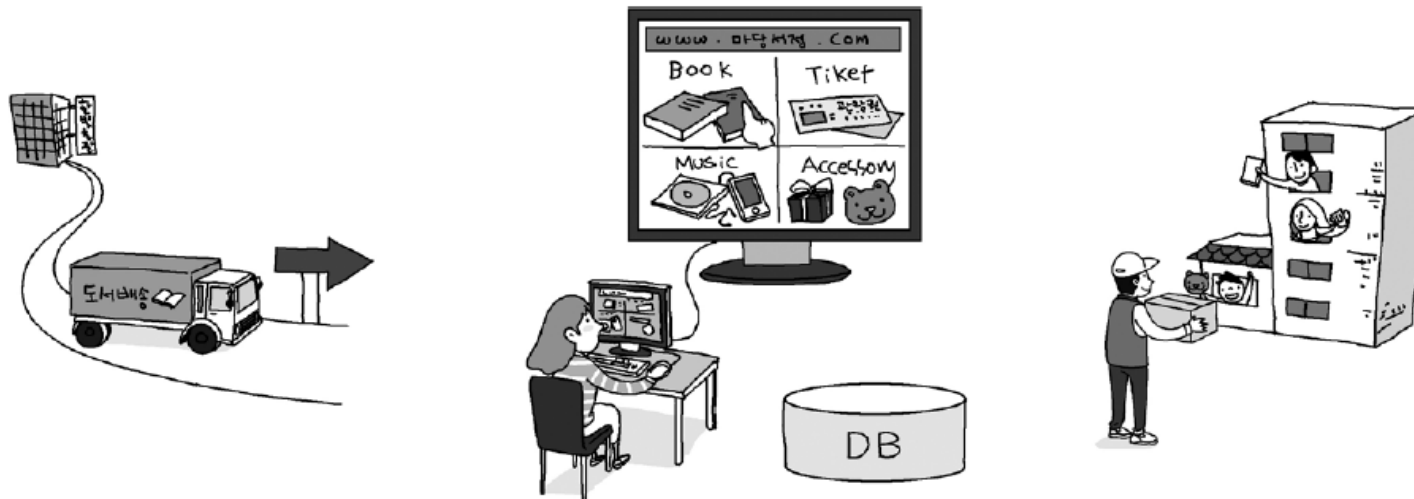
[4단계] 홈페이지 구축



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

그림 1-9 마당서점 인터넷 서비스 실시

[5단계] 인터넷 쇼핑몰 운영



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

그림 1-10 마당서점 인터넷 쇼핑몰 운영

1. 마당서점과 데이터베이스 시스템

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

① 파일 시스템

- 데이터를 파일 단위로 파일 서버에 저장
- 각 컴퓨터는 LAN을 통하여 파일 서버에 연결되어 있고, 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

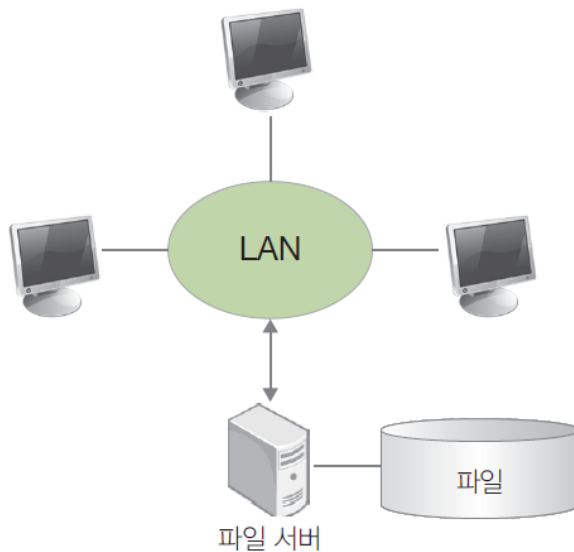


그림 1-11 파일 시스템

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

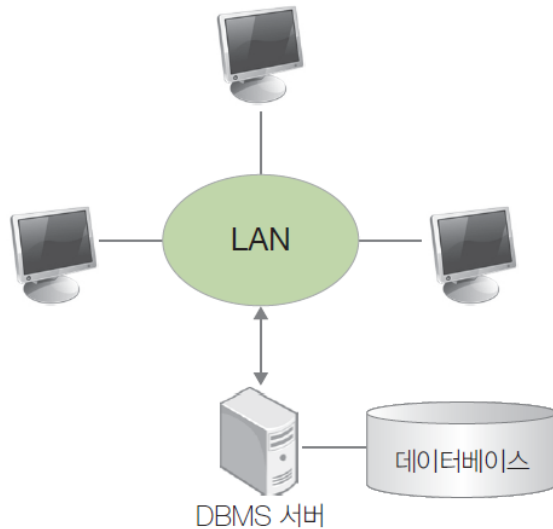


그림 1-12 데이터베이스 시스템

2. 정보 시스템의 발전

③ 웹 데이터베이스 시스템

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

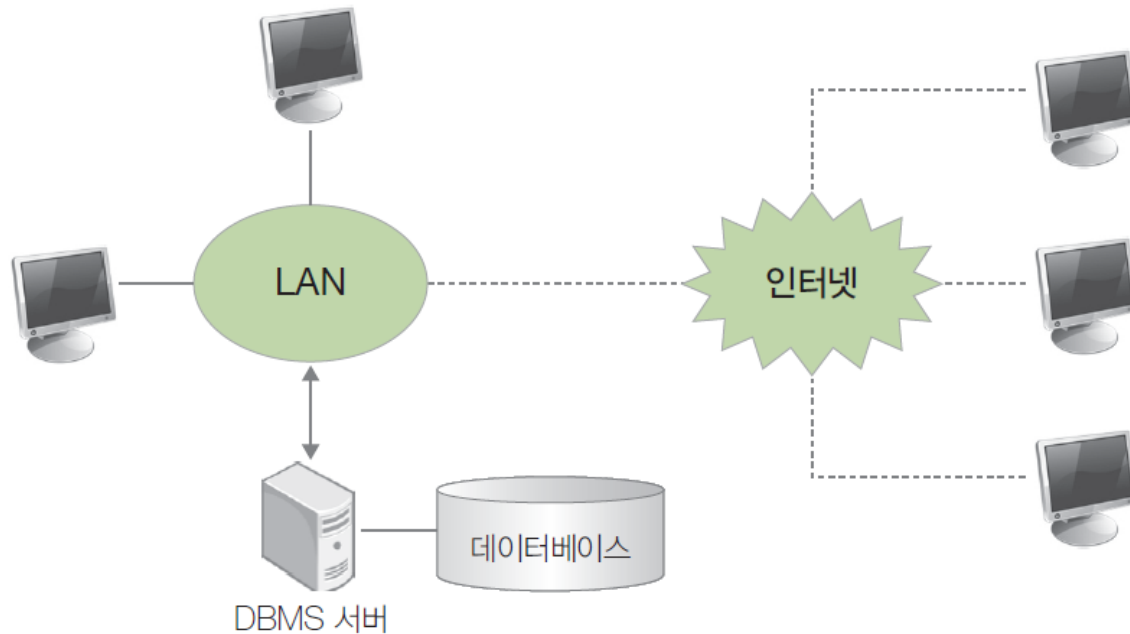


그림 1-13 웹 데이터베이스 시스템

2. 정보 시스템의 발전

4 분산 데이터베이스 시스템

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

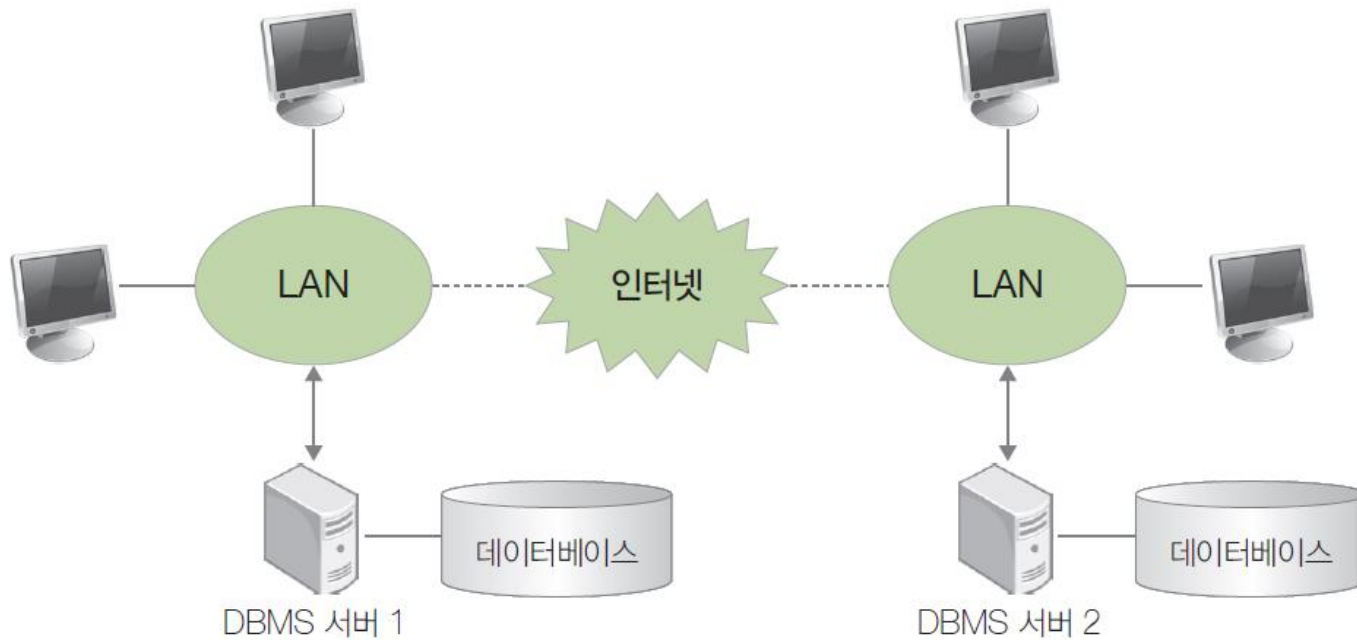


그림 1-14 분산 데이터베이스 시스템

2. 정보 시스템의 발전

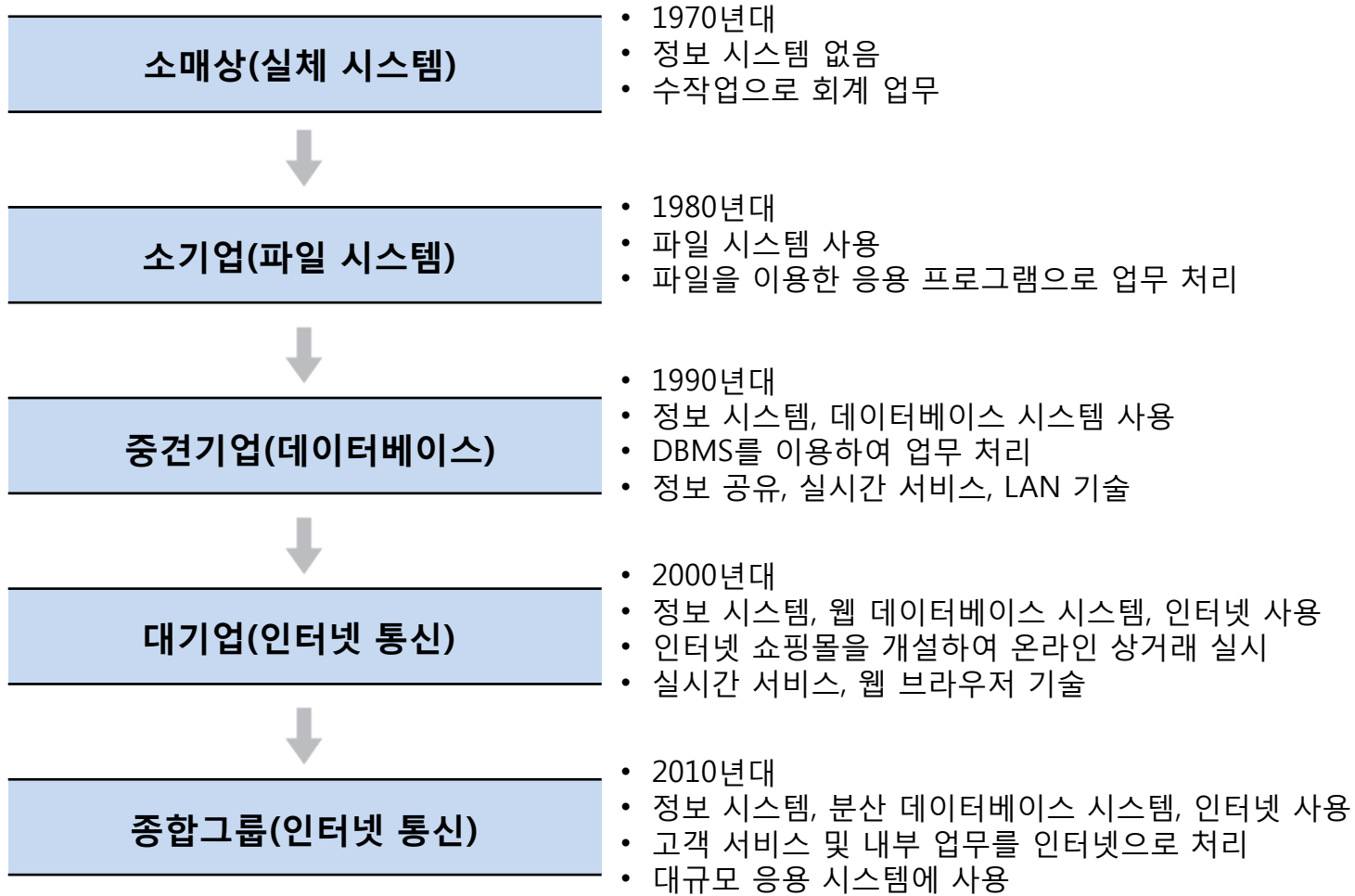


그림 1-15 정보 시스템의 발전과 기업의 업무 환경 변화

03. 파일 시스템과 DBMS

- 마당서점 데이터를 저장하는 방법
- 마당서점 데이터의 저장 방법 비교
- 파일 시스템과 DBMS의 비교

1. 마당서점 데이터를 저장하는 방법

1. 데이터를 프로그램 내부에 저장하는 방법
2. 파일 시스템을 사용하는 방법
3. DBMS를 사용하는 방법



그림 1-16 고객 서비스를 온라인 정보 서비스로 전환

1.1 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

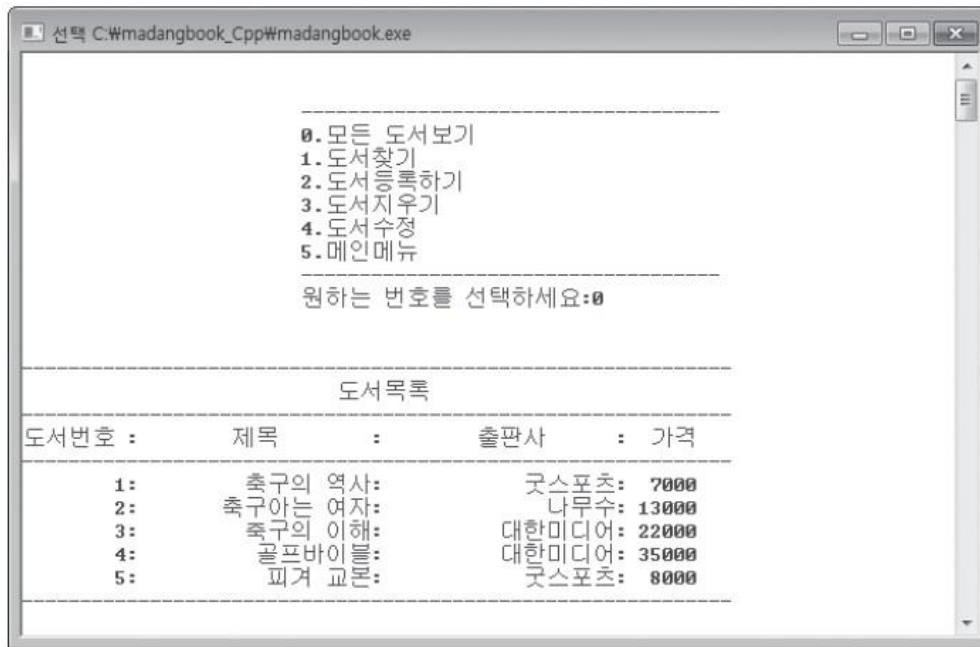


그림 1-17 도서 검색 프로그램

1.1 데이터를 프로그램 내부에 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();
    ....
}
```

1.2 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

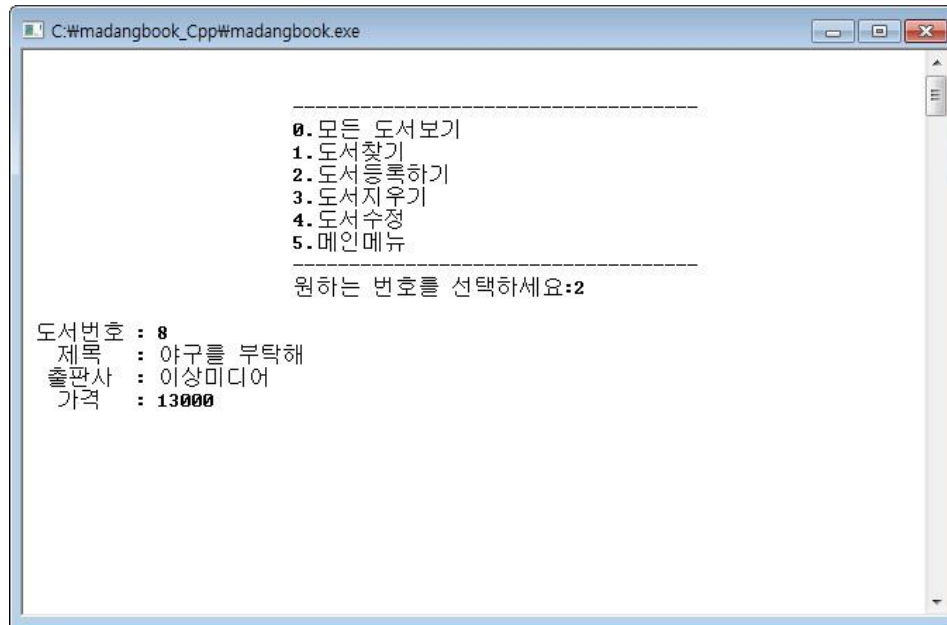


그림 1-18 도서 검색 프로그램에서 도서를 등록하는 화면

1.2 파일 시스템을 사용하는 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( );

    .....
}
```

1.3 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

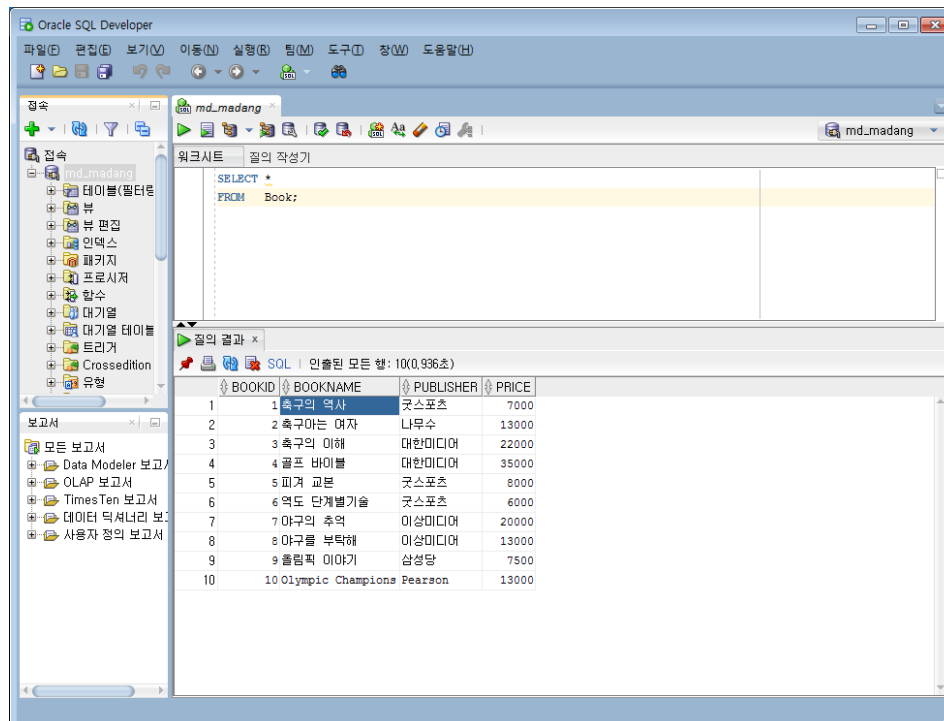


그림 1-19 오라클(SQL Developer)의 데이터베이스 관리 화면

1.3 DBMS를 사용하는 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;

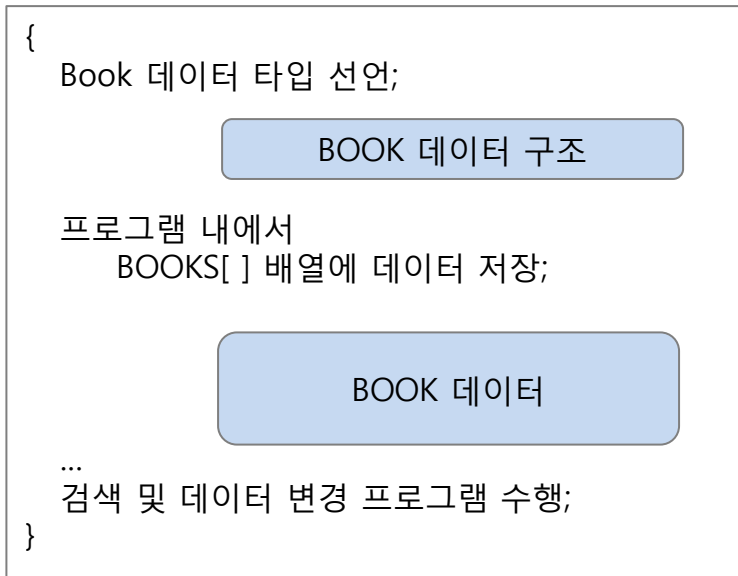
    /* 모든 도서보기 프로그램 호출 */
    search_all( );

    /* 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. 마당서점 데이터의 저장 방법 비교

[프로그램 1]

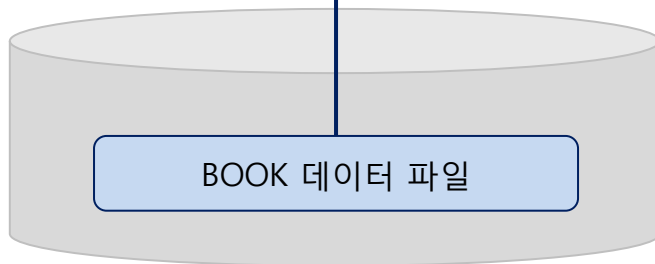


- 프로그램에 데이터 정의와 데이터 값을 모두 포함하는 방식
- 프로그램에 BOOK 데이터 구조를 정의하고 데이터 값도 직접 변수에 저장함
- 데이터 구조 혹은 데이터 값이 바뀌면 프로그램을 다시 컴파일해야 함

2. 마당서점 데이터의 저장 방법 비교

[프로그램 2]

```
{  
  Book 데이터 타입 선언;  
    
  BOOK 데이터 구조  
    
  파일로부터 데이터를 불러와  
  BOOKS[ ] 배열에 저장;  
  ...  
  검색 및 데이터 변경 프로그램 수행;  
}
```



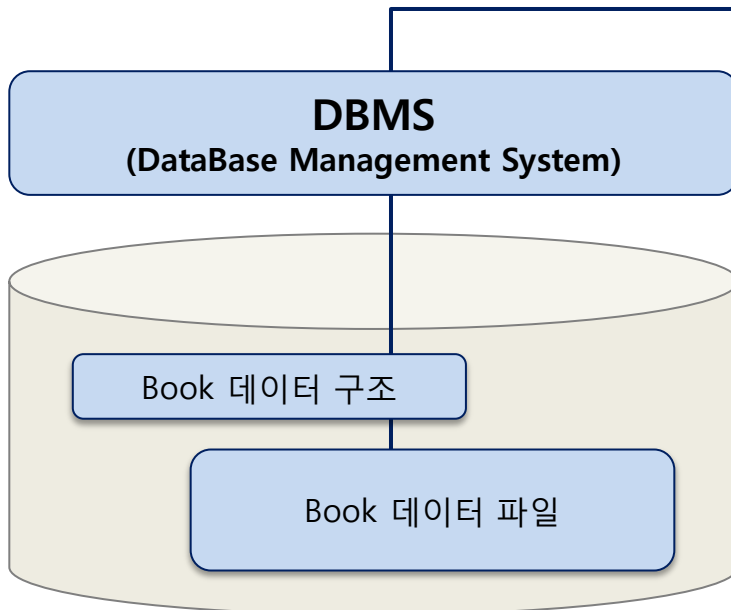
- 파일에 데이터 값, 프로그램에 데이터 정의를 포함하는 방식
- 프로그램에 BOOK 데이터 구조만 정의하고, 데이터 값은 book.dat라는 파일에 저장됨
- 데이터 값이 바뀌면 프로그램에 변경이 없지만, 데이터 구조가 바뀌면 프로그램을 다시 컴파일해야 함

2. 마당서점 데이터의 저장 방법 비교

[프로그램 3]

```
{  
  /* 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

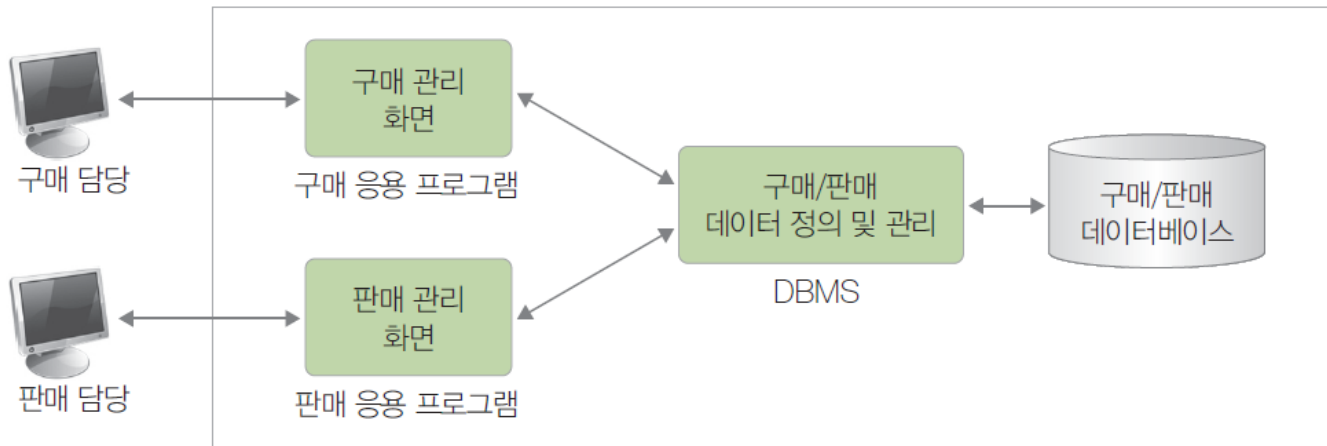


그림 1-21 DBMS로 구축된 구매 및 판매 응용 프로그램

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	보통	데이터 무결성 유지, 데이터 표준 준수 용이

04. 데이터베이스 시스템의 구성

- 데이터베이스 언어
- 데이터베이스 사용자
- DBMS
- 데이터 모델
- 데이터베이스의 개념적 구조

04. 데이터베이스 시스템의 구성

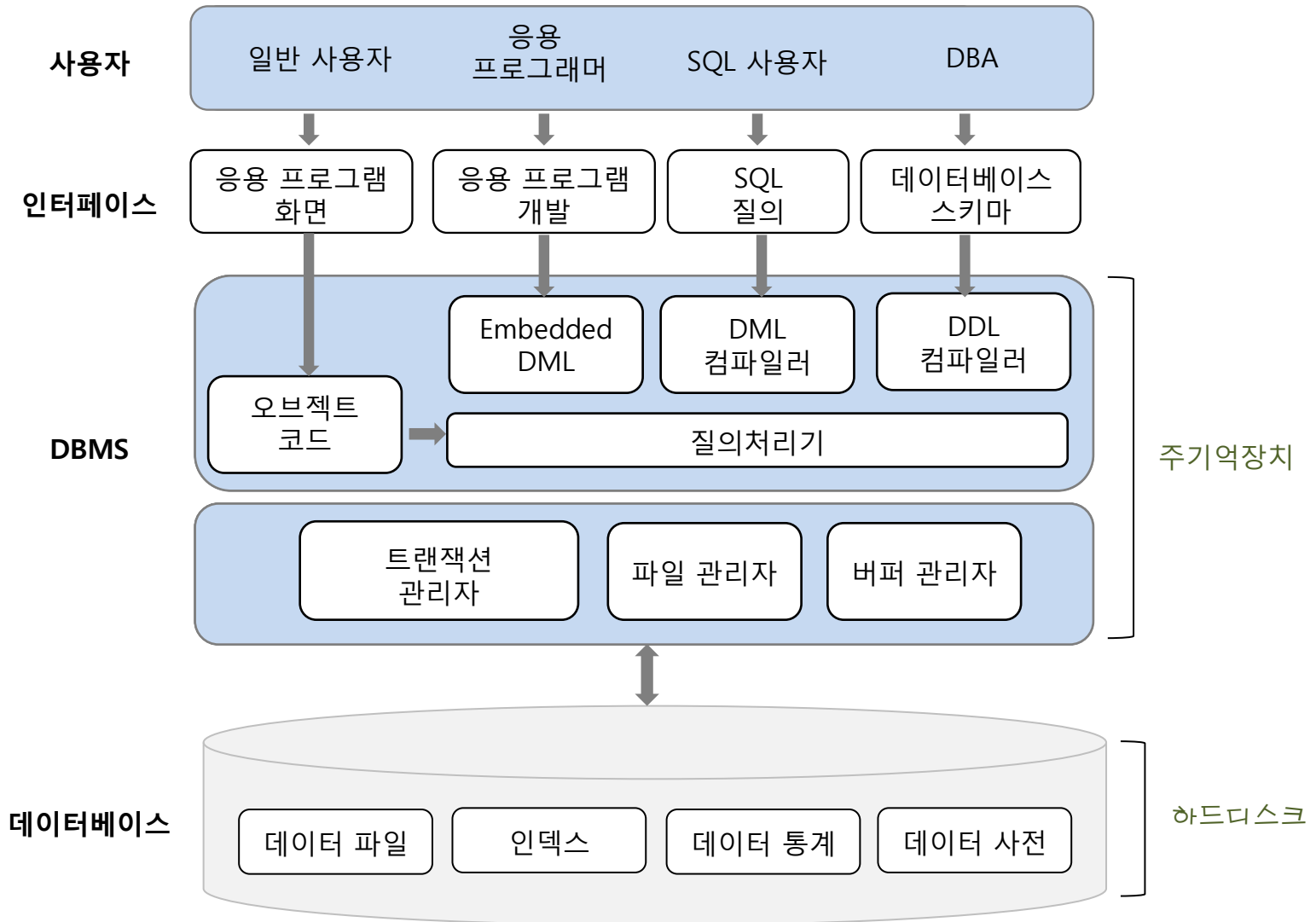


그림 1-22 데이터베이스 시스템의 구성

1. 데이터베이스 언어

■ SQL

- 데이터 정의어(DDL, Data Definition Language)
- 데이터 조작어(DML, Data Manipulation Language)
- 데이터 제어어(DCL, Data Control Language)

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

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 등의 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

4. 데이터 모델

- 계층 데이터 모델(hierarchical data model)
- 네트워크 데이터 모델(network data model)
- 객체 데이터 모델(relational data model)
- 관계 데이터 모델(object 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

① 포인터 사용 : 계층 데이터 모델, 네트워크 데이터 모델

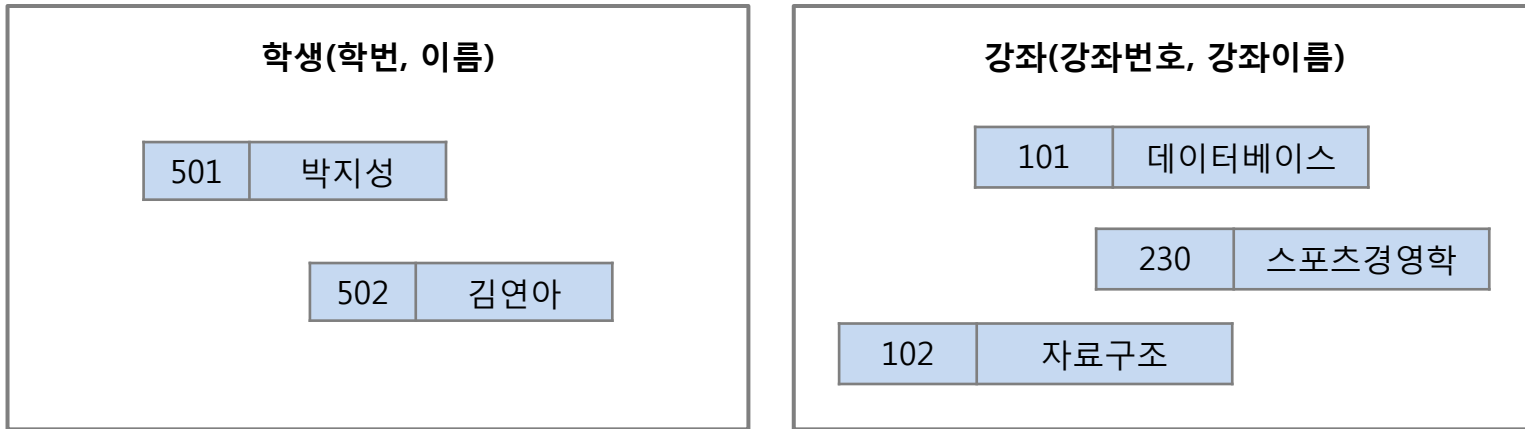


그림 1-23 관계 표현을 위한 예시

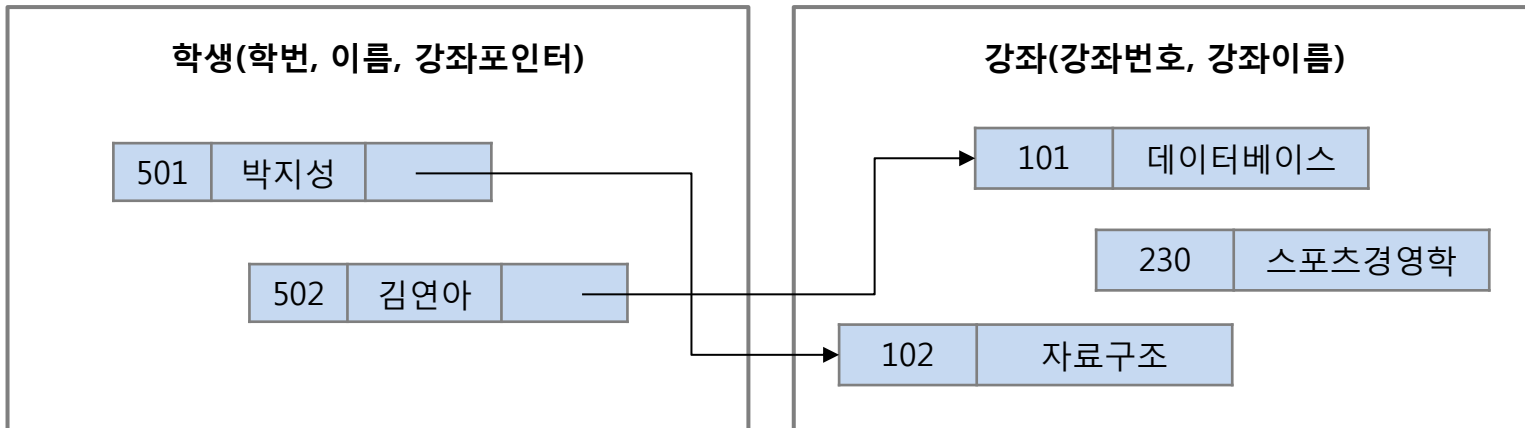


그림 1-24 포인터를 사용하여 관계 표현

4. 데이터 모델

2 속성 값 사용 : 관계 데이터 모델

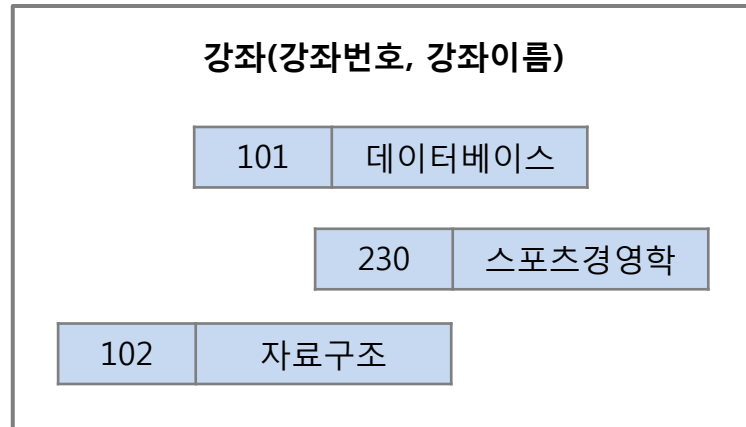
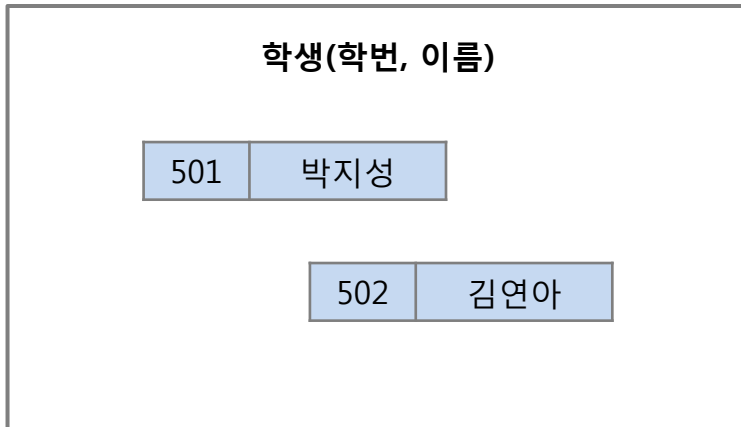


그림 1-23 관계 표현을 위한 예시

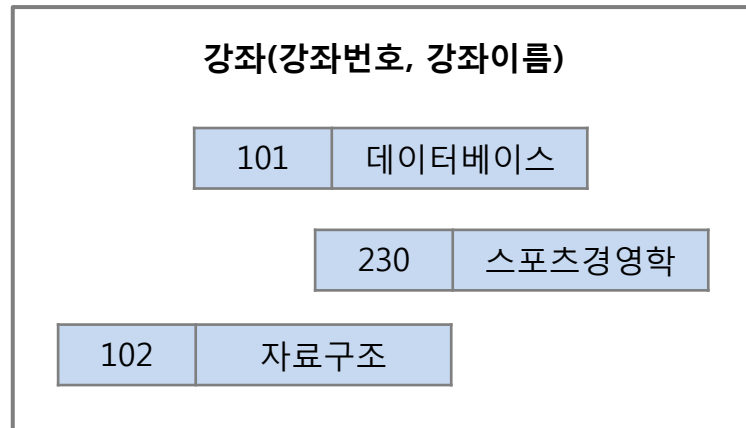
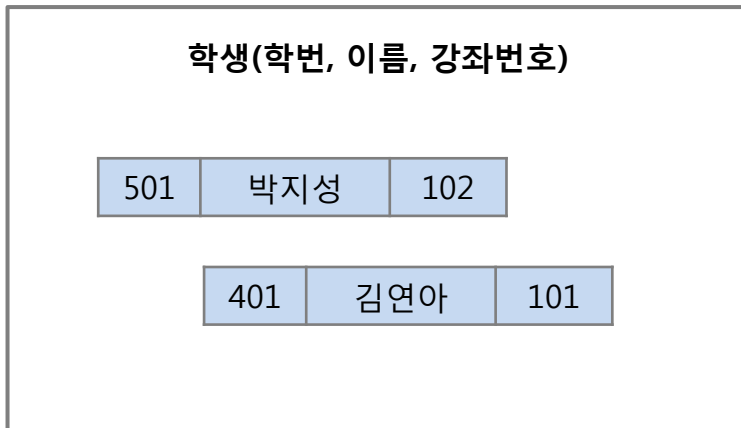


그림 1-25 속성 값을 사용하여 관계 표현

4. 데이터 모델

3 객체식별자 사용 : 객체 데이터 모델

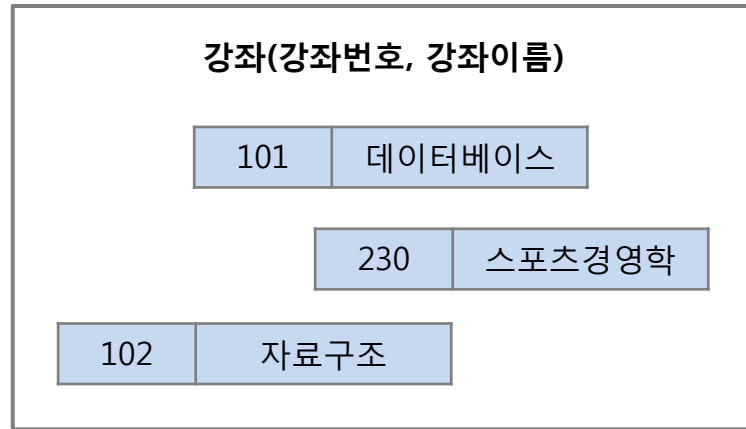
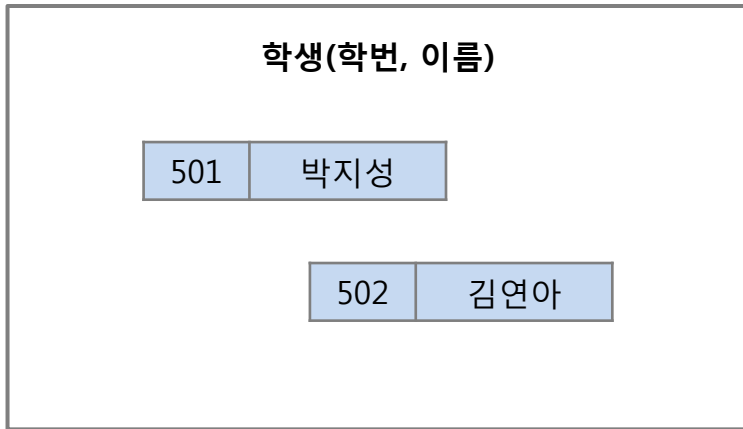


그림 1-23 관계 표현을 위한 예시

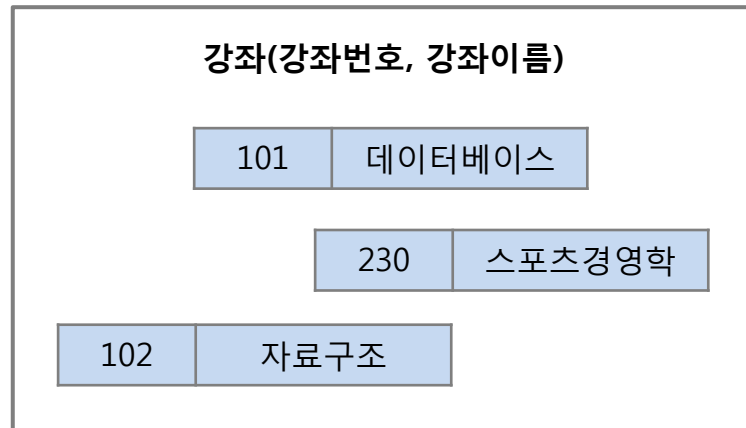
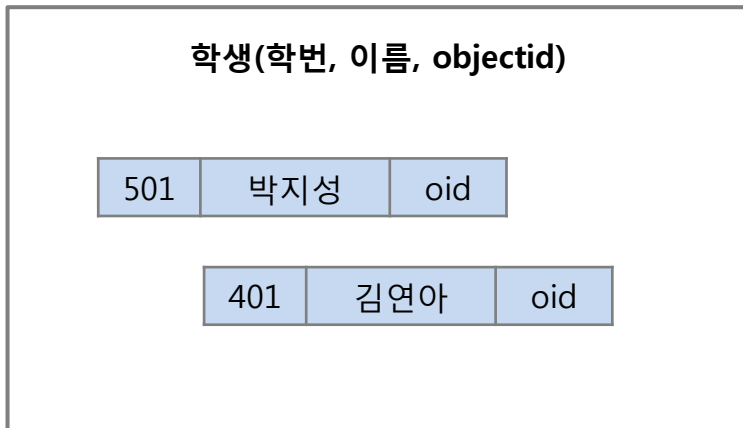
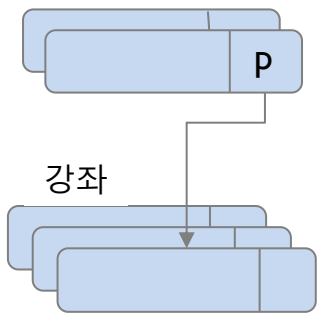
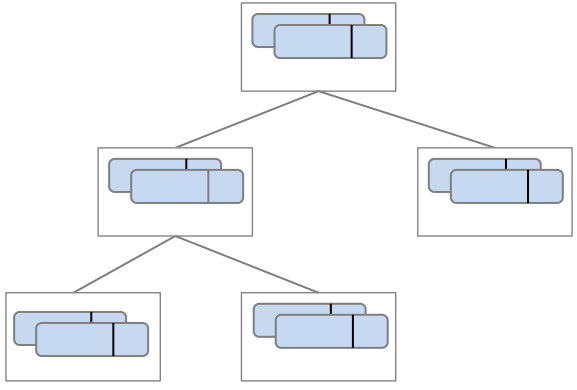
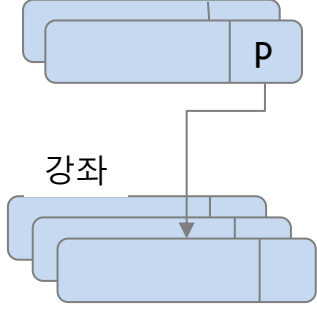
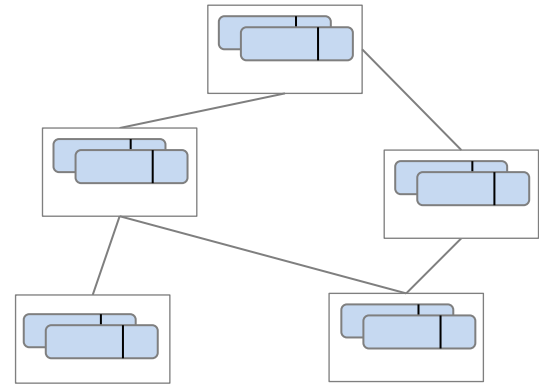


그림 1-26 객체식별자를 사용하여 관계 표현


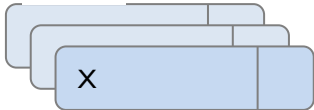
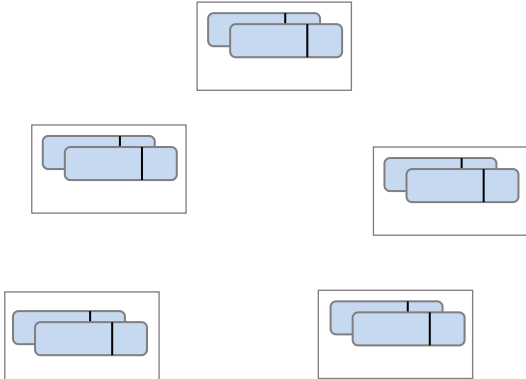
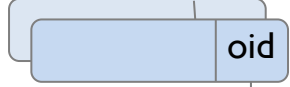

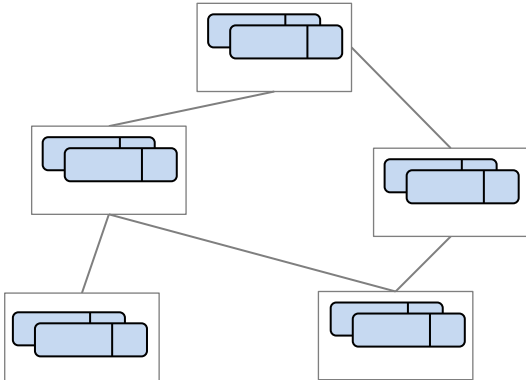
4. 데이터 모델

표 1-9 데이터 모델과 각 모델에서 관계의 표현 방법

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

4. 데이터 모델

표 1-9 데이터 모델과 각 모델에서 관계의 표현 방법

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

4. 데이터 모델

표 1-10 데이터 모델의 역사

데이터 모델	1970년대	1980년대	1990년대	2000년대	2010년대
제품 종류					
계층데이터모델 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				⇒	⇒

(주) 그림은 데이터 모델이 주로 사용되는 시기를 표시한다.

계층과 네트워크 모델은 1960년대, 관계 데이터모델은 1970년대에 처음 사용되기 시작하였다. ,

5.1 3단계 데이터베이스 구조

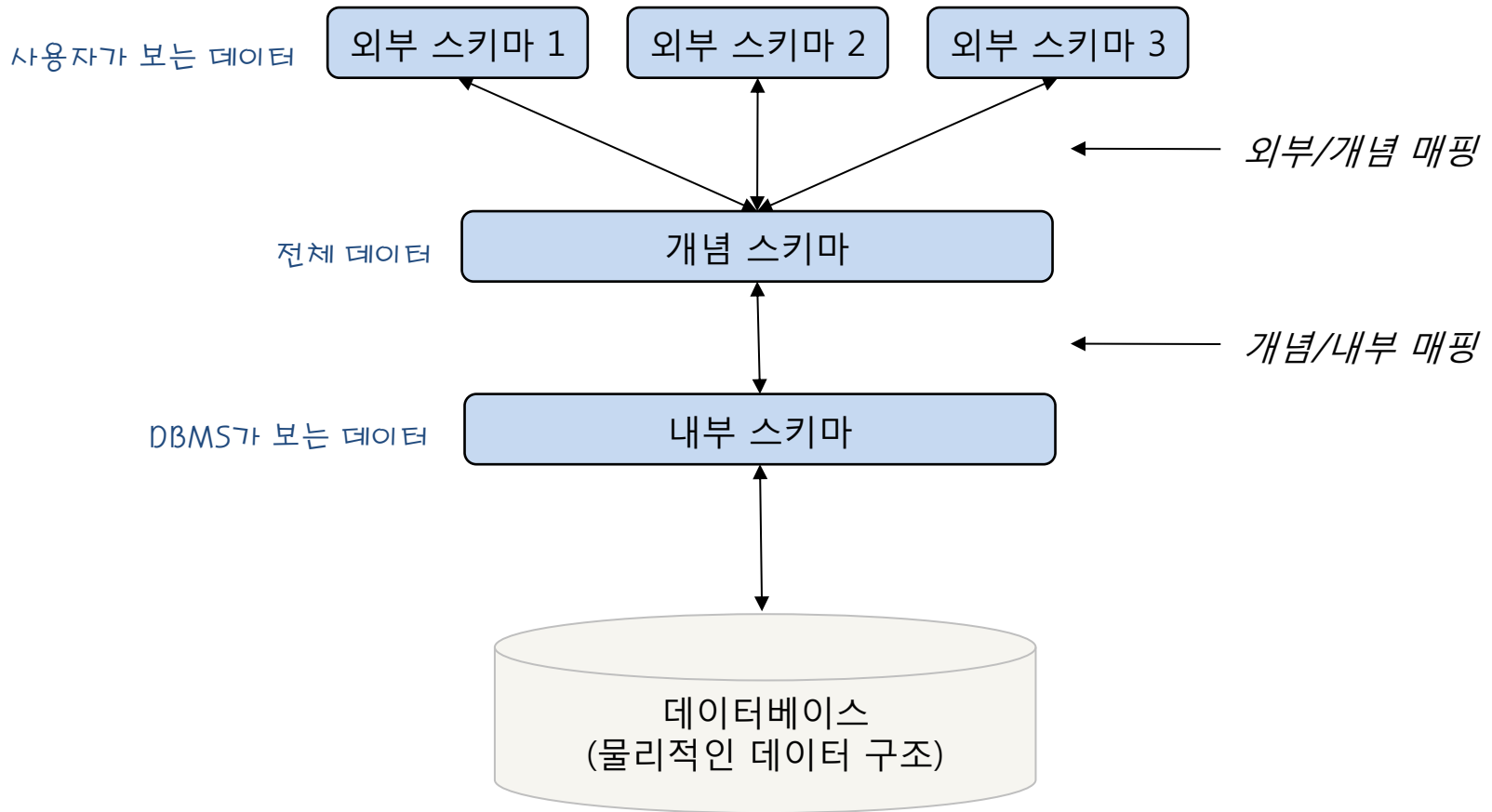


그림 1-27 ANSI의 3단계 데이터베이스 구조

5.1 3단계 데이터베이스 구조

■ 외부 스키마

- 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 스키마

- 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

5.1 3단계 데이터베이스 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

5.1 3단계 데이터베이스 구조

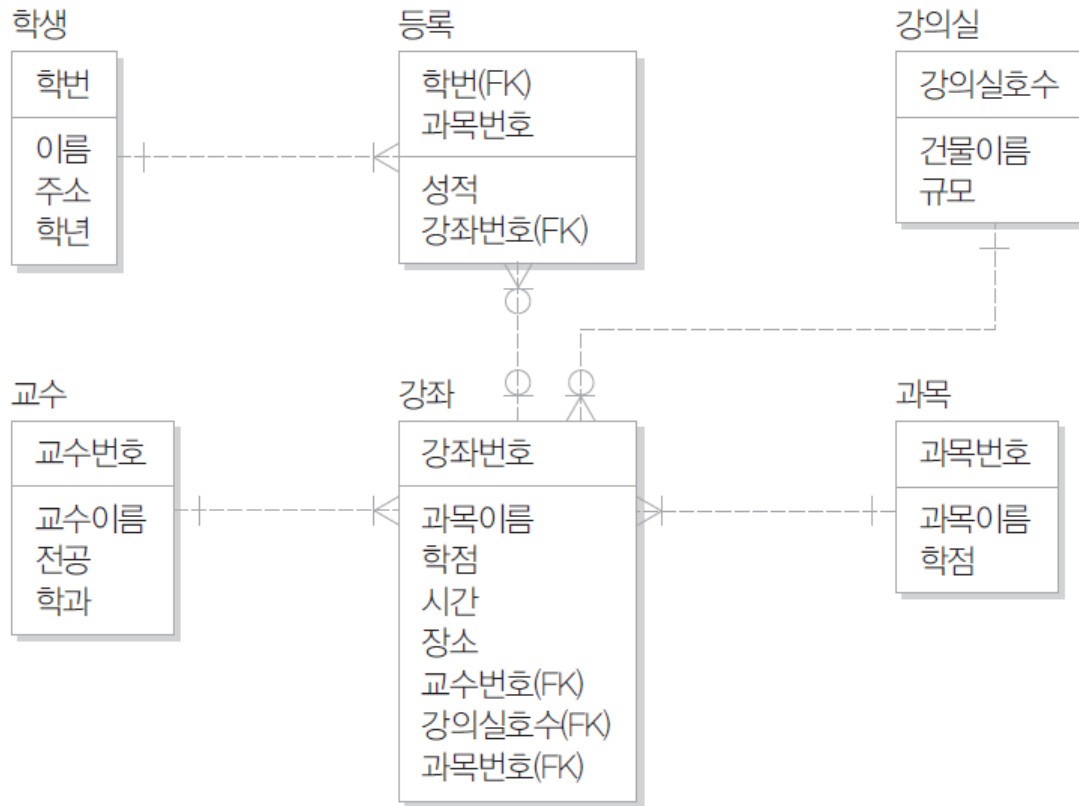


그림 1-28 수강신청 데이터베이스의 개념 스키마

5.1 3단계 데이터베이스 구조

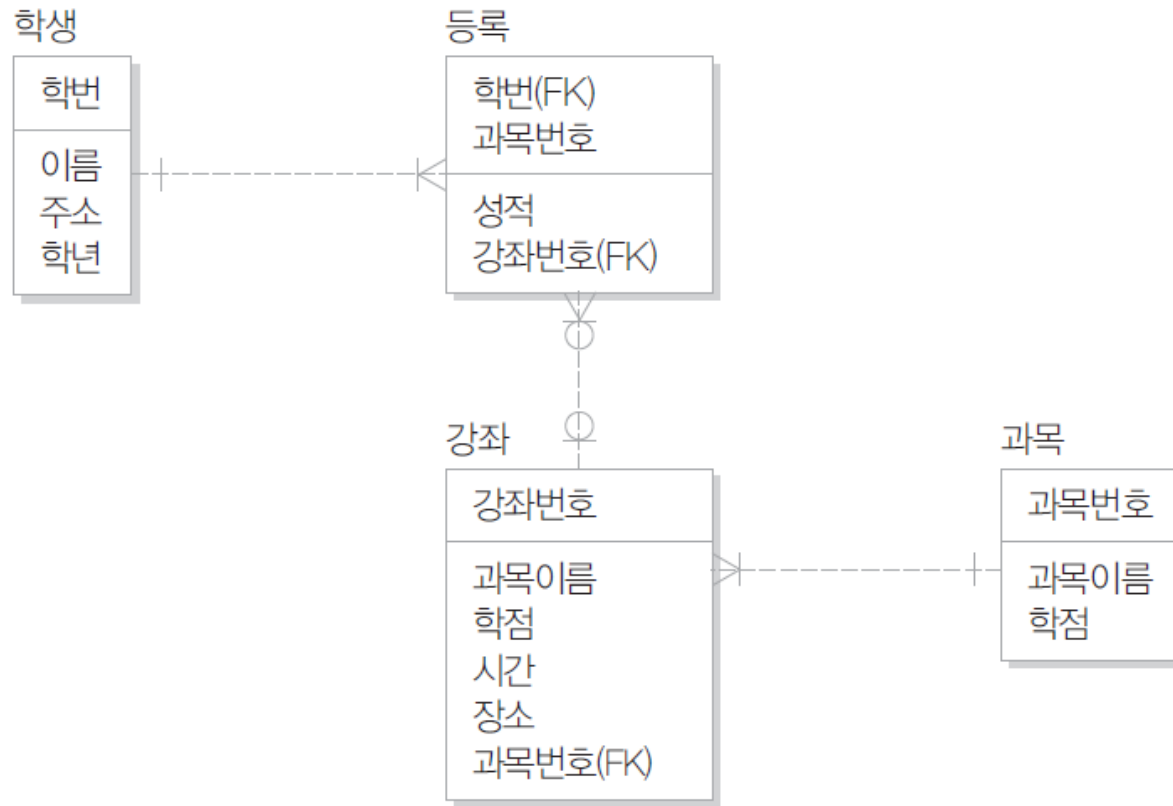


그림 1-29 수강등록 담당 부서에서 필요한 데이터베이스(외부 스키마1)

5.1 3단계 데이터베이스 구조

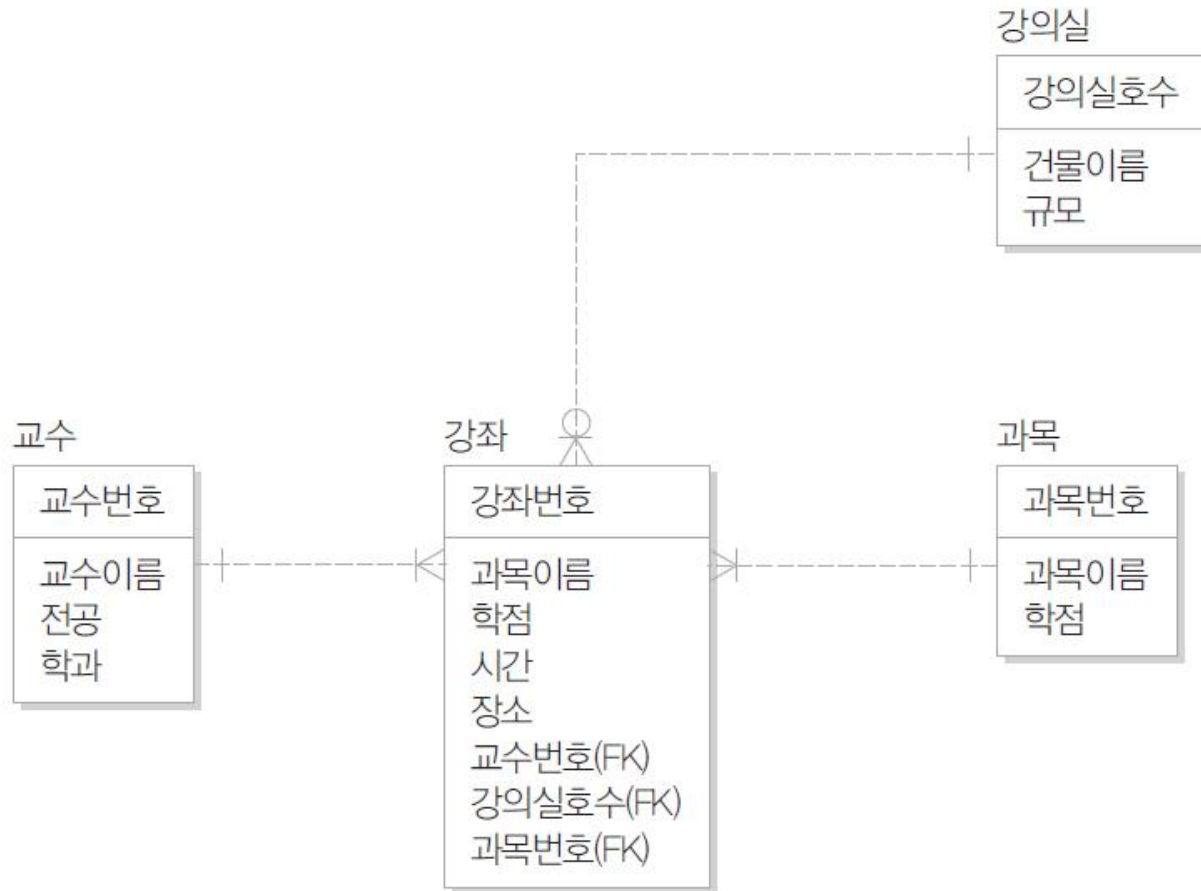


그림 1-30 시간표 담당 부서에서 필요한 데이터베이스(외부 스키마2)

5.1 3단계 데이터베이스 구조

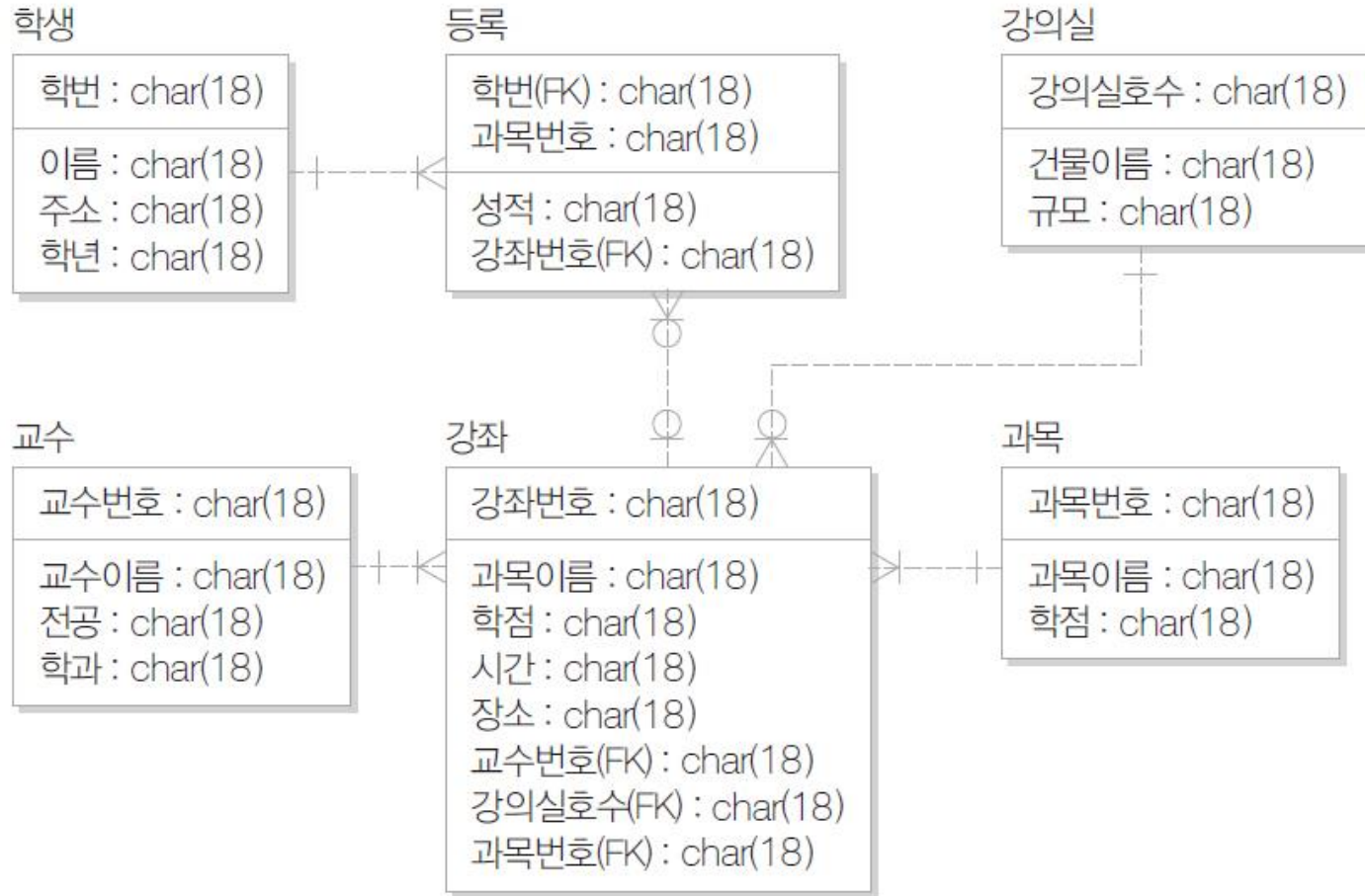


그림 1-31 수강신청 데이터베이스의 내부 스키마

5.1 3단계 데이터베이스 구조

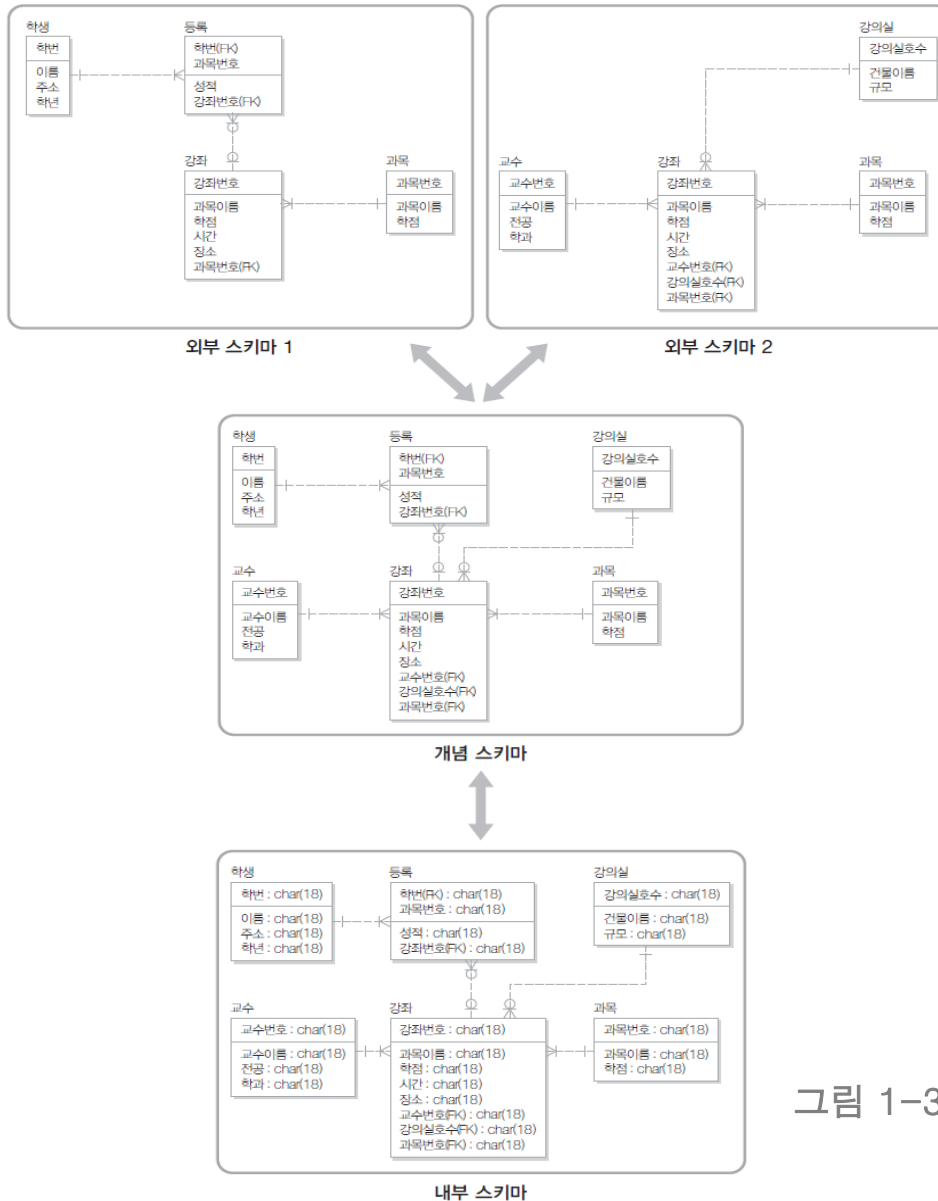


그림 1-32 수강신청 데이터베이스의 3단계 구조

■ 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

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