

# 데이터 분석을 위한 SQL

---

- 데이터베이스 기초 -



Chapter 01

# 데이터베이스 시스템

# 목차

**01**

**데이터베이스와 데이터베이스 시스템**

**02**

**데이터베이스 시스템의 발전**

**03**

**파일 시스템과 DBMS**

**04**

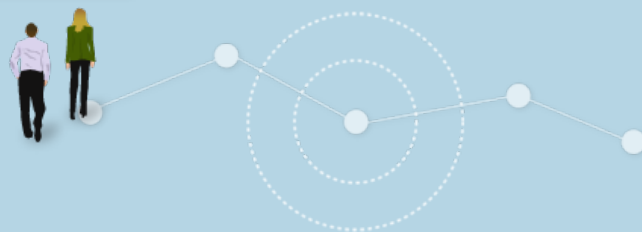
**데이터베이스 시스템의 구성**

# 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 개념 및 특징
4. 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

### ❖ 데이터베이스란?

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

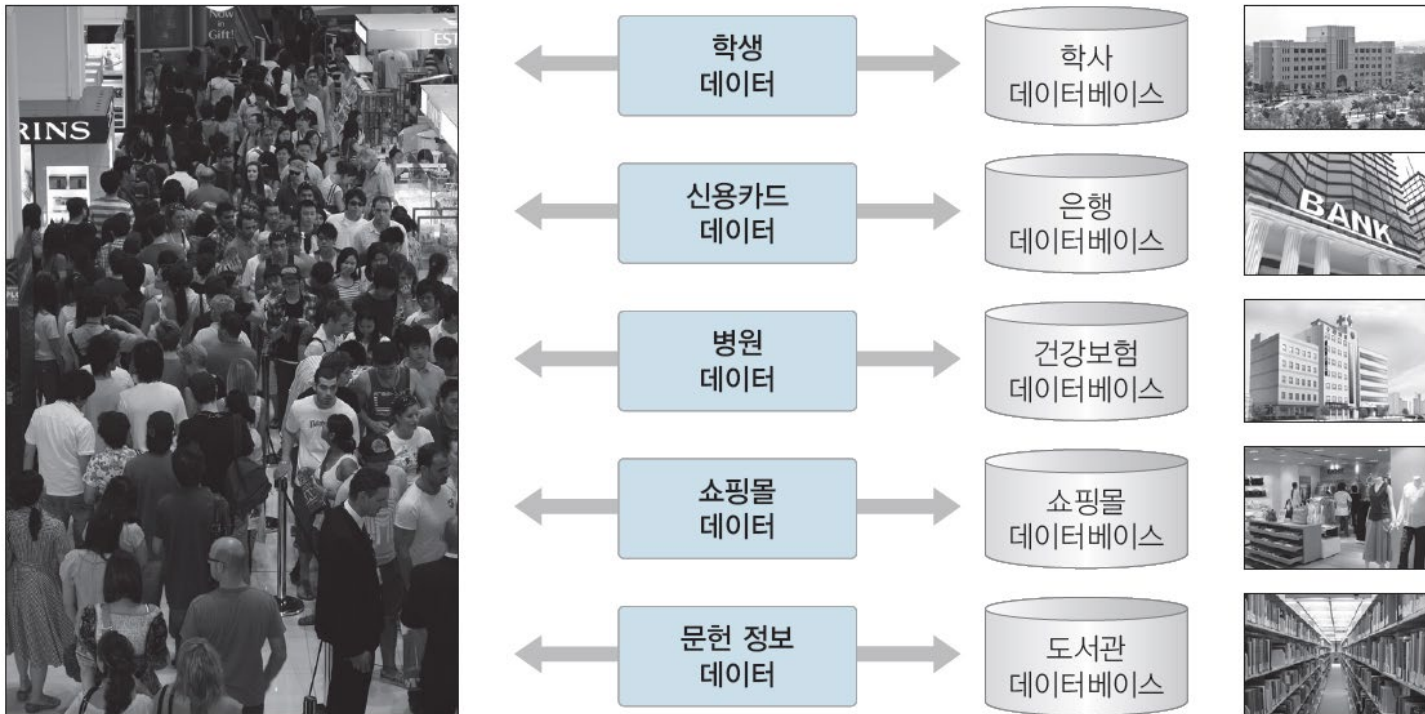


그림 1-2 일상생활에서 생성되는 데이터베이스

## 2. 일상생활의 데이터베이스

**UU Again**

\*\*\*\*\* 최근영수증발행인쇄 \*\*\*\*\*  
 CU 김릉점  
 사업자등록번호: 1751350895  
 강원도 김릉시 수촌로64 (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
<b>총 구 매 액</b>	<b>1</b>	<b>1,200</b>
과세물품가액		1,091
부 가 세		109
<b>*결 제 금 액</b>		<b>1,200</b>
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  
 객층: 14 담당: 최 회 NO: 6193 13:03

95119051933177093193

**KORAIL**

**열차승차권**  
 Train Ticket

승차일자  
 2019년 09월 22일 (금/Fri)

출발 From  
 송주  
 Yongsan

도착 To  
 용산  
 Yongsan

16:39 → 18:33

\*승차일자와 이용구간을 반드시 확인하시기 바랍니다.

Train No.  
**4188 열차 KTX 산천**

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

영수액₩ 42,100원

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

발행일자 03/5 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"><li>• 기상정보 : 날씨 관련 정보를 제공</li><li>• 교통정보 : 교통상황 관련 정보를 제공</li><li>• 문화예술정보 : 공연이나 인물에 관한 정보를 제공</li></ul>
비즈니스	<ul style="list-style-type: none"><li>• 금융정보 : 금융, 증권, 신용에 관한 정보를 제공</li><li>• 취업정보 : 노동부와 기업의 채용 관련 정보를 제공</li><li>• 부동산정보 : 공공기관이나 민간의 토지, 매물, 세금 정보를 제공</li></ul>
학술정보	<ul style="list-style-type: none"><li>• 연구학술정보 : 논문, 서적, 저작물에 관한 정보를 제공</li><li>• 특허정보 : 특허청의 정보를 기업과 연구자에게 제공</li><li>• 법률정보 : 법제처와 대법원의 법률에 관한 정보를 제공</li><li>• 통계정보 : 국가기관의 통계에 관한 정보를 제공</li></ul>

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

구축이  
쉬움



구축이  
어려움

### 3. 데이터베이스의 개념 및 특징

#### ❖ 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 데이터, 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 및 특징

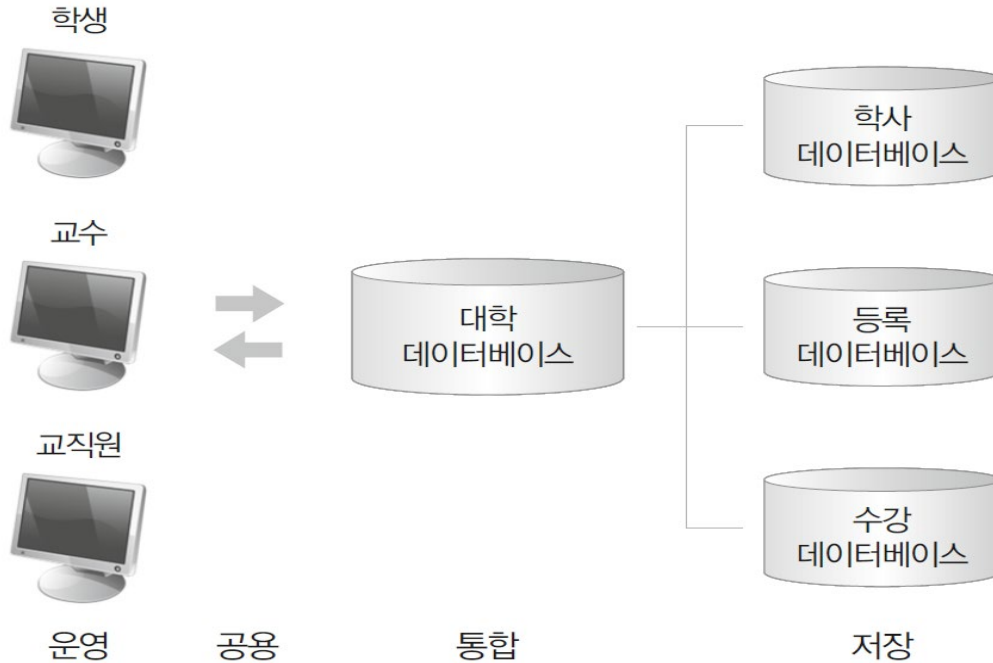


그림 1-4 데이터베이스의 개념 : 데이터베이스는 운영 데이터를 통합하여 저장하며 공용으로 사용된다

### 3. 데이터베이스의 개념 및 특징

#### ❖ 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 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.

##### ③ 동시 공유(concurrent sharing)

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

##### ④ 내용에 따른 참조(reference by content)

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

## 4. 데이터베이스 시스템의 구성

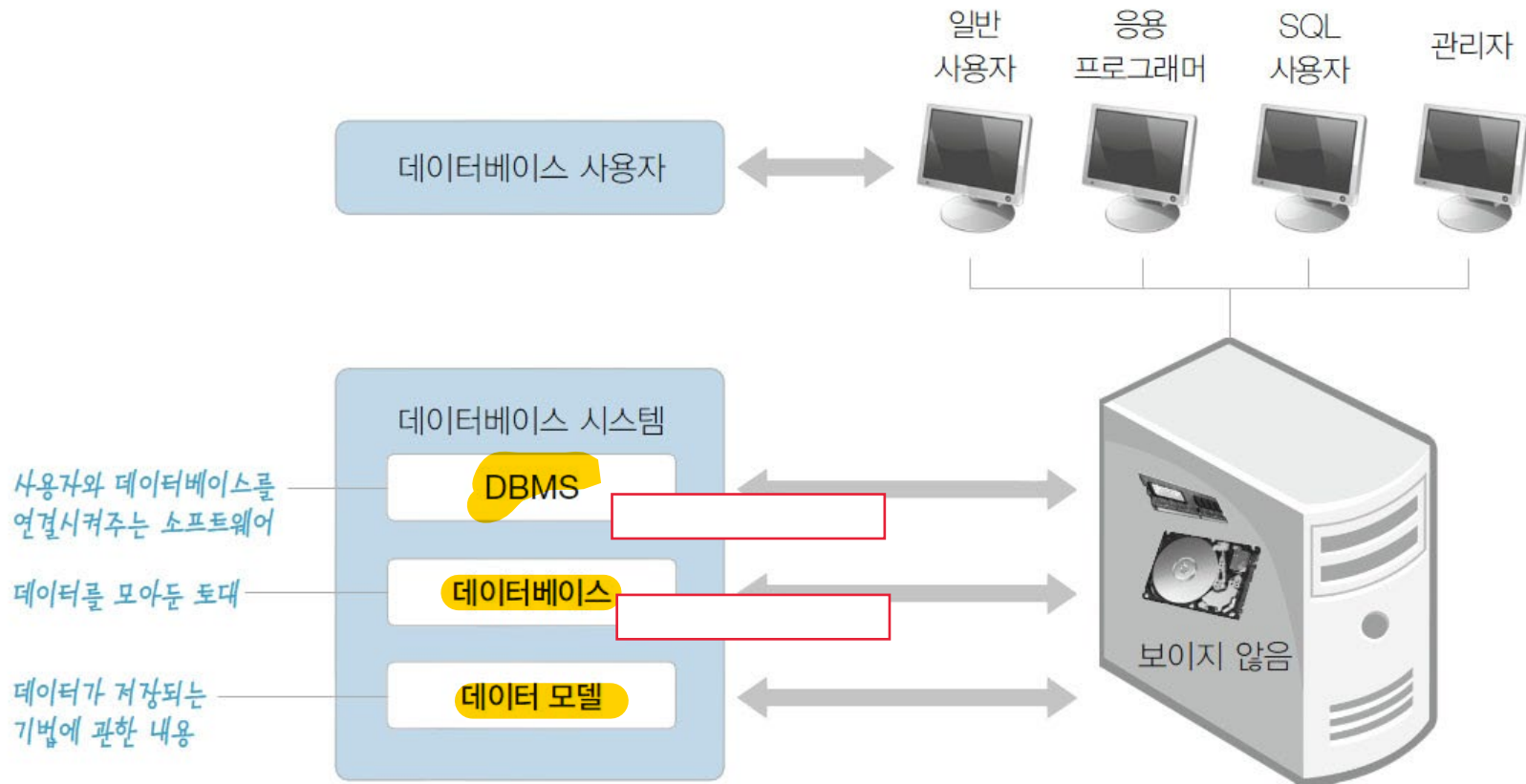
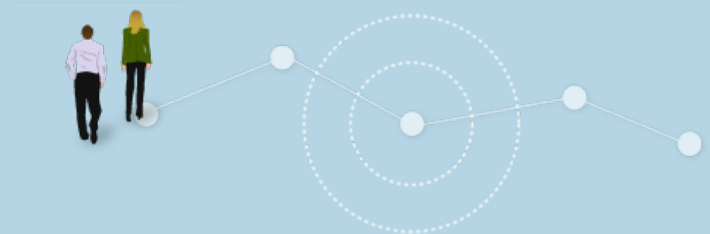


그림 1-5 데이터베이스 시스템의 구성 요소와 물리적인 위치

## 02 데이터베이스 시스템의 발전

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

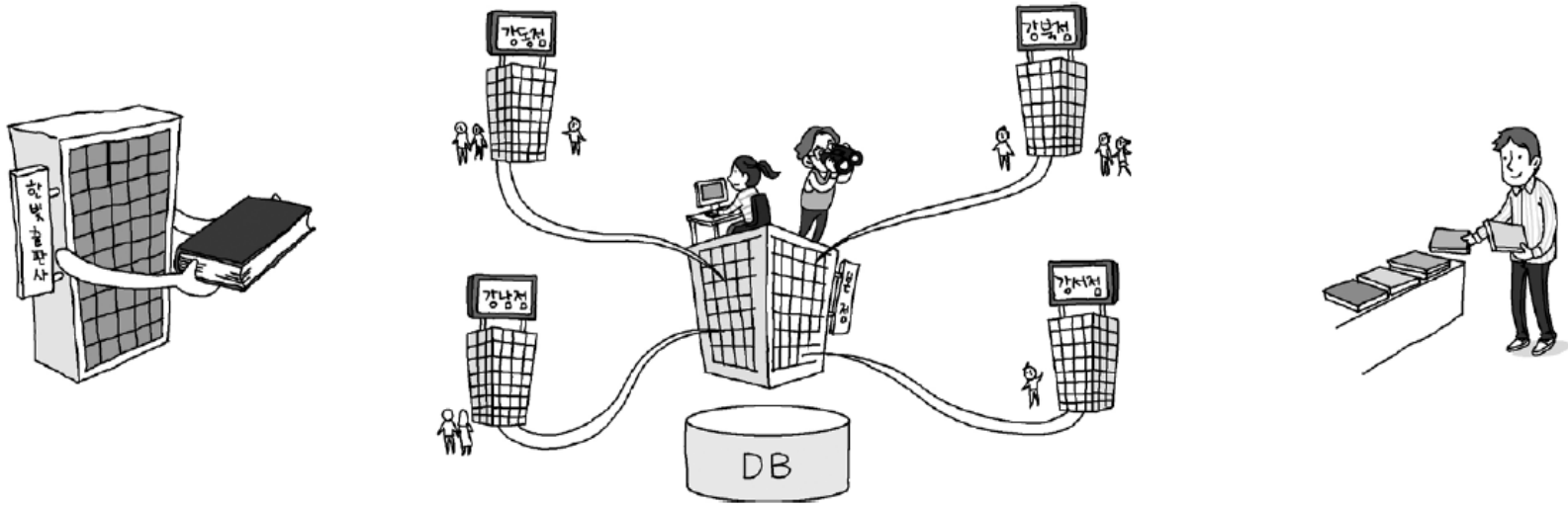


- 도서 : 1,000권
- 고객 : 근처 학교의 학생, 지역 주민
- 업무 : 회계 업무(컴퓨터 사용), 파일 시스템
- 고객 서비스 : 컴퓨터를 이용하여 도서 검색, 직원 고용

그림 1-7 마당서점 전산화

# 1. 데이터베이스 시스템의 예 : 마당서점

## ❖ [3단계] 지점 개설 및 데이터베이스 구축

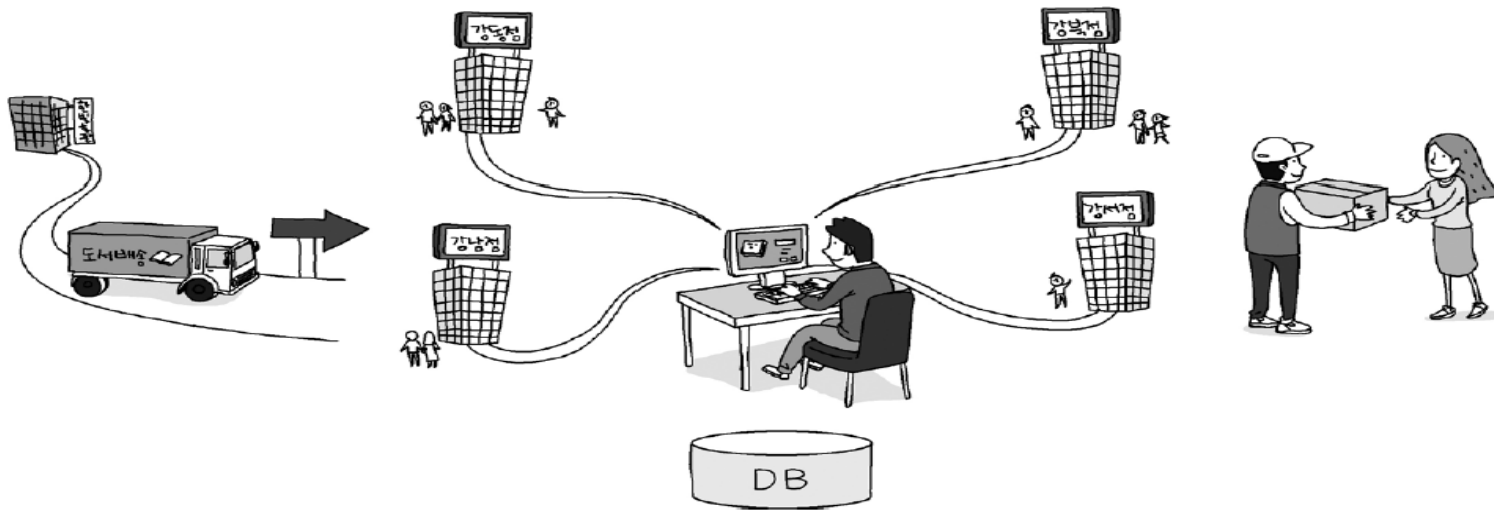


- 도서 : 10,000권
- 고객 : 서울 지역 고객
- 업무 : 회계 업무(컴퓨터 사용), 데이터베이스 시스템
- 고객 서비스 : 클라이언트/서버 시스템으로 지점을 연결하여 도서 검색 서비스 제공

그림 1-8 마당서점 DBMS 도입

# 1. 데이터베이스 시스템의 예 : 마당서점

## ❖ [4단계] 홈페이지 구축

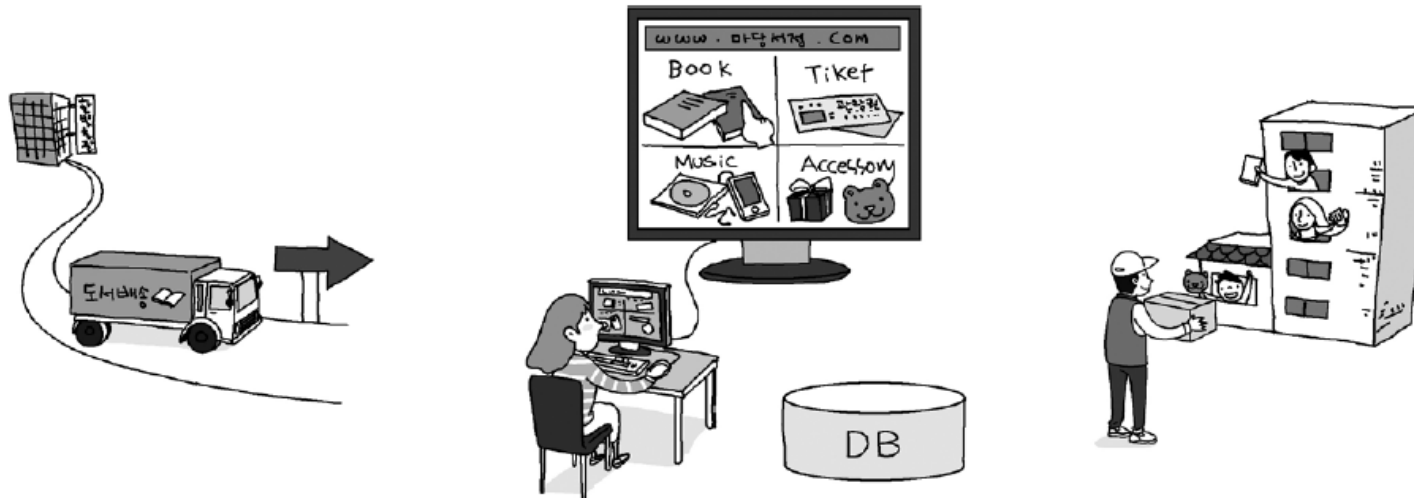


- 도서 : 100,000권
- 고객 : 국민(전국으로 배송)
- 업무 : 회계/인사 업무(컴퓨터와 인터넷 사용), 웹 DB 시스템으로 지점 간 연계
- 고객 서비스 : 인터넷으로 도서 검색 및 주문

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

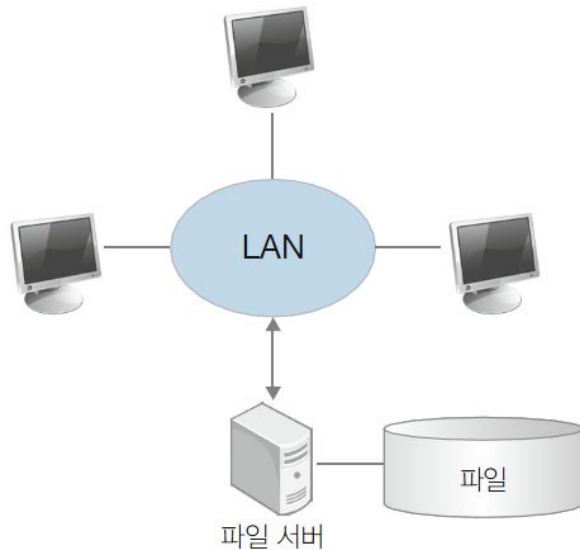


그림 1-11 파일 시스템

## 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

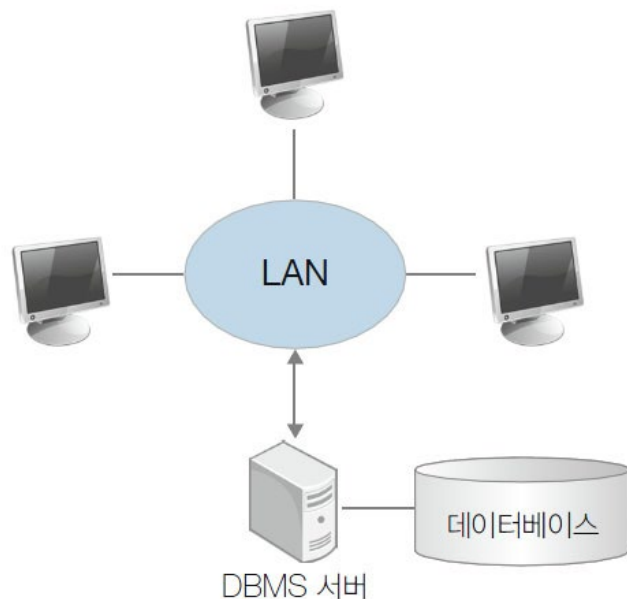


그림 1-12 데이터베이스 시스템

## 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

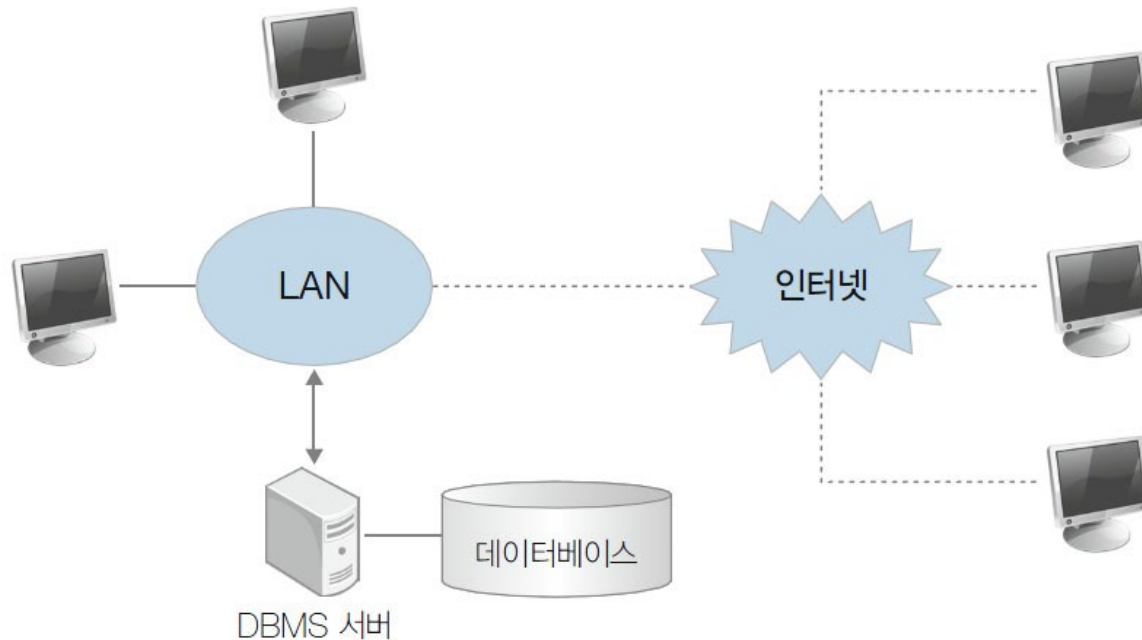


그림 1-13 웹 데이터베이스 시스템



## 2. 정보 시스템의 발전

### ④ 분산 데이터베이스 시스템

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

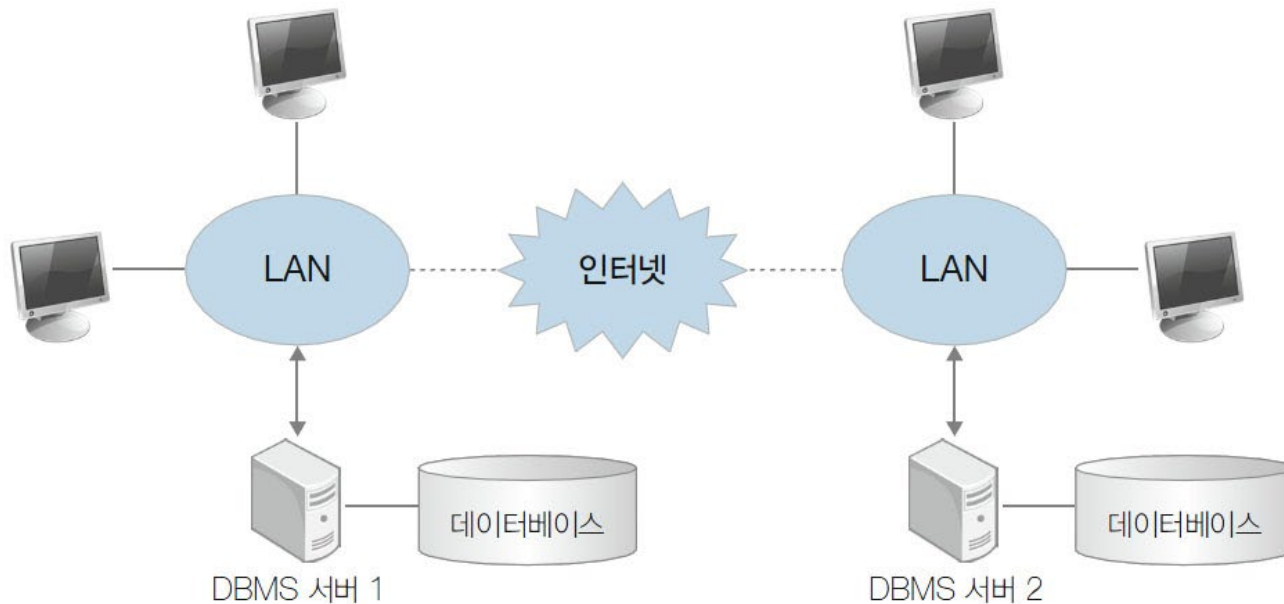


그림 1-14 분산 데이터베이스 시스템

## 2. 정보 시스템의 발전

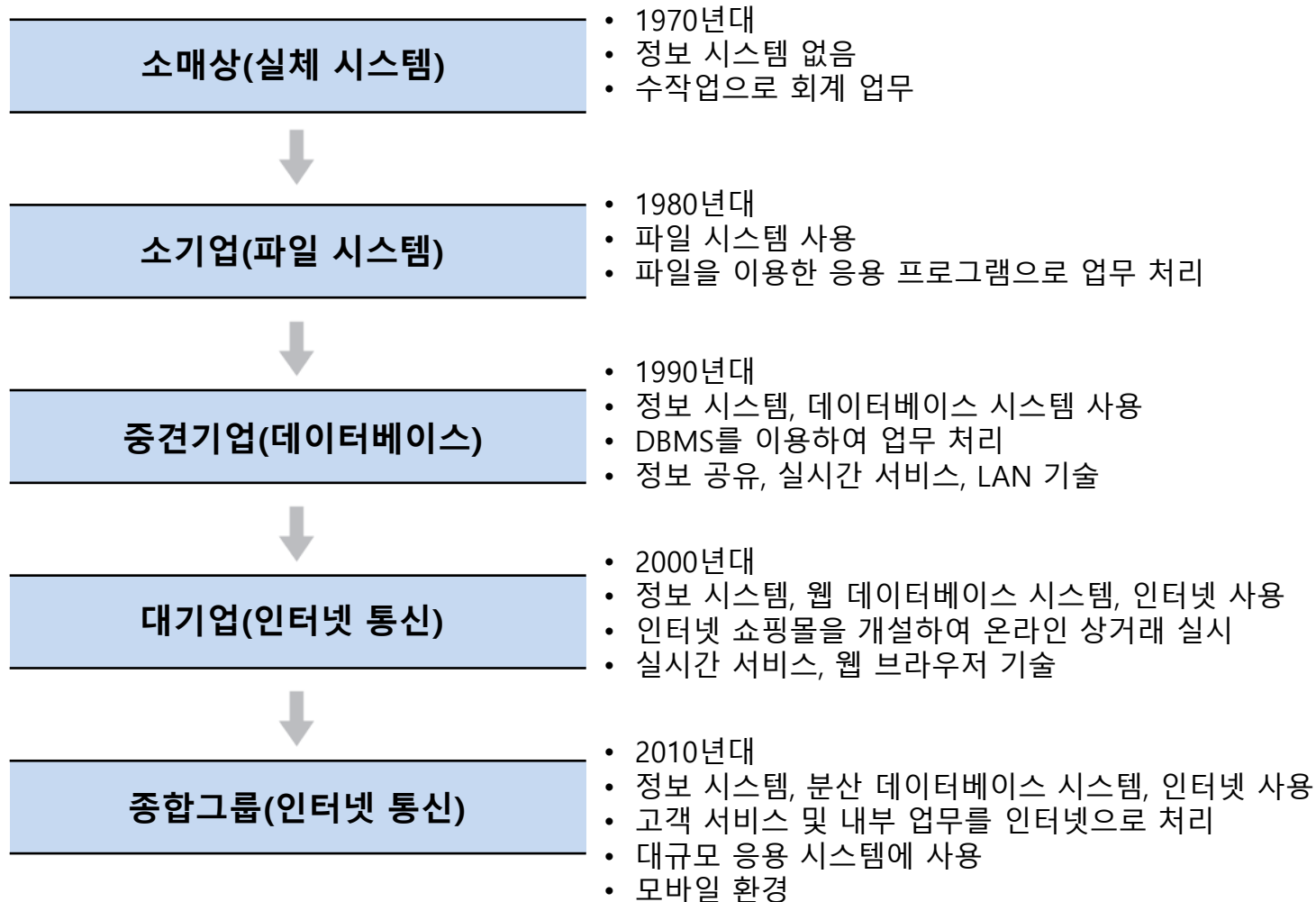
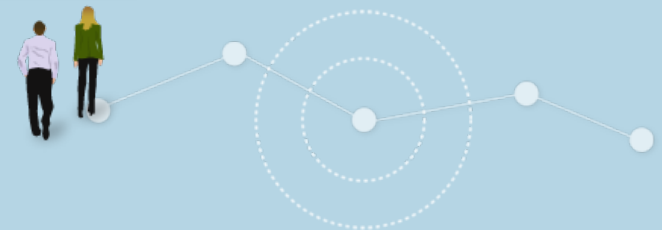


그림 1-15 정보 시스템의 발전과 기업의 업무 환경 변화

## 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

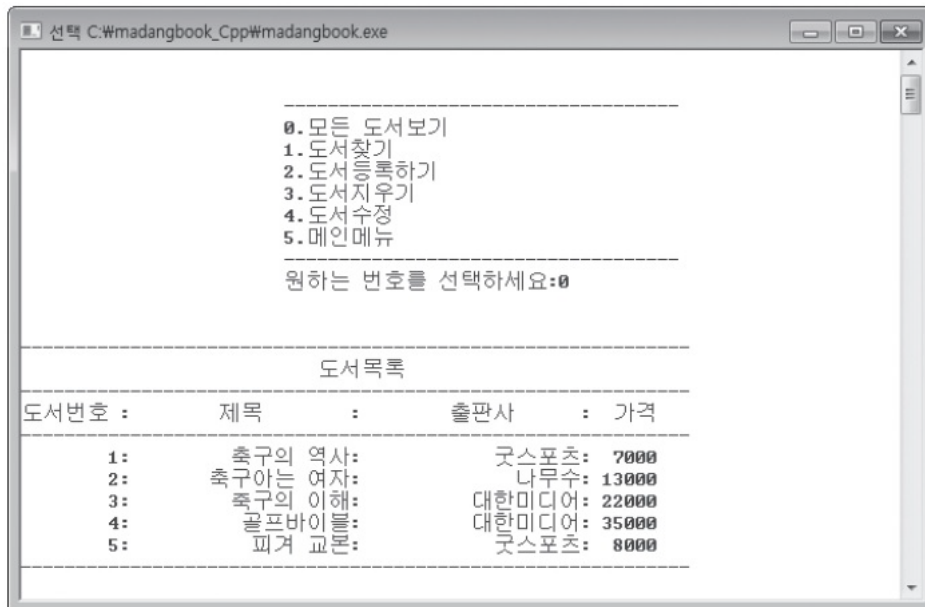


그림 1-17 도서 검색 프로그램

# 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

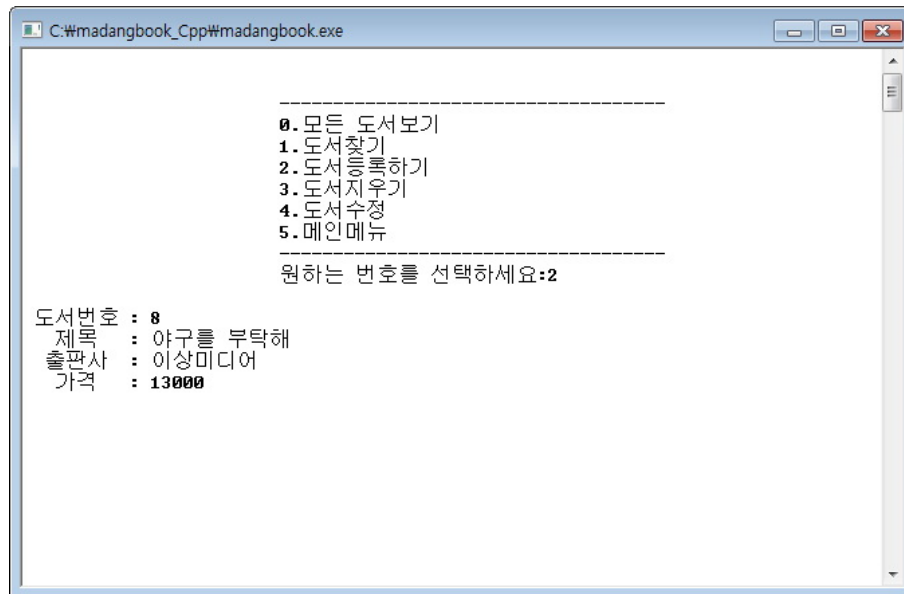


그림 1-18 도서 검색 프로그램에서 도서를 등록하는 화면

# 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

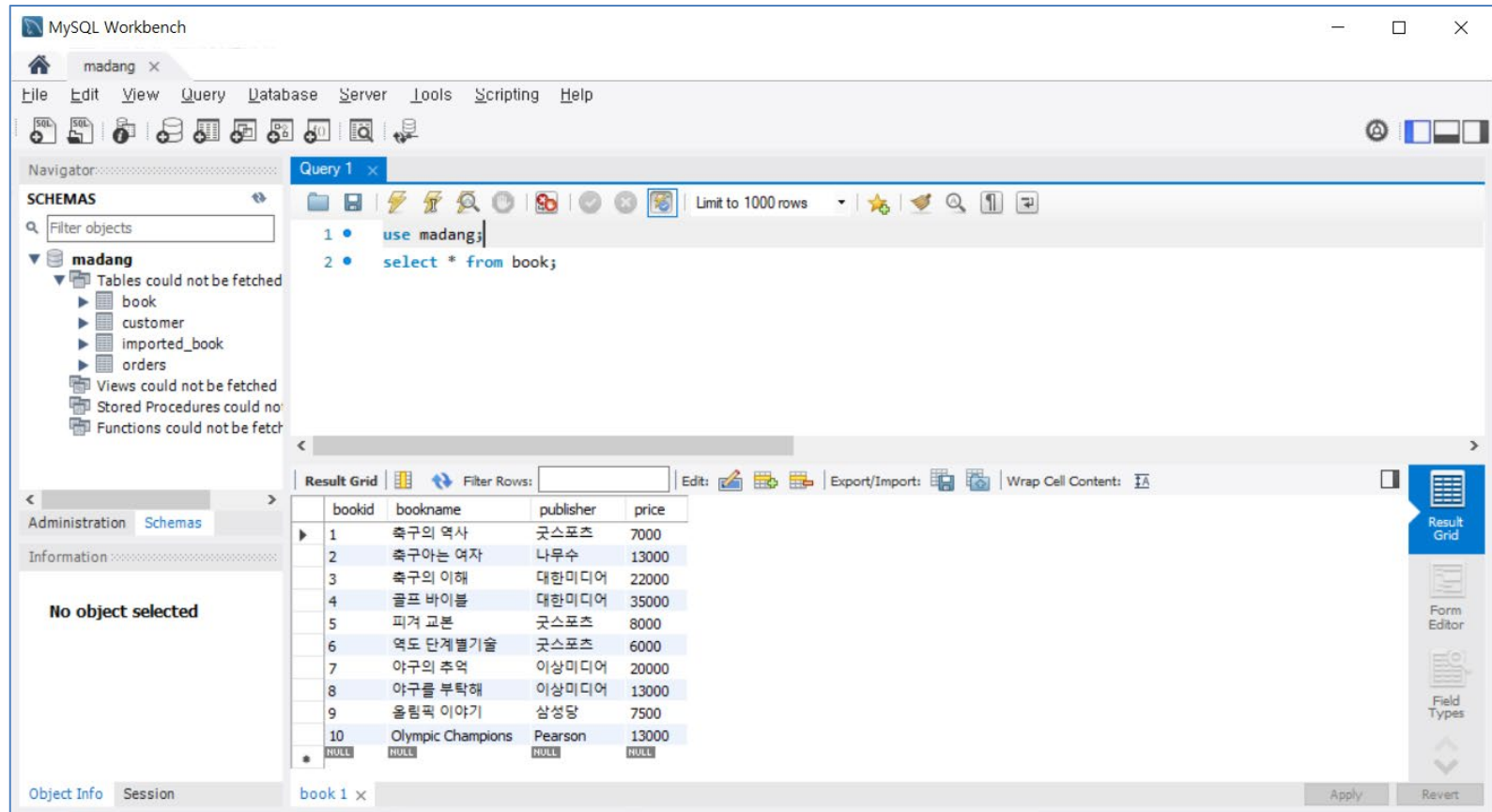


그림 1-19 MySQL Workbench의 데이터베이스 관리 화면

# 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;

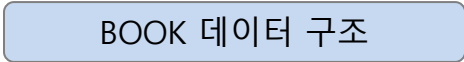
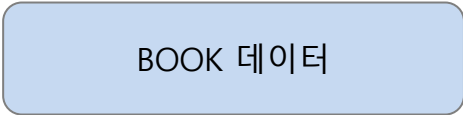
    /* 모든 도서보기 프로그램 호출 */
    search_all( );

    /* 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

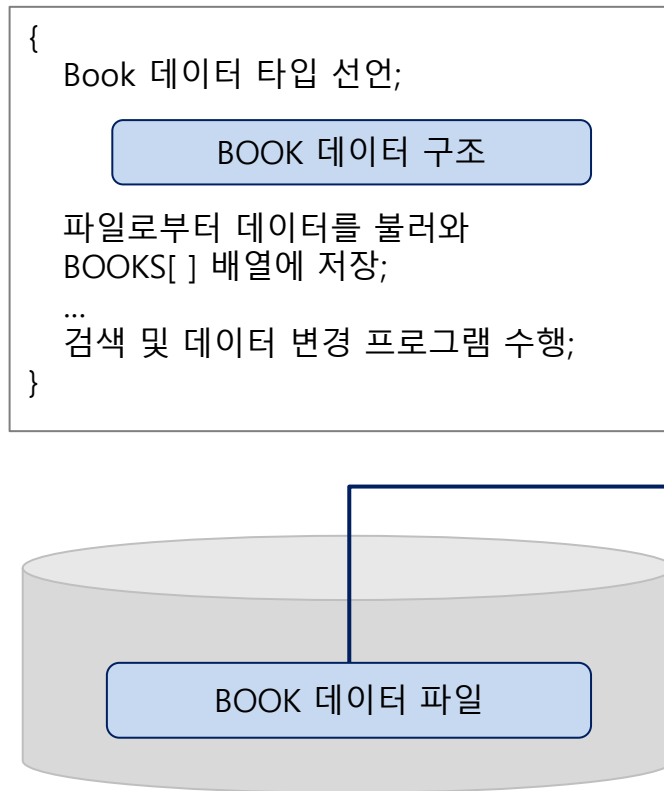
### [프로그램 1] 구조

```
{  
  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



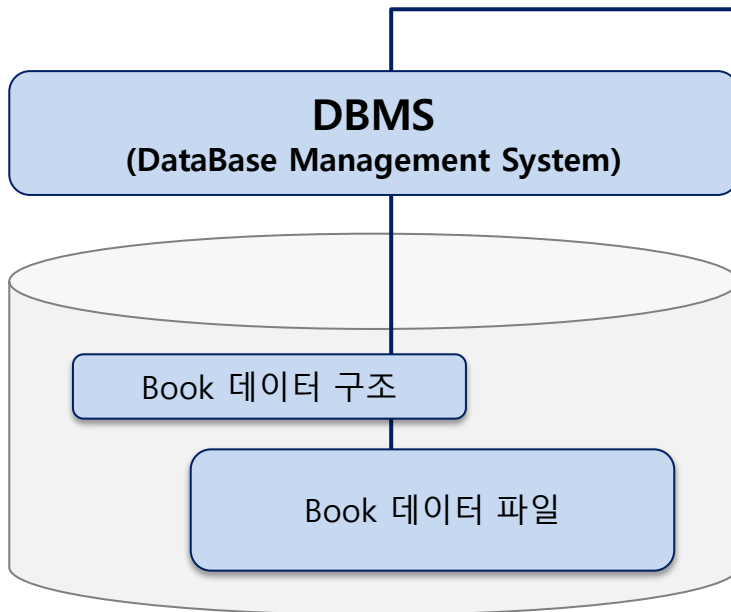
- 파일에 데이터 값, 프로그램에 데이터 정의를 포함하는 방식
- 프로그램에 BOOK 데이터 구조만 정의하고, 데이터 값은 book.dat라는 파일에 저장됨
- 데이터 값이 바뀌면 프로그램에 변경이 없지만, 데이터 구조가 바뀌면 프로그램을 다시 컴파일해야 함

## 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

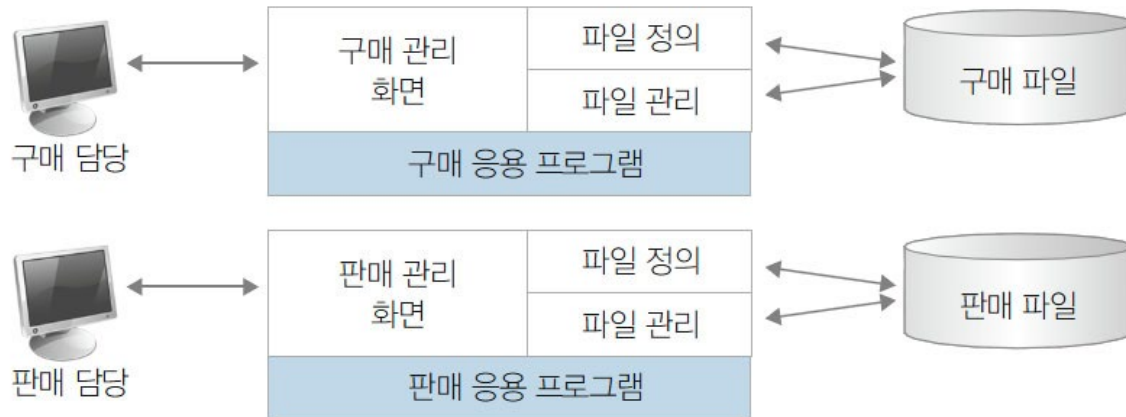


그림 1-20 파일 시스템으로 구축된 구매 및 판매 응용 프로그램

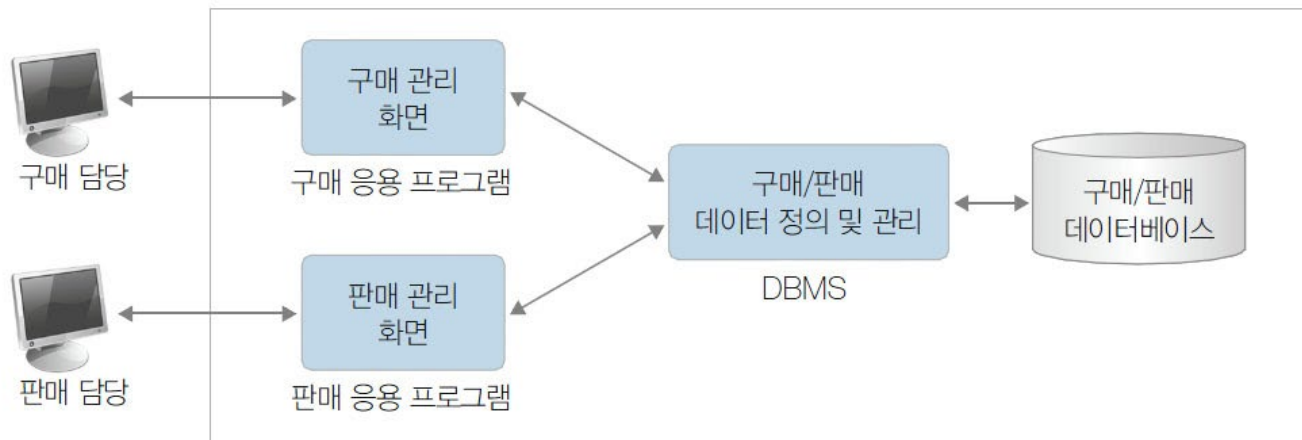


그림 1-21 DBMS로 구축된 구매 및 판매 응용 프로그램



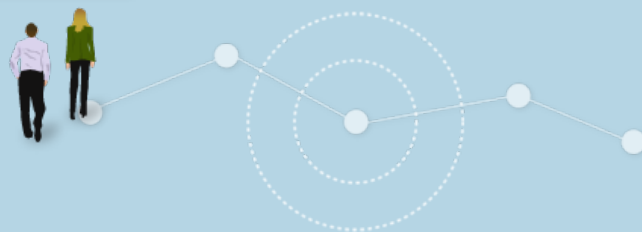
### 3. 파일 시스템과 DBMS의 비교

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>

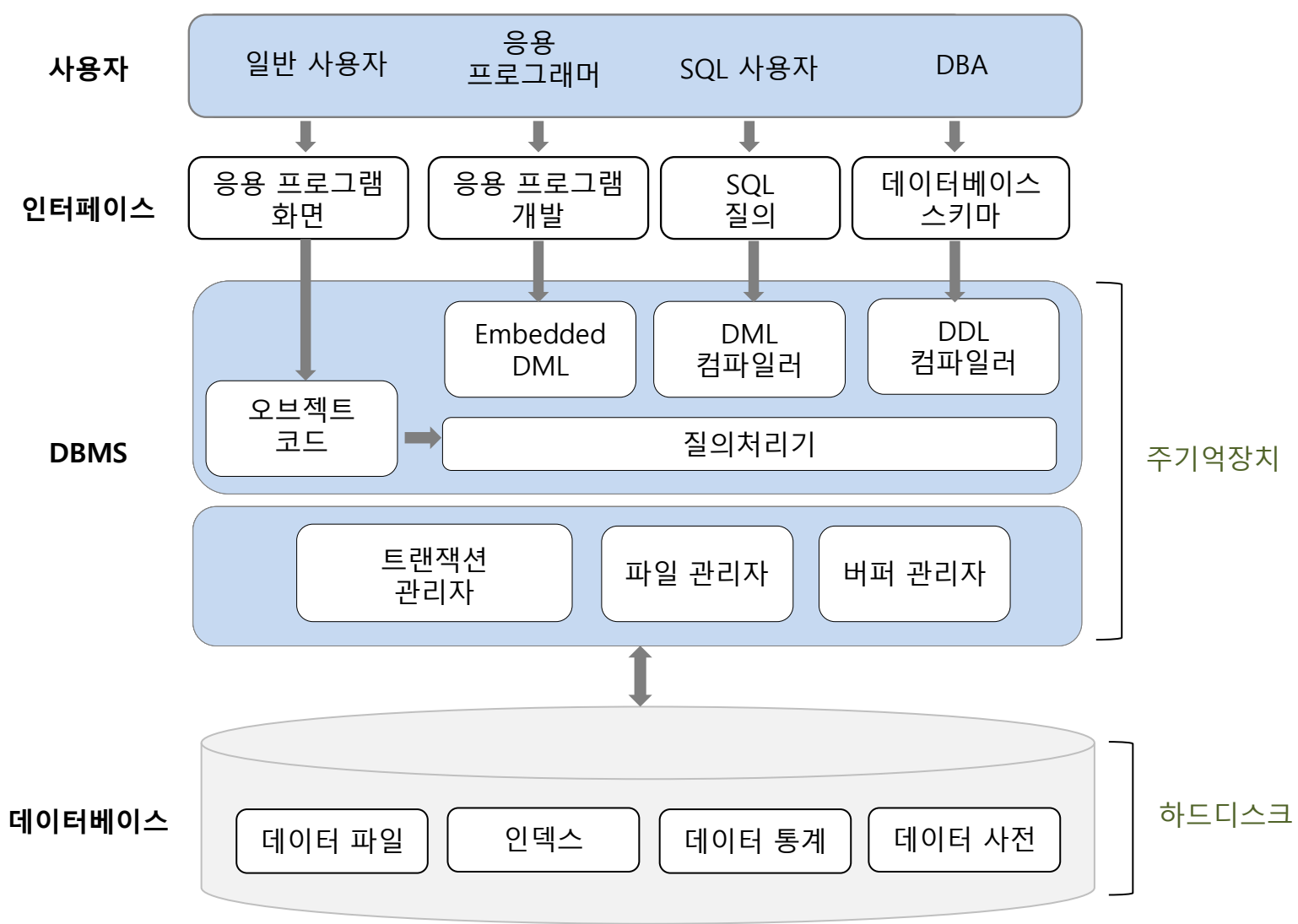


그림 1-22 데이터베이스 시스템의 구성

# 1. 데이터베이스 언어

## ■ SQL

- 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;
```



bookname	publisher
축구아는 여자	나무수
축구의 이해	대한미디어
골프 바이블	대한미디어

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

## 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



## 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

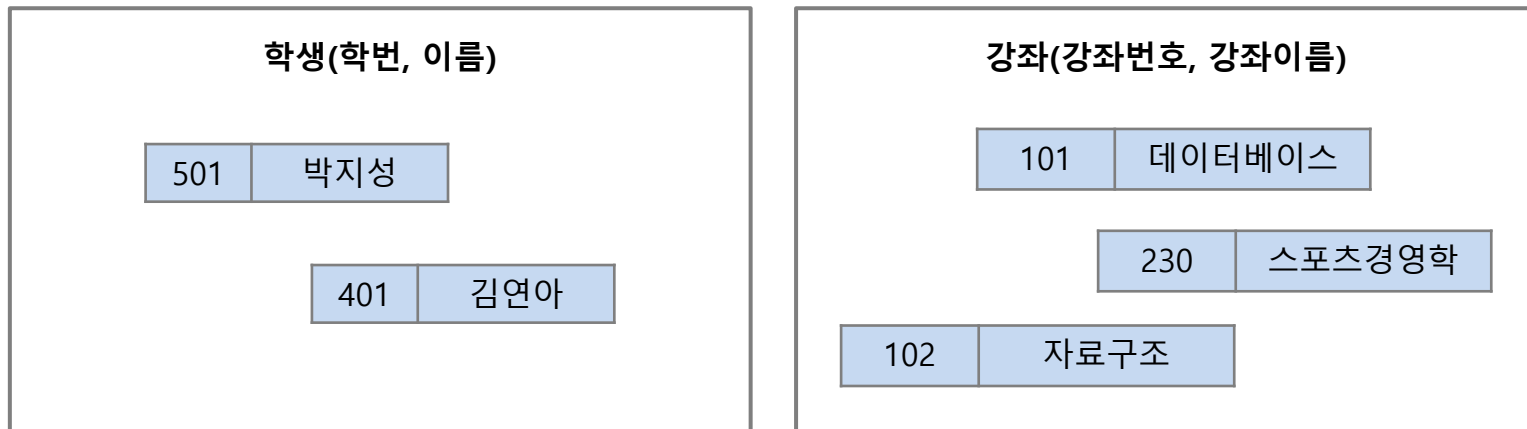


그림 1-23 관계 표현을 위한 예시

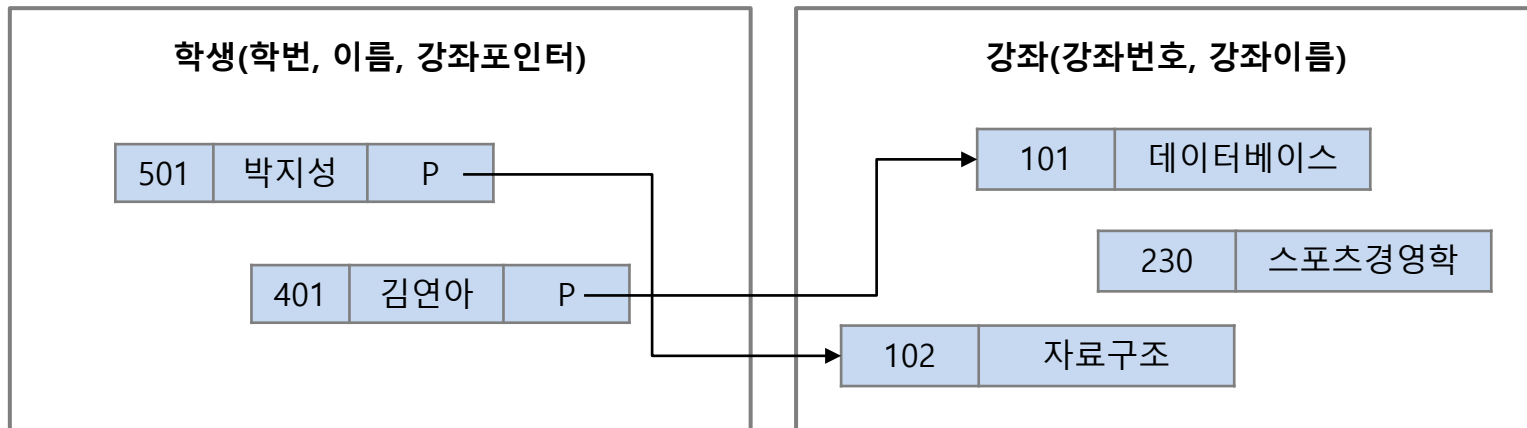


그림 1-24 포인터를 사용하여 관계 표현

## 4. 데이터 모델

### ② 속성 값 사용 : 관계 데이터 모델

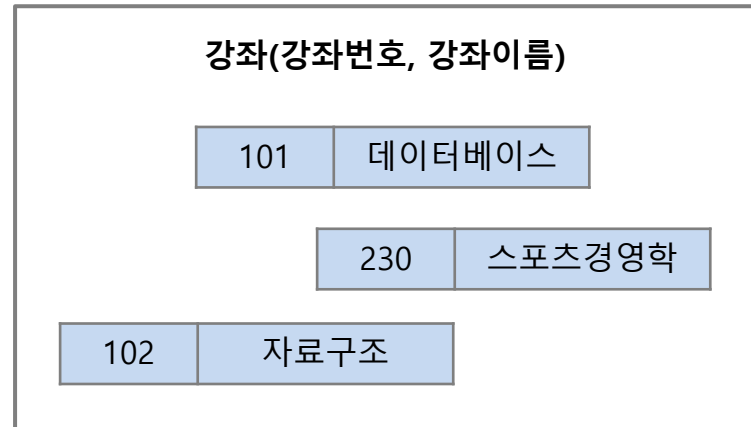
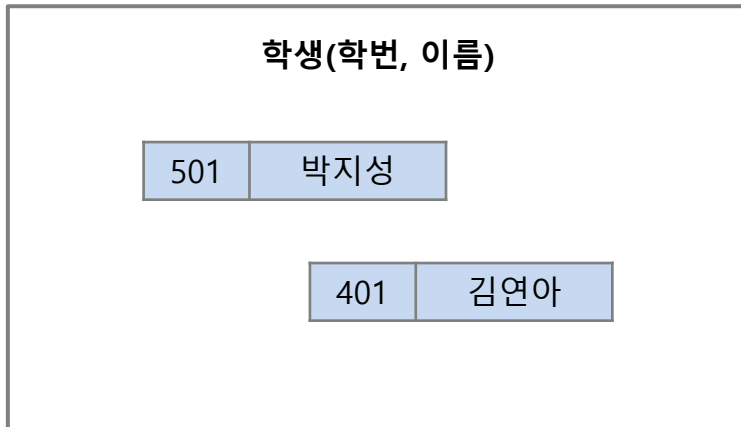


그림 1-23 관계 표현을 위한 예시

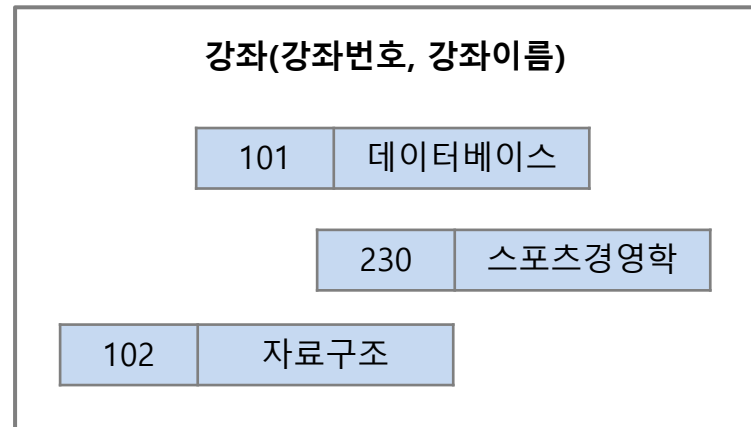
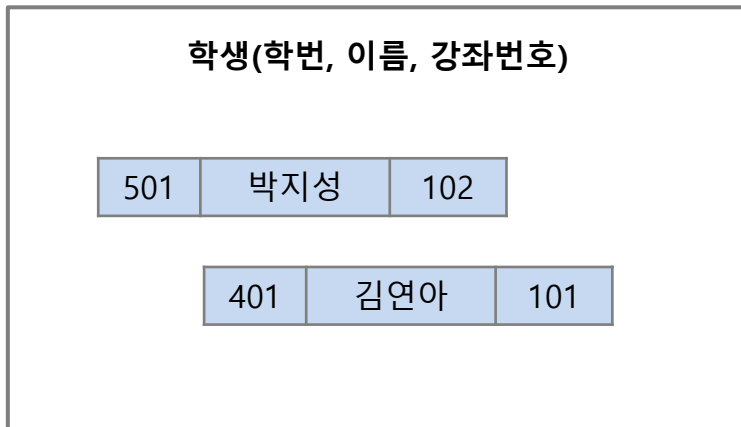


그림 1-25 속성 값을 사용하여 관계 표현

## 4. 데이터 모델

### ㉓ 객체식별자 사용 : 객체 데이터 모델

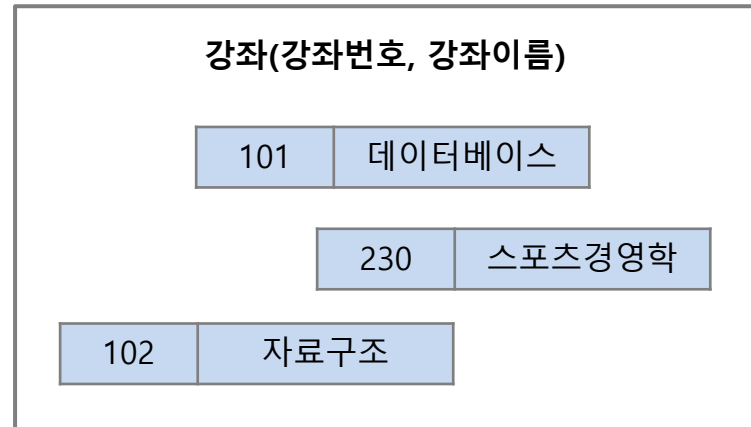
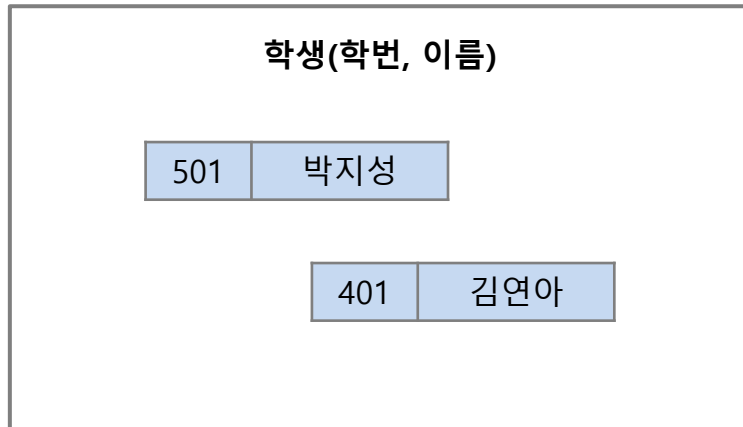


그림 1-23 관계 표현을 위한 예시

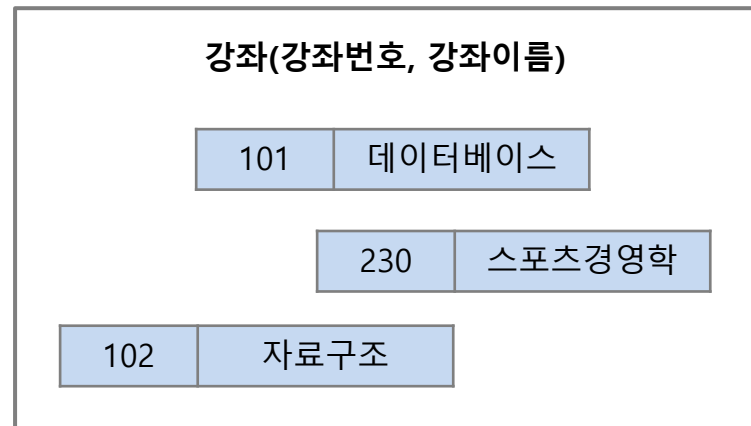
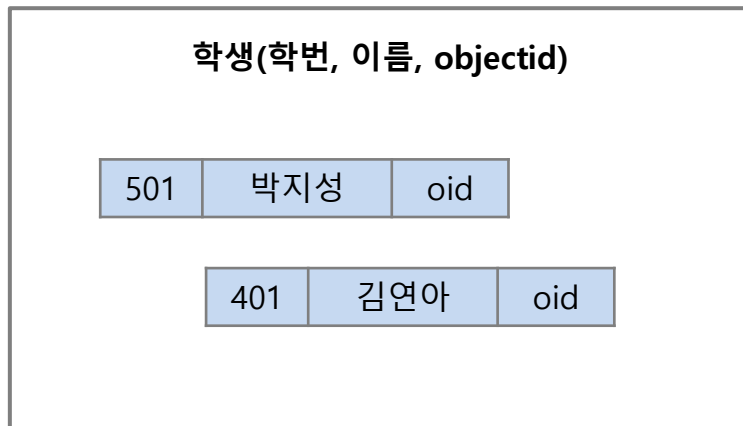


그림 1-26 객체식별자를 사용하여 관계 표현

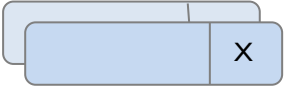
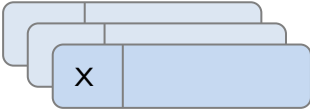
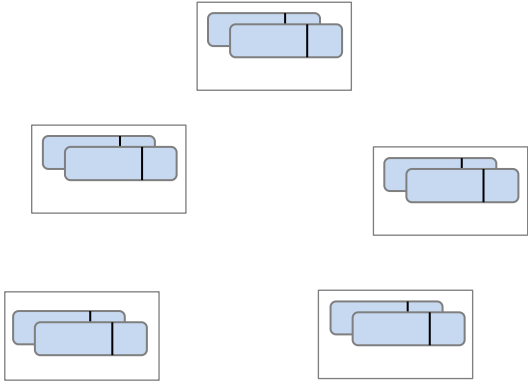
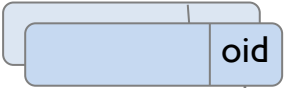
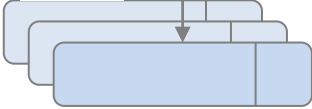
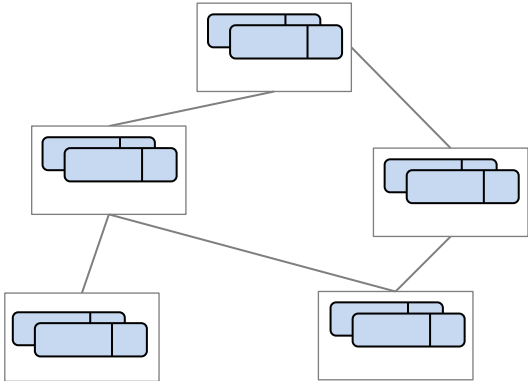
# 4. 데이터 모델

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>↓</div> <div>강좌</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>↓</div> <div>강좌</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)	<div>학생</div>  <div>강좌</div> 	
객체 데이터 모델 (객체식별자 사용)	<div>학생</div>  <div>강좌</div> 	

## 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

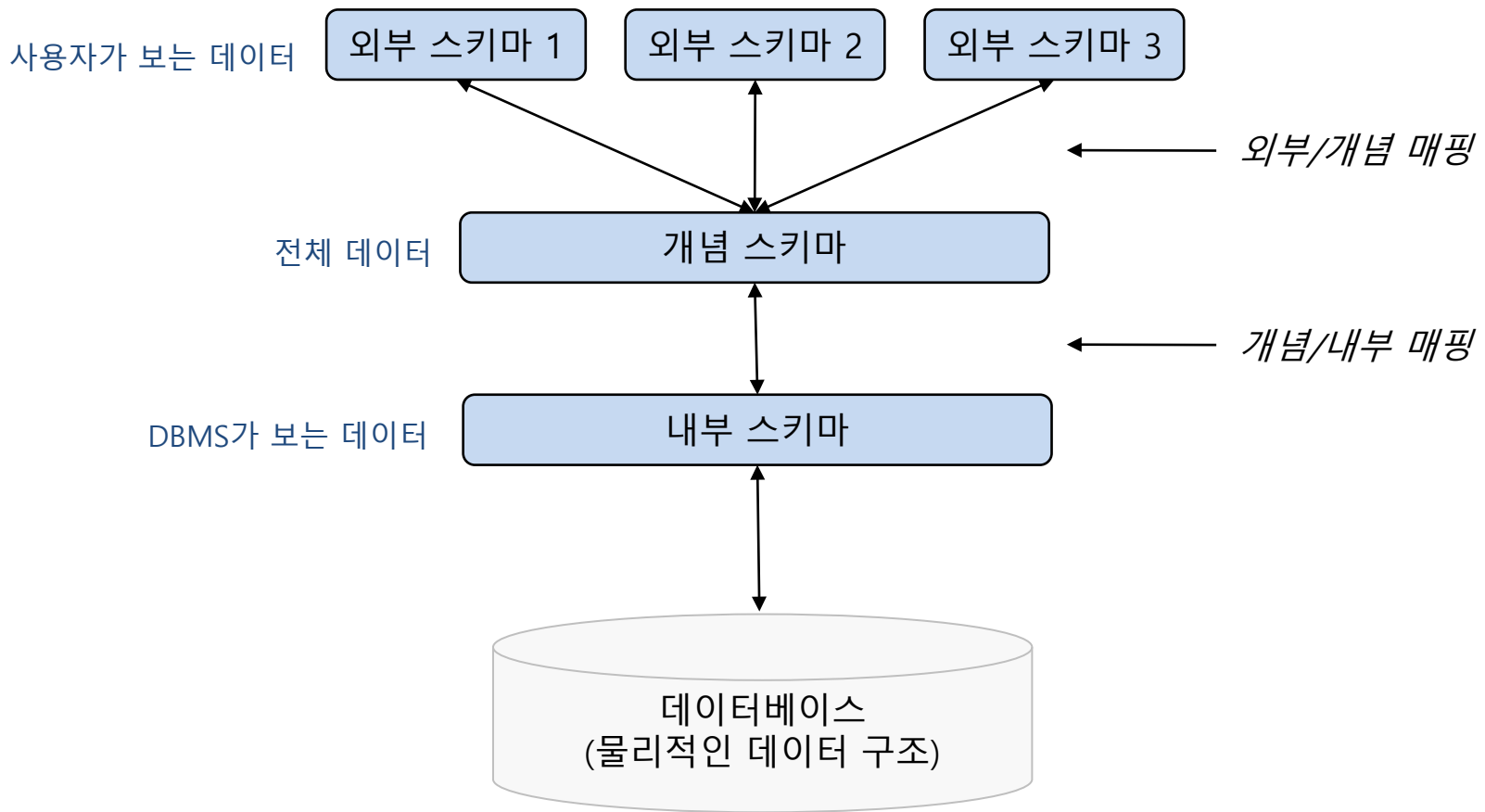


그림 1-27 ANSI의 3단계 데이터베이스 구조



## 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

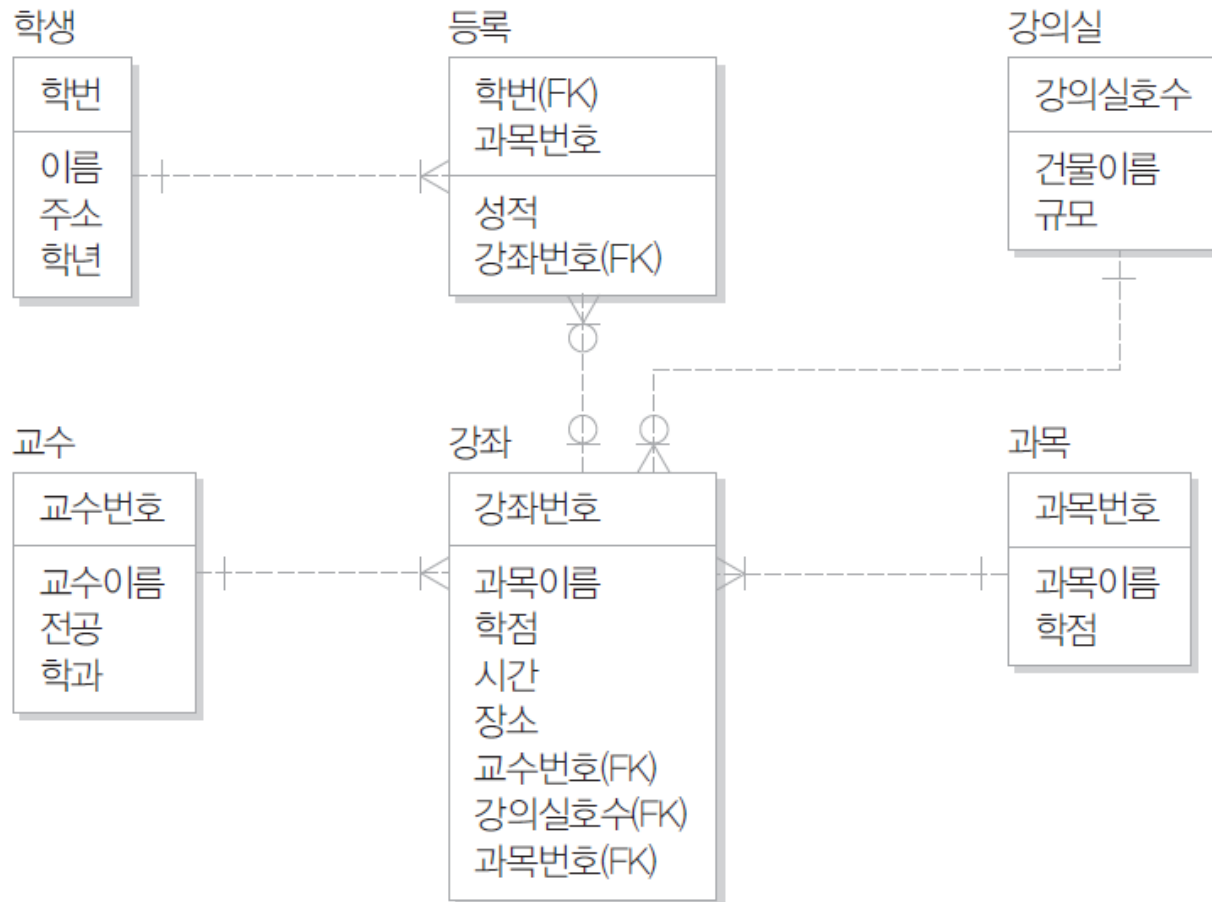


그림 1-28 수강신청 데이터베이스의 개념 스키마

## 5. 데이터베이스의 개념적 구조



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

## 5. 데이터베이스의 개념적 구조

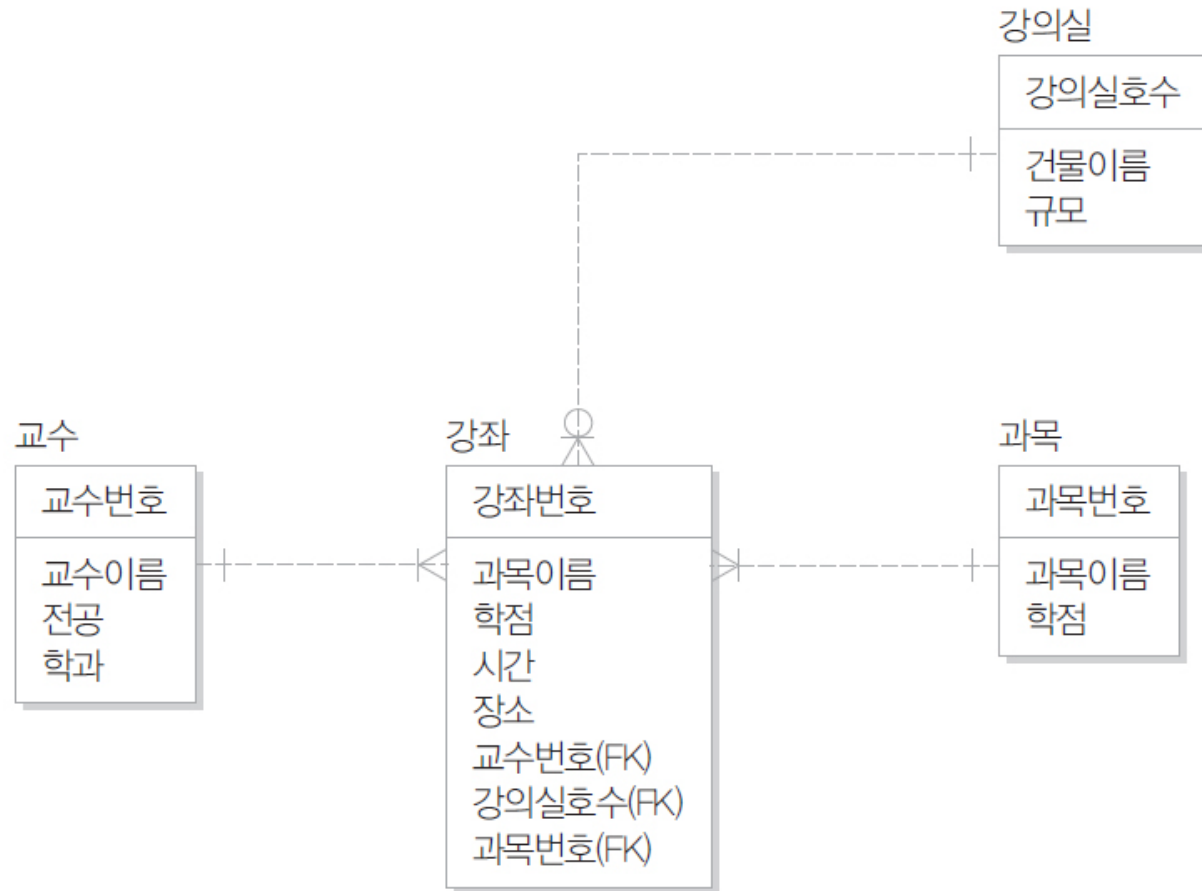


그림 1-30 시간표 담당 부서에서 필요한 데이터베이스(외부 스키마2)

## 5. 데이터베이스의 개념적 구조

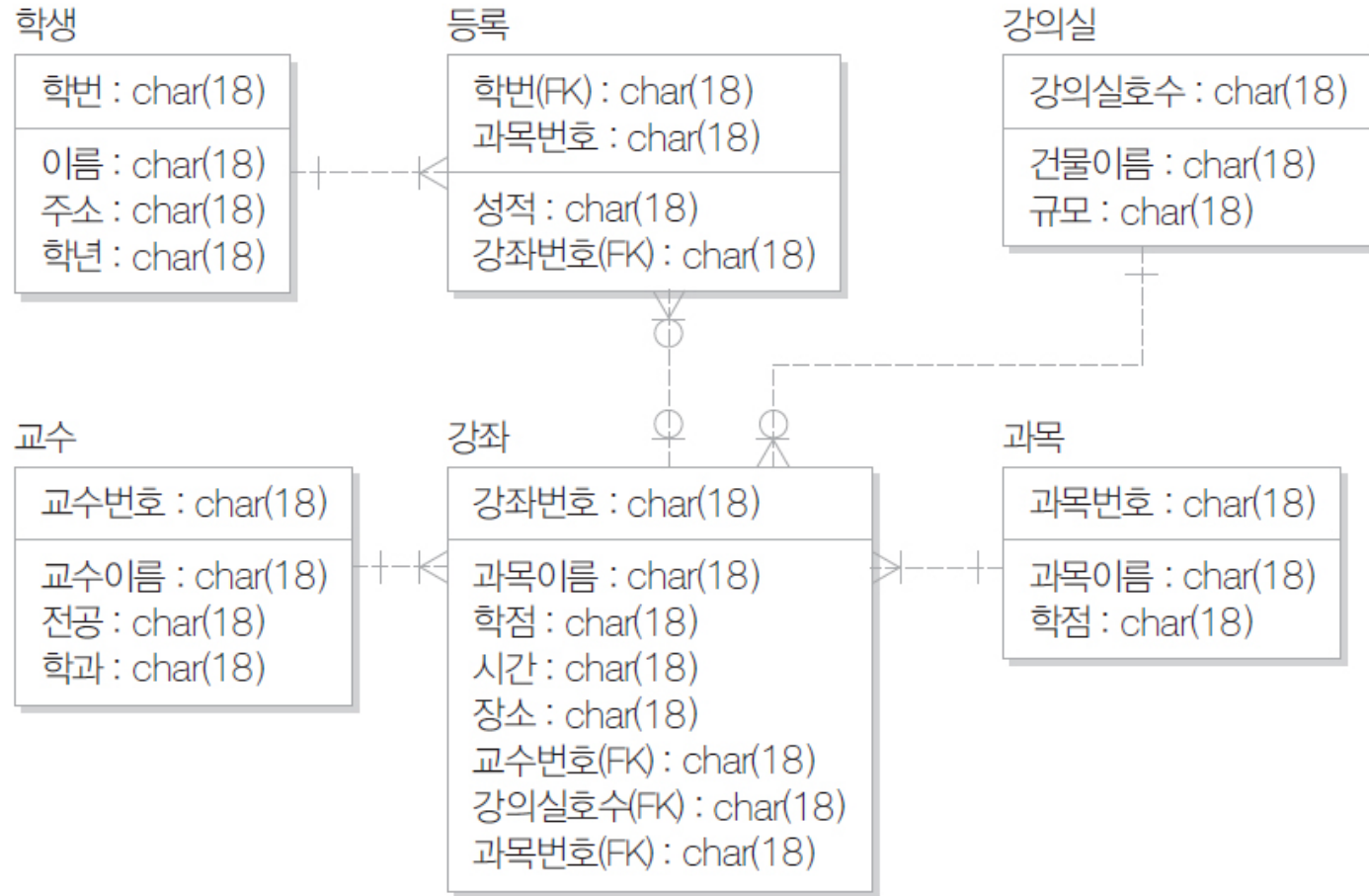


그림 1-31 수강신청 데이터베이스의 내부 스키마

# 5. 데이터베이스의 개념적 구조

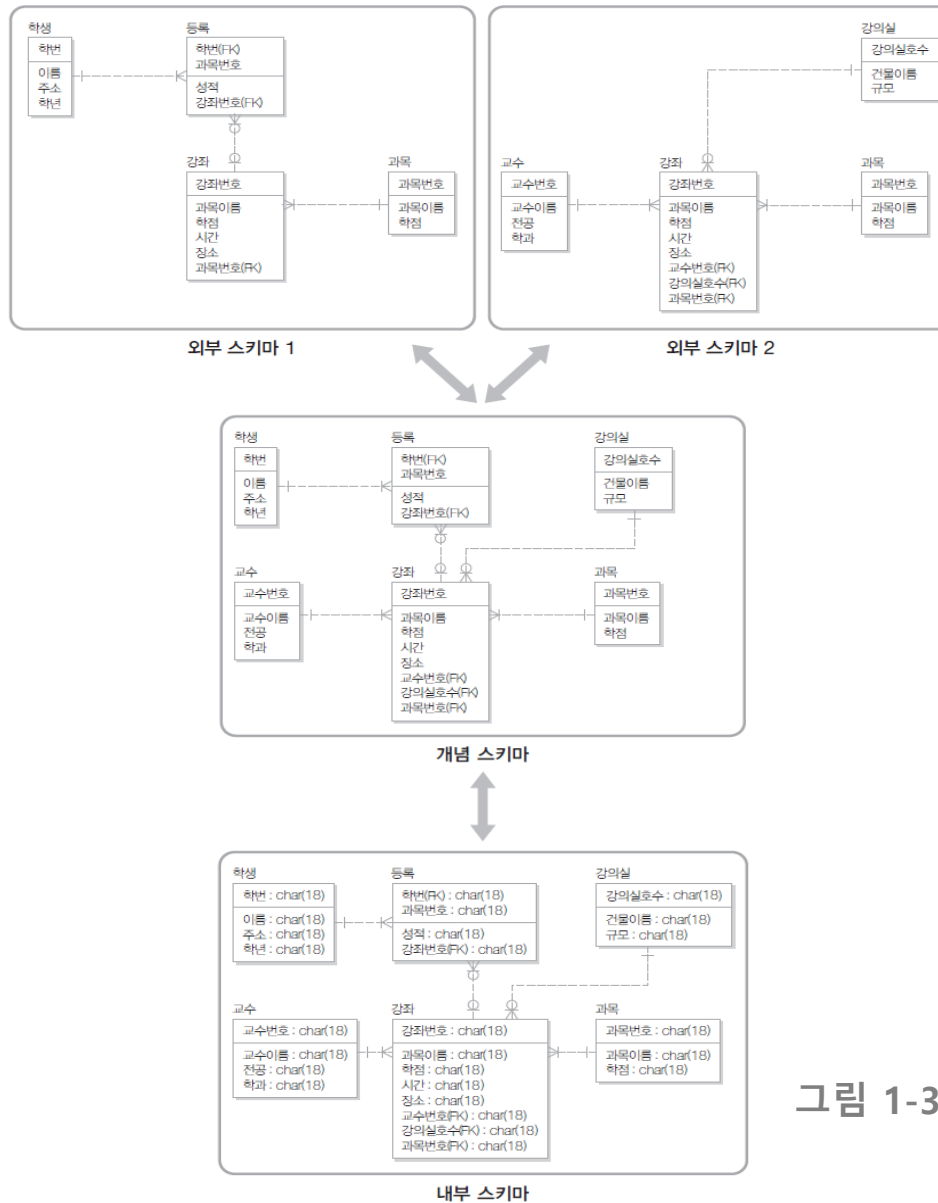


그림 1-32 수강신청 데이터베이스의 3단계 구조

## 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



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



Chapter 02

# 관계 데이터 모델

# 목차

01

관계 데이터 모델의 개념

02

무결성 제약조건

03

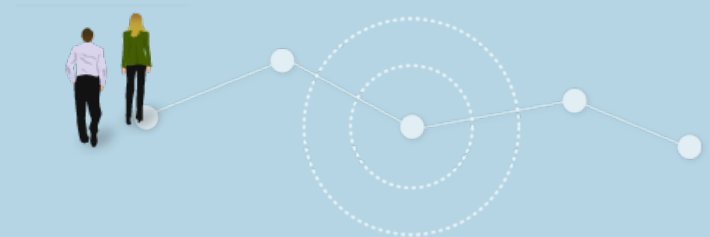
관계대수

# 학습목표

- ❖ 관계 데이터 모델의 개념을 이해한다.
- ❖ 관계 데이터 모델의 제약조건을 알아본다.
- ❖ 관계 데이터 모델의 연산인 관계대수의 종류와 작성법을 알아본다.

# 01. 관계 데이터 모델의 개념

1. 릴레이션
2. 릴레이션 스키마와 인스턴스
3. 릴레이션의 특징
4. 관계 데이터 모델



# 1. 릴레이션

- 릴레이션(relation) : 행과 열로 구성된 테이블

표 2-1 릴레이션과 관련된 한글 용어

용어	한글 용어	비고
relation	릴레이션, 테이블	"관계"라고 하지 않음
relational data model	관계 데이터 모델	
relational database	관계 데이터베이스	
relational algebra	관계대수	
relationship	관계	

# 1. 릴레이션

도서 1, 축구의 역사, 굿스포츠, 7000
도서 2, 축구아는 여자, 나무수, 13000
도서 3, 축구의 이해, 대한미디어, 22000
도서 4, 골프 바이블, 대한미디어, 35000
도서 5, 피겨 교본, 굿스포츠, 8000



도서번호	도서이름	출판사	가격
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

그림 2-1 데이터와 테이블(릴레이션)

도서번호 = {1, 2, 3, 4, 5}

도서이름 = {축구의 역사, 축구아는 여자, 축구의 이해, 골프 바이블, 피겨 교본}

출판사 = {굿스포츠, 나무수, 대한미디어}

가격 = {7000, 13000, 22000, 35000, 8000}

→ 첫 번째 행(1, 축구의 역사, 굿스포츠, 7000)의 경우 네 개의 집합에서 각각 원소 한 개씩 선택하여 만들어진 것으로 이 원소들이 관계(relationship)를 맺고 있다.

# 1. 릴레이션

## ❖ 관계(relationship)

- ❶ 릴레이션 내에서 생성되는 관계 : 릴레이션 내 데이터들의 관계
- ❷ 릴레이션 간에 생성되는 관계 : 릴레이션 간의 관계

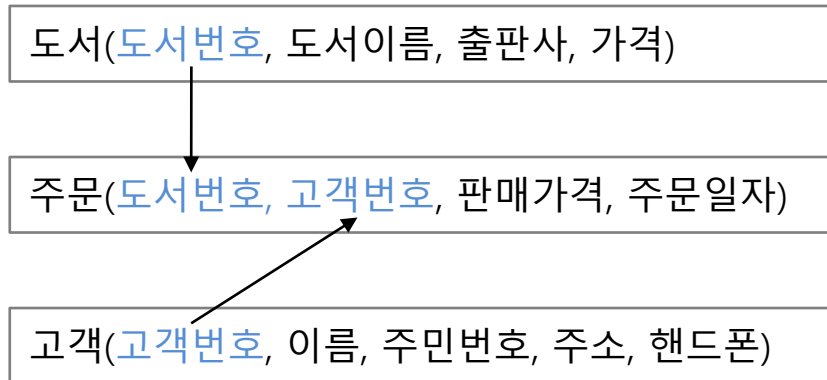


그림 2-2 릴레이션 간의 관계



## 2. 릴레이션 스키마와 인스턴스

속성(애트리뷰트),  
열(column) 이라고도 함  
(차수=4)

도서

도서번호	도서이름	출판사	가격
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

스키마(내포)  
Schema

인스턴스(외연)  
Instance

튜플(tuple),  
행(row) 이라고도 함  
(카디널리티=5)

그림 2-3 도서 릴레이션

## 2. 릴레이션 스키마와 인스턴스

### ❖ 릴레이션 스키마

#### ■ 스키마의 요소

- 속성(attribute) : 릴레이션 스키마의 열
- 도메인(domain) : 속성이 가질 수 있는 값의 집합
- 차수(degree) : 속성의 개수

#### ■ 스키마의 표현

- 릴레이션 이름(속성1 : 도메인1, 속성2 : 도메인2, 속성3 : 도메인3 ...)  
EX) 도서(도서번호, 도서이름, 출판사, 가격)

## 2. 릴레이션 스키마와 인스턴스

### ❖ 릴레이션 인스턴스

#### ■ 인스턴스 요소

- 튜플(tuple) : 릴레이션의 행
- 카디널리티(cardinality) : 튜플의 수

→ 튜플이 가지는 속성의 개수는 릴레이션 스키마의 차수와 동일하고,  
릴레이션 내의 모든 튜플들은 서로 중복되지 않아야 함

표 2-2 릴레이션 구조와 관련된 용어

릴레이션 용어	같은 의미로 통용되는 용어	파일 시스템 용어
릴레이션(relation)	테이블(table)	파일(file)
스키마(schema)	내포(intension)	헤더(header)
인스턴스(instance)	외연(extension)	데이터(data)
튜플(tuple)	행(row)	레코드(record)
속성(attribute)	열(column)	필드(field)

### 3. 릴레이션의 특징

#### ① 속성은 단일 값을 가진다

각 속성의 값은 도메인에 정의된 값만을 가지며 그 값은 모두 단일 값이어야 함

#### ② 속성은 서로 다른 이름을 가진다

속성은 한 릴레이션에서 서로 다른 이름을 가져야만 함

#### ③ 한 속성의 값은 모두 같은 도메인 값을 가진다

한 속성에 속한 열은 모두 그 속성에서 정의한 도메인 값만 가질 수 있음

#### ④ 속성의 순서는 상관없다

속성의 순서가 달라도 릴레이션 스키마는 같음

예) 릴레이션 스키마에서 (이름, 주소) 순으로 속성을 표시하거나 (주소, 이름) 순으로 표시하여도 상관없음

### 3. 릴레이션의 특징

#### ⑤ 릴레이션 내의 중복된 튜플은 허용하지 않는다

하나의 릴레이션 인스턴스 내에서는 서로 중복된 값을 가질 수 없음,  
즉 모든 튜플은 서로 값이 달라야 함

#### ⑥ 튜플의 순서는 상관없다

튜플의 순서가 달라도 같은 릴레이션임. 관계 데이터 모델의 튜플은 실제적인 값을 가지고 있으며 이 값은 시간이 지남에 따라 데이터의 삭제, 수정, 삽입에 따라 순서가 바뀔 수 있음

### 3. 릴레이션의 특징

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구 아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000
5	피겨 교본	굿스포츠	8000
6	피겨 교본, 피겨 기초	굿스포츠	8000

동일한 투플이 중복되면 안 됨

속성의 값은 단일 값이어야 함

그림 2-4 릴레이션의 특징에 위배된 경우

## 4. 관계 데이터 모델

- 관계 데이터 모델은 데이터를 2차원 테이블 형태인 릴레이션으로 표현함
- 릴레이션에 대한 제약조건과 관계 연산을 위한 관계대수를 정의함

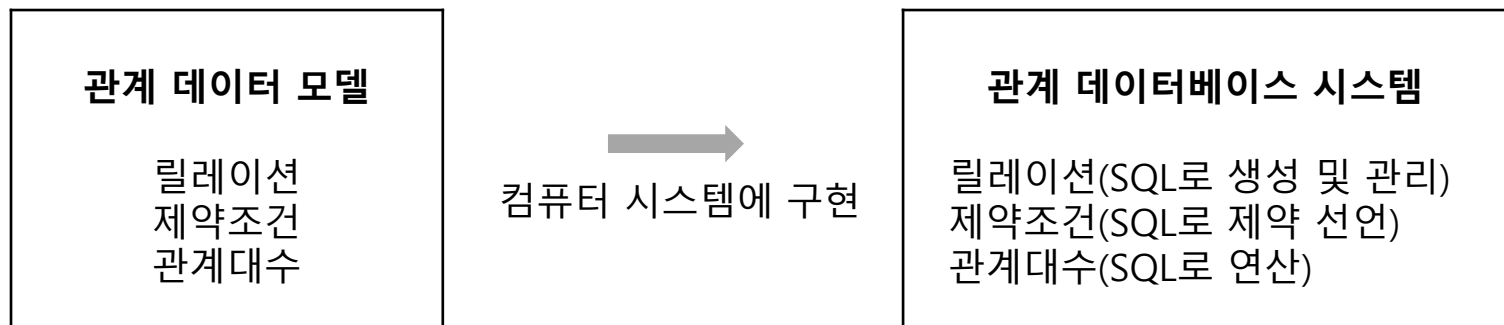


그림 2-5 관계 데이터베이스 시스템

# 연습문제

## 1. 다음 중 관계 데이터 모델의 릴레이션에 대한 설명 중 옳지 않은 것은?

- ① 릴레이션은 릴레이션 스키마와 릴레이션 인스턴스로 구성된다.
- ② 릴레이션 스키마를 릴레이션 외연(extension)이라고 한다.
- ③ 릴레이션의 스키마는 정적인 성질을 가진다.
- ④ 릴레이션 인스턴스는 동적인 성질을 가진다.

## 2. 릴레이션의 특징으로 알맞은 것은?

- ① 중복된 튜플이 존재한다.
- ② 튜플 간의 순서가 정의된다.
- ③ 속성 간의 순서가 정의된다.
- ④ 모든 속성 값은 원자값이다.

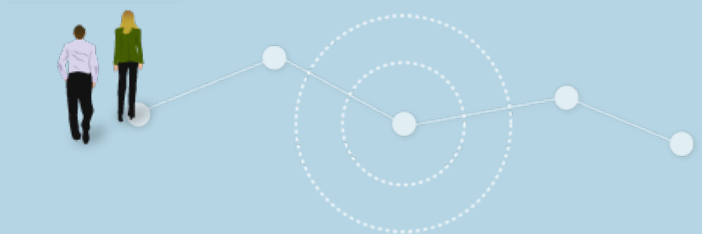
## 3. 하나의 속성이 가질 수 있는 값을 총칭하여 무엇이라 하는가?

- ① 튜플                      ② 릴레이션
- ③ 도메인                  ④ 엔티티



# 02. 무결성 제약조건

1. 키
2. 무결성 제약조건
3. 무결성 제약조건의 수행



# 1. 키

- 특정 투플을 식별할 때 사용하는 속성 혹은 속성의 집합
- 릴레이션은 중복된 투플을 허용하지 않음 → 각각의 투플에 포함된 속성들 중 어느 하나(혹은 하나 이상)는 값이 달라야 함. 즉 키가 되는 속성(혹은 속성의 집합)은 반드시 값이 달라서 투플들을 서로 구별할 수 있어야 함
- 키는 릴레이션 간의 관계를 맺는 데도 사용됨



그림 2-6 자동차 한 대당 키는 단 하나

# 1. 키

고객	고객번호	이름	주민번호	주소	핸드폰
	1	박지성	810101-1111111	영국 맨체스타	000-5000-0001
	2	김연아	900101-2222222	대한민국 서울	000-6000-0001
	3	장미란	830101-2333333	대한민국 강원도	000-7000-0001
	4	추신수	820101-1444444	미국 클리블랜드	000-8000-0001

도서	도서번호	도서이름	출판사	가격
	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

주문	고객번호	도서번호	판매가격	주문일자
	1	1	7000	2014-07-01
	1	2	13000	2014-07-03
	2	5	8000	2014-07-03
	3	2	13000	2014-07-04
	4	4	35000	2014-07-05
	1	3	22000	2014-07-07
	4	3	22000	2014-07-07

그림 2-7 마당서점 데이터베이스

# 1. 키

## ❖ 슈퍼키

### ■ 튜플을 유일하게 식별할 수 있는 하나의 속성 혹은 속성의 집합

튜플을 유일하게 식별할 수 있는 값이면 모두 슈퍼키가 될 수 있음

#### (고객 릴레이션 예)

- 고객번호 : 고객별로 유일한 값이 부여되어 있기 때문에 튜플을 식별할 수 있음
- 이름 : 동명이인이 있을 경우 튜플을 유일하게 식별할 수 없음
- 주민번호 : 개인별로 유일한 값이 부여되어 있기 때문에 튜플을 식별할 수 있음
- 주소 : 가족끼리는 같은 정보를 사용하므로 튜플을 식별할 수 없음
- 핸드폰 : 한 사람이 여러 개의 핸드폰을 사용할 수 있고 반대로 핸드폰을 사용하지 않는 사람이 있을 수 있기 때문에 튜플을 식별할 수 없음

### ■ 고객 릴레이션은 고객번호와 주민번호를 포함한 모든 속성의 집합이 슈퍼키가 됨

EX) (주민번호), (주민번호, 이름), (주민번호, 이름, 주소), (주민번호, 이름, 핸드폰),  
(고객번호), (고객번호, 이름, 주소), (고객번호, 이름, 주민번호, 주소, 핸드폰) 등

# 1. 키

## ❖ 후보키

### ■ 튜플을 유일하게 식별할 수 있는 속성의 최소 집합

(주문 릴레이션 예)

- 고객번호 : 한 명의 고객이 여러 권의 도서를 구입할 수 있으므로 후보키가 될 수 없고,  
고객번호가 1인 박지성 고객은 세 번의 주문 기록이 있으므로 튜플을 유일하게 식별할 수 없음
- 도서번호 : 도서번호가 2인 '축구아는 여자'는 두 번의 주문 기록이 있으므로 튜플을 유일하게 식별할 수 없음

### ■ 주문 릴레이션의 후보키는 2개의 속성을 합한 (고객번호, 도서번호)가 됨

### ■ 2개 이상의 속성으로 이루어진 키를 복합키(composite key)라고 함

# 1. 키

## ❖ 기본키

- 여러 후보키 중 하나를 선정하여 대표로 삼는 키
- 후보키가 하나뿐이라면 그 후보키를 기본키로 사용하면 되고, 여러 개라면 릴레이션의 특성을 반영하여 하나를 선택하면 됨

### ■ 기본키 선정 시 고려사항

- 릴레이션 내 튜플을 식별할 수 있는 고유한 값을 가져야 함.
- NULL 값은 허용하지 않음.
- 키 값의 변동이 일어나지 않아야 함.
- 최대한 적은 수의 속성을 가진 것이라야 함.
- 향후 키를 사용하는 데 있어서 문제 발생 소지가 없어야 함.

### ■ 릴레이션 스키마를 표현할 때 기본키는 밑줄을 그어 표시함

릴레이션 이름(속성1, 속성2, .... 속성N)

EX) 고객(고객번호, 이름, 주민번호, 주소, 핸드폰)

도서(도서번호, 도서이름, 출판사, 가격)

# 1. 키

## ❖ 대리키

- 기본키가 보안을 요하거나, 여러 개의 속성으로 구성되어 복잡하거나, 마땅한 기본키가 없을 때는 일련번호 같은 가상의 속성을 만들어 기본키로 삼는 경우가 있음  
이러한 키를 대리키(surrogate key) 혹은 인조키(artificial key)라고 함
- 대리키는 DBMS나 관련 소프트웨어에서 임의로 생성하는 값으로 사용자가 직관적으로 그 값의 의미를 알 수 없음

주문

주문번호	고객번호	도서번호	판매가격	주문일자
1	1	1	7000	2014-07-01
2	1	2	13000	2014-07-03
3	2	5	8000	2014-07-03
4	3	2	13000	2014-07-04
5	4	4	35000	2014-07-05
6	1	3	22000	2014-07-07
7	4	3	22000	2014-07-07

그림 2-8 대리키를 사용하도록 변경된 주문 릴레이션

# 1. 키

---

## ❖ 대체키

- 기본키로 선정되지 않은 후보키
- 고객 릴레이션의 경우 고객번호와 주민번호 중 고객번호를 기본키로 정하면 주민번호가 대체키가 됨



# 1. 키

## ❖ 외래키

FK

- 다른 릴레이션의 기본키를 참조하는 속성을 말함
- 다른 릴레이션의 기본키를 참조하여 관계 데이터 모델의 특징인 릴레이션 간의 관계(relationship)를 표현함
- 외래키의 특징
  - 관계 데이터 모델의 릴레이션 간의 관계를 표현함
  - 다른 릴레이션의 기본키를 참조하는 속성임
  - 참조하고(외래키) 참조되는(기본키) 양쪽 릴레이션의 도메인은 서로 같아야 함
  - 참조되는(기본키) 값이 변경되면 참조하는(외래키) 값도 변경됨
  - NULL 값과 중복 값 등이 허용됨
  - 자기 자신의 기본키를 참조하는 외래키도 가능함
  - 외래키가 기본키의 일부가 될 수 있음

# 1. 키

고객

고객번호	이름	주민번호	주소	핸드폰
1	박지성	810101-1111111	영국 맨체스타	000-5000-0001
2	김연아	900101-2222222	대한민국 서울	000-6000-0001
3	장미란	830101-2333333	대한민국 강원도	000-7000-0001
4	추신수	820101-1444444	미국 클리블랜드	000-8000-0001

기본키

도서

도서번호	도서이름	출판사	가격
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

기본키

참조

외래키

참조

주문

주문번호	고객번호	도서번호	판매가격	주문일자
1	1	1	7000	2014-07-01
2	1	2	13000	2014-07-03
3	2	5	8000	2014-07-03
4	3	2	13000	2014-07-04
5	4	4	35000	2014-07-05
6	1	3	22000	2014-07-07
7	4	3	22000	2014-07-07


기본키

그림 2-9 릴레이션 간의 참조 관계

# 1. 키

## ❖ 외래키

- 외래키 사용 시 참조하는 릴레이션과 참조되는 릴레이션이 꼭 다른 릴레이션일 필요는 없음. 즉 자기 자신의 기본키를 참조할 수도 있음



선수번호	이름	주소	멘토번호
1	박지성	영국 맨체스타	NULL
2	김연아	대한민국 서울	3
3	장미란	대한민국 강원도	4
4	추신수	미국 클리블랜드	NULL

그림 2-10 멘토 릴레이션

# 1. 키

## ❖ 요약

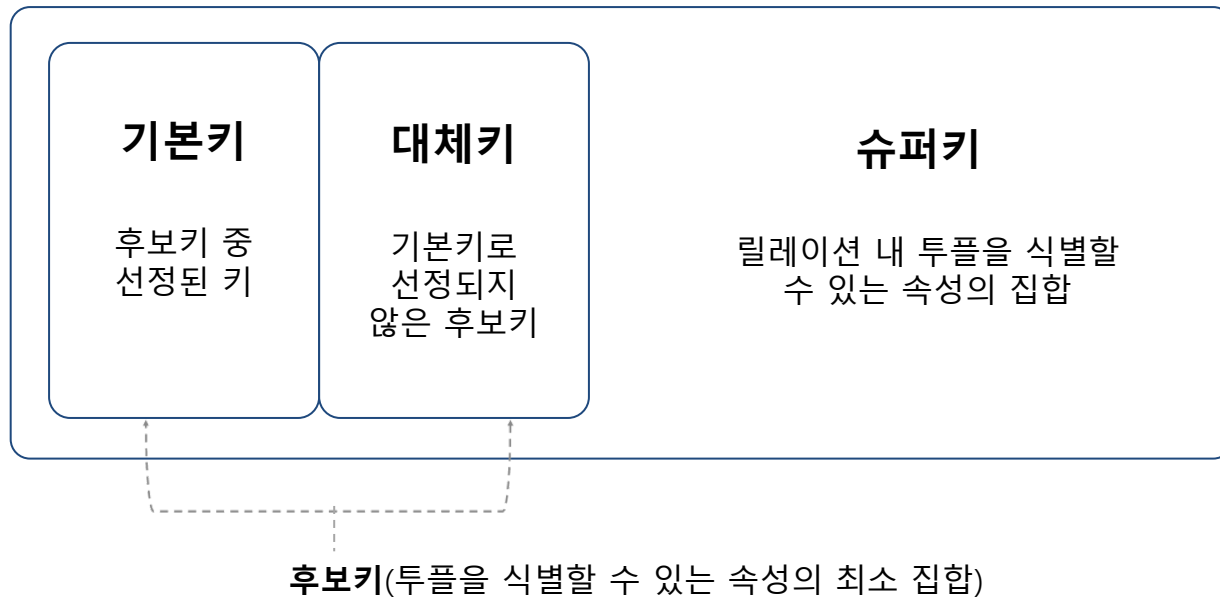


그림 2-11 키의 포함 관계

## 2. 무결성 제약조건

### ■ 데이터 무결성(integrity, 無缺性)

데이터베이스에 저장된 데이터의 일관성과 정확성을 지키는 것

### ■ 도메인 무결성 제약조건

도메인 제약(domain constraint)이라고도 하며, 릴레이션 내의 튜플들이 각 속성의 도메인에 지정된 값만을 가져야 한다는 조건. SQL 문에서 데이터 형식(type), 널(null/not null), 기본 값(default), 체크(check) 등을 사용하여 지정할 수 있음.

### ■ 개체 무결성 제약조건

기본키 제약(primary key constraint)이라고도 함. 릴레이션은 기본키를 지정하고 그에 따른 무결성 원칙, 즉 기본키는 NULL 값을 가져서는 안 되며 릴레이션 내에 오직 하나의 값만 존재해야 한다는 조건임.

### ■ 참조 무결성 제약조건

외래키 제약(foreign key constraint)이라고도 하며, 릴레이션 간의 참조 관계를 선언하는 제약조건임. 자식 릴레이션의 외래키는 부모 릴레이션의 기본키와 도메인이 동일해야 하며, 자식 릴레이션의 값이 변경될 때 부모 릴레이션의 제약을 받는다는 것임

## 2. 무결성 제약조건

표 2-3 제약조건의 정리

구분	도메인	키	
	도메인 무결성 제약조건	개체 무결성 제약조건	참조 무결성 제약조건
제약 대상	속성	투플	속성과 투플
같은 용어	도메인 제약 (Domain Constraint)	기본키 제약 (Primary Key Constraint)	외래키 제약 (Foreign Key Constraint)
해당되는 키	-	기본키	외래키
NULL 값 허용 여부	허용	불가	허용
릴레이션 내 제약조건의 개수	속성의 개수와 동일	1개	0~여러 개
기타	<ul style="list-style-type: none"> <li>투플 삽입, 수정 시 제약 사항 우선 확인</li> </ul>	<ul style="list-style-type: none"> <li>투플 삽입/수정 시 제약 사항 우선 확인</li> </ul>	<ul style="list-style-type: none"> <li>투플 삽입/수정 시 제약사항 우선 확인</li> <li>부모 릴레이션의 투플 수정/삭제 시 제약사항 우선 확인</li> </ul>

## 2. 무결성 제약조건

### ❖ 개체 무결성 제약조건

- 삽입 : 기본키 값이 같으면 삽입이 금지됨
- 수정 : 기본키 값이 같거나 NULL로도 수정이 금지됨
- 삭제 : 특별한 확인이 필요하지 않으며 즉시 수행함

학번	이름	학과코드
501	박지성	1001
401	김연아	2001
402	장미란	2001
502	추신수	1001

그림 2-12 학생 릴레이션

(501, 남슬찬, 1001)

↓ 삽입 거부

학번	이름	학과코드
501	박지성	1001
401	김연아	2001
402	장미란	2001
502	추신수	1001

(NULL, 남슬찬, 1001)

↓ 삽입 거부

학번	이름	학과코드
501	박지성	1001
401	김연아	2001
402	장미란	2001
502	추신수	1001

그림 2-13 개체 무결성 제약조건 수행 예  
(기본키 충돌 및 NULL 값 삽입)

## 2. 무결성 제약조건

### ❖ 참조 무결성 제약조건

#### ■ 삽입

- 학과(부모 릴레이션) : 튜플 삽입한 후 수행하면 정상적으로 진행된다.
- 학생(자식 릴레이션) : 참조받는 테이블에 외래키 값이 없으므로 삽입이 금지된다.

학생(자식 릴레이션)

학번	이름	학과코드
501	박지성	1001
401	김연아	2001
402	장미란	2001
502	추신수	1001

학과(부모 릴레이션)

학과코드	학과명
1001	컴퓨터학과
2001	체육학과

참조

그림 2-14 학생관리 데이터베이스



## 2. 무결성 제약조건

### ❖ 참조 무결성 제약조건

#### ■ 삭제

- 학과(부모 릴레이션) : 참조하는 테이블을 같이 삭제할 수 있어서 금지하거나 다른 추가 작업이 필요함
- 학생(자식 릴레이션) : 바로 삭제 가능함

※ 부모 릴레이션에서 튜플을 삭제할 경우 참조 무결성 조건을 수행하기 위한 고려사항

- ❶ 즉시 작업을 중지
- ❷ 자식 릴레이션의 관련 튜플을 삭제
- ❸ 초기에 설정된 다른 어떤 값으로 변경
- ❹ NULL 값으로 설정

## 2. 무결성 제약조건

---

### ❖ 참조 무결성 제약조건

#### ■ 수정

- 삭제와 삽입 명령이 연속해서 수행됨.
- 부모 릴레이션의 수정이 일어날 경우 삭제 옵션에 따라 처리된 후 문제가 없으면 다시 삽입 제약조건에 따라 처리됨.

## 2. 무결성 제약조건

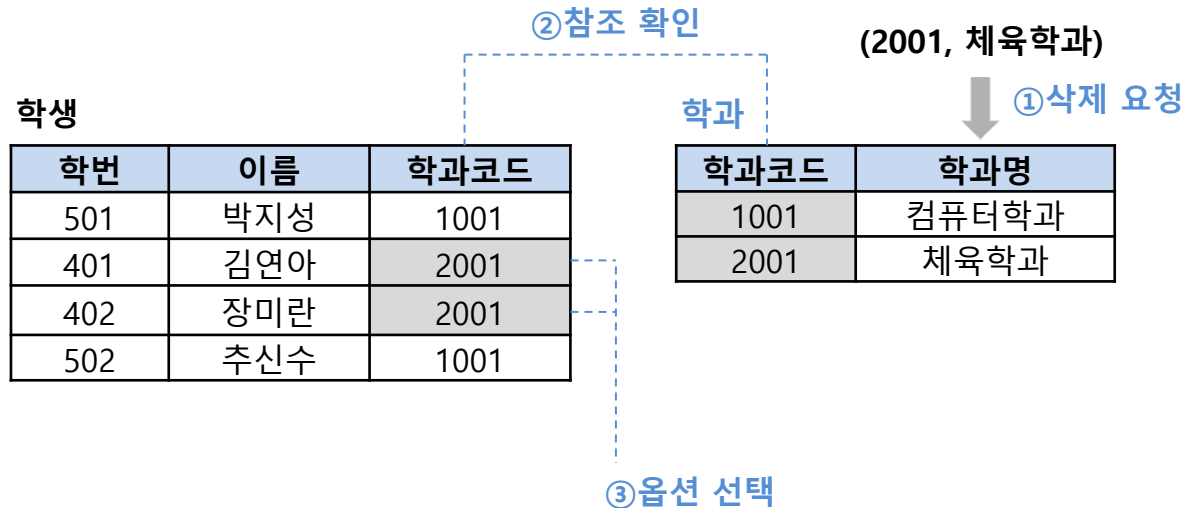
### ❖ 참조 무결성 제약조건

표 2-4 참조 무결성 제약조건의 옵션(부모 릴레이션에서 튜플을 삭제할 경우)

명령어	의미	예
RESTRICTED	자식 릴레이션에서 참조하고 있을 경우 부모 릴레이션의 삭제 작업을 거부함	학과 릴레이션의 튜플 삭제 거부
CASCADE	자식 릴레이션의 관련 튜플을 같이 삭제 처리함	학생 릴레이션의 관련 튜플을 삭제
DEFAULT	자식 릴레이션의 관련 튜플을 미리 설정해둔 값으로 변경함	학생 릴레이션의 학과가 다른 학과로 자동 배정
NULL	자식 릴레이션의 관련 튜플을 NULL 값으로 설정함(NULL 값을 허가한 경우)	학과 릴레이션의 학과가 NULL 값으로 변경

## 2. 무결성 제약조건

### ❖ 참조 무결성 제약조건

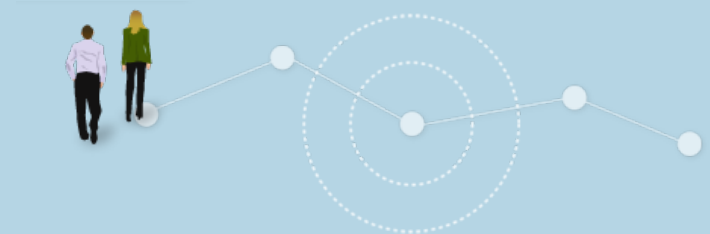


- ① RESTRICTED : 요청한 삭제 작업중지(에러 처리)
- ② CASCADE : 학생 릴레이션의 해당 튜플을 같이 연쇄적으로 삭제(CASCADE)
- ③ 기본값으로 변경(미리 설정한 값, DEFAULT)
- ④ NULL 값으로 설정

그림 2-15 참조 무결성 제약조건에서 부모 릴레이션의 튜플을 삭제할 경우

### 03. 관계대수

1. 관계대수
2. 선택과 투영
3. 집합 연산
4. 조인
5. 디비전
6. 관계대수 예제



# 1. 관계대수

## ■ 관계대수(relational algebra, 關係代數)

릴레이션에서 원하는 결과를 얻기 위해 수학의 대수와 같은 연산을 이용하여 질의하는 방법을 기술하는 언어

## ■ 관계대수와 관계해석

- **관계대수** : 어떤 데이터를 어떻게 찾는지에 대한 처리 절차를 명시하는 절차적인 언어이며, DBMS 내부의 처리 언어로 사용됨
- **관계해석** : 어떤 데이터를 찾는지만 명시하는 선언적인 언어로 관계대수와 함께 관계 DBMS의 표준 언어인 SQL의 이론적인 기반을 제공함

→ 관계대수와 관계해석은 모두 관계 데이터 모델의 중요한 언어이며 실제 동일한 표현 능력을 가지고 있음.

# 1. 관계대수

## ❖ 관계의 수학적 의미

### ■ 릴레이션(relation)의 수학적 개념

예)  $A = \{2, 4\}$ ,  $B = \{1, 3, 5\}$  일 때

$$A \times B = \{(2, 1), (2, 3), (2, 5), (4, 1), (4, 3), (4, 5)\}$$

릴레이션 R은 카티전 프로덕트의 부분집합으로 정의

$$\text{예) } R1 = \{(2, 1), (4, 1)\}, R2 = \{(2, 1), (2, 3), (2, 5)\}, R3 = \{(2, 3), (2, 5), (4, 3), (4, 5)\}$$

원소 개수가  $n$ 인 집합  $S$ 의 부분집합의 개수는  $2^n$ 이므로, 카티전 프로덕트  $A \times B$ 의 부분집합의 개수는  $2^{|A| \times |B|}$ 임

카티전 프로덕트의 기초 집합  $A, B$  각각이 가질 수 있는 값의 범위를 도메인(domain)이라고 함.  
즉 집합  $A$ 의 도메인은  $\{2, 4\}$ 임

릴레이션 역시 집합이므로 집합에서 집합에서 가능한 연산은 합집합( $\cup$ ), 교집합( $\cap$ ), 카티전 프로덕트( $\times$ ) 등이 있음

$$R1 \cup R2 = \{(2, 1), (4, 1), (2, 3), (2, 5)\}$$

$$R1 \cap R2 = \{(2, 1)\}$$

# 1. 관계대수

## ❖ 관계의 수학적 의미

### ■ 릴레이션의 현실 세계 적용

예) 학번={2, 4}, 과목={데이터베이스, 자료구조, 프로그래밍}일 때

두 집합의 카티전 프로덕트 학번×과목은 학번 원소와 과목 원소의 순서쌍의 집합임.

즉, 학번×과목={(2, 데이터베이스), (2, 자료구조), (2, 프로그래밍), (4, 데이터베이스), (4, 자료구조), (4, 프로그래밍)}을 말함.

학번×과목의 각 원소는 학생이 과목을 수강할 수 있는 모든 경우를 나열한 것임.

수강={(2, 데이터베이스), (2, 자료구조), (4, 프로그래밍)}은 카티전 프로덕트 학번×과목의 부분집합으로 하나의 릴레이션 인스턴스임. 수강 릴레이션의 튜플은 위에서 나열한 여섯 개 원소 중 하나로, 아래 수강 테이블을 데이터베이스에서는 릴레이션(relation)이라고 함.

수강

학번	과목
2	데이터베이스
2	자료구조
4	프로그래밍

그림 2-16 수강 릴레이션



# 1. 관계대수

## ❖ 관계대수 연산자

표 2-5 관계대수 연산자

연산자 종류	대상	연산자 이름	기호	설명
기본	단항	선택	$\sigma$	릴레이션에서 조건에 만족하는 튜플을 선택
기본	단항	프로젝션	$\pi$	릴레이션의 속성을 선택
추가	단항	개명	$\rho$	릴레이션이나 속성의 이름을 변경
유도	이항	디비전	$\div$	부모 릴레이션에 포함된 튜플의 값을 모두 갖고 있는 튜플을 분자 릴레이션에서 추출
기본	이항	합집합	$\cup$	두 릴레이션의 합집합
기본	이항	차집합	$-$	두 릴레이션의 차집합
유도	이항	교집합	$\cap$	두 릴레이션의 교집합
기본	이항	카디전 프로덕트	$\times$	두 릴레이션에 속한 모든 튜플의 집합

# 1. 관계대수

## ❖ 관계대수 연산자

표 2-5 관계대수 연산자

연산자 종류	대상	연산자 이름		기호	설명
유도	이항	조인	세타		두 릴레이션 간의 비교 조건에 만족하는 집합
			동등		두 릴레이션 간의 같은 값을 가진 집합
			자연		동등 조인에서 중복 속성을 제거
			세미	left	자연 조인 후 오른쪽 속성을 제거
				right	자연 조인 후 왼쪽 속성을 제거
		외부	left		<ul style="list-style-type: none"> <li>자연 조인 후 각각 왼쪽(left), 오른쪽(right), 양쪽(full)의 모든 값을 결과로 추출</li> <li>조인이 실패(또는 값이 없을 경우)한 쪽의 값을 NULL로 채움</li> </ul>
			right		
			full		

# 1. 관계대수

## ❖ 관계대수식

관계대수는 릴레이션 간 연산을 통해 결과 릴레이션을 찾는 절차를 기술한 언어로, 이 연산을 수행하기 위한 식을 관계대수식(relational algebra expression)이라고 함. 관계대수식은 대상이 되는 릴레이션과 연산자로 구성되며, 결과는 릴레이션으로 반환됨. 반환된 릴레이션은 릴레이션의 모든 특징을 따름

- 단항 연산자 : 연산자<sub><조건></sub> 릴레이션
- 이항 연산자 : 릴레이션1 연산자<sub><조건></sub> 릴레이션2

R1			R2		
A	B	C	A	B	C
a1	b1	c1	a1	b1	c1
a2	b3	c3	a2	b3	c3
a3	b4	c2	a3	b3	c1

그림 2-17 관계대수식을 이해하기 위한 예제 데이터

R1

A	B	C
a1	b1	c1
a2	b3	c3
a3	b4	c2

R2

A	B	C
a1	b1	c1
a2	b3	c3
a3	b3	c1

표 2-6 관계대수 식의 사용 예

주요 연산자	사용 예	결과	설명																								
선택선( $\sigma$ )	$\sigma_{A=a1 \text{ or } A=a2} (R1)$	<table><tr><th>A</th><th>B</th><th>C</th></tr><tr><td>a1</td><td>b1</td><td>c1</td></tr><tr><td>a2</td><td>b3</td><td>c3</td></tr></table>	A	B	C	a1	b1	c1	a2	b3	c3	R1에서 조건에 맞는 튜플을 추출함															
A	B	C																									
a1	b1	c1																									
a2	b3	c3																									
프로젝션( $\pi$ )	$\pi_{A,B}(R2)$	<table><tr><th>A</th><th>B</th></tr><tr><td>a1</td><td>b1</td></tr><tr><td>a2</td><td>b3</td></tr><tr><td>a3</td><td>b3</td></tr></table>	A	B	a1	b1	a2	b3	a3	b3	R2에서 조건에 맞는 속성만을 추출함																
A	B																										
a1	b1																										
a2	b3																										
a3	b3																										
합집합( $\cup$ )	$R1 \cup R2$	<table><tr><th>A</th><th>B</th><th>C</th></tr><tr><td>a1</td><td>b1</td><td>c1</td></tr><tr><td>a2</td><td>b3</td><td>c3</td></tr><tr><td>a3</td><td>b4</td><td>c2</td></tr><tr><td>a3</td><td>b3</td><td>c1</td></tr></table>	A	B	C	a1	b1	c1	a2	b3	c3	a3	b4	c2	a3	b3	c1	R1과 R2의 합집합을 구함									
A	B	C																									
a1	b1	c1																									
a2	b3	c3																									
a3	b4	c2																									
a3	b3	c1																									
차집합( $-$ )	$R1 - R2$	<table><tr><th>A</th><th>B</th><th>C</th></tr><tr><td>a3</td><td>b4</td><td>c2</td></tr></table>	A	B	C	a3	b4	c2	R1과 R2의 차집합을 구함																		
A	B	C																									
a3	b4	c2																									
조인	$R1 \bowtie_{R1.C=R2.C} R2$	<table><tr><th>R1.A</th><th>R1.B</th><th>R1.C</th><th>R2.A</th><th>R2.B</th><th>R2.C</th></tr><tr><td>a1</td><td>b1</td><td>c1</td><td>a1</td><td>b1</td><td>c1</td></tr><tr><td>a1</td><td>b1</td><td>c1</td><td>a3</td><td>b3</td><td>c1</td></tr><tr><td>a2</td><td>b3</td><td>c3</td><td>a2</td><td>b3</td><td>c3</td></tr></table>	R1.A	R1.B	R1.C	R2.A	R2.B	R2.C	a1	b1	c1	a1	b1	c1	a1	b1	c1	a3	b3	c1	a2	b3	c3	a2	b3	c3	R1과 R2의 카티전 프로덕트를 구하여 조건에 맞는 튜플을 추출함
R1.A	R1.B	R1.C	R2.A	R2.B	R2.C																						
a1	b1	c1	a1	b1	c1																						
a1	b1	c1	a3	b3	c1																						
a2	b3	c3	a2	b3	c3																						

## 2. 선택과 프로젝션

### ❖ 선택(selection)

- 릴레이션의 튜플을 추출하기 위한 연산임. 하나의 릴레이션을 대상으로 하는 단항 연산자며, 찾고자 하는 튜플의 조건(predicate)을 명시하고 그 조건에 만족하는 튜플을 반환함
- 형식 :  $\sigma_{\langle \text{조건} \rangle} (R)$  (R은 릴레이션,  $\sigma$ 는 그리스 문자이며 대문자는  $\Sigma$ )

질의 2-1 마당서점에서 판매하는 도서 중 8,000원 이하인 도서를 검색하시오.

$\sigma_{\text{가격} \leq 8000} (\text{도서})$

도서

도서번호	도서이름	출판사	가격
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

$\sigma_{\text{가격} \leq 8000}$   
(도서)



도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
5	피겨 교본	굿스포츠	8000

그림 2-18 선택의 예

## 2. 셀렉션과 프로젝션

### ❖ 셀렉션의 확장

- 형식 :  $\sigma_{\langle \text{복합조건} \rangle} (R)$  (R은 릴레이션,  $\sigma$ 는 그리스 문자이며 대문자는  $\Sigma$ )
- 여러 개의 조건을  $\wedge$  (and),  $\vee$  (or),  $\neg$  (not) 기호를 이용하여 복합조건을 표시할 수 있다. 예를 들어, “가격이 8,000원 이하이고, 도서번호가 3 이상인 책을 찾아라”는 질의는 다음과 같이 표현한다.

$\sigma_{(\text{가격} \leq 8000 \wedge \text{도서번호} \geq 3)} (\text{도서})$

## 2. 선택과 프로젝션

### ❖ 프로젝션(projection)

- 릴레이션의 속성을 추출하기 위한 연산으로 단항 연산자임.
- 형식 :  $\pi_{\langle \text{속성리스트} \rangle} (R)$  (R은 릴레이션,  $\pi$  는 그리스 문자이며 대문자는  $\Pi$  )

질의 2-2 신간도서 안내를 위해  
고객의 (이름, 주소, 핸드폰)이 적힌  
카탈로그 주소록을 만드시오.

$\pi_{\text{이름, 주소, 핸드폰}} (\text{고객})$

고객

고객번호	이름	주민번호	주소	핸드폰
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001
2	김연아	900905-2222222	대한민국 서울	000-6000-0001
3	장미란	831009-2333333	대한민국 강원도	000-7000-0001
4	추신수	820713-1444444	미국 클리블랜드	000-8000-0001

$\pi_{\text{이름, 주소, 핸드폰}} (\text{고객})$

이름	주소	핸드폰
박지성	영국 맨체스타	000-5000-0001
김연아	대한민국 서울	000-6000-0001
장미란	대한민국 강원도	000-7000-0001
추신수	미국 클리블랜드	000-8000-0001

그림 2-19 프로젝션의 예

### 3. 집합 연산

#### ❖ 합집합

- 두 개의 릴레이션을 합하여 하나의 릴레이션을 반환함. 이 때 두 개의 릴레이션은 서로 같은 속성 순서와 도메인을 가져야 함
- 형식 :  $R \cup S$

질의 2-3 마당서점은 지점A와 지점B가 있다. 두 지점의 도서는 각 지점에서 관리하며 릴레이션 이름은 각각 도서A, 도서B다. 마당서점의 도서를 하나의 릴레이션으로 보이시오.

도서A  $\cup$  도서B

도서A

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000

$\cup$

도서B

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000



도서번호	도서이름	출판사	가격
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

그림 2-20 합집합의 예



### 3. 집합 연산

#### ❖ 교집합

- 합병가능한 두 릴레이션을 대상으로 하며, 두 릴레이션이 공통으로 가지고 있는 튜플을 반환함
- 형식 :  $R \cap S$

질의 2-4 마당서점의 두 지점에서 동일하게 보유하고 있는 도서 목록을 보이시오.

도서A  $\cap$  도서B

도서A

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000

$\cap$

도서B

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000



도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000

그림 2-21 교집합의 예

### 3. 집합 연산

#### ❖ 차집합

- 첫 번째 릴레이션에는 속하고 두 번째 릴레이션에는 속하지 않는 튜플을 반환함.
- 형식 :  $R - S$

질의 2-5 마당서점 두 지점 중 지점 A에서만 보유하고 있는 도서 목록을 보이시오.

도서A - 도서B

도서A

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000

도서B

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000

-



도서번호	도서이름	출판사	가격
2	축구아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000

그림 2-22 차집합의 예

### 3. 집합 연산

#### ❖ 카티전 프로덕트(cartesian product)

- 두 릴레이션을 연결시켜 하나로 합칠 때 사용함. 결과 릴레이션은 첫 번째 릴레이션의 오른쪽에 두 번째 릴레이션의 모든 튜플을 순서대로 배열하여 반환함. 결과 릴레이션의 차수는 두 릴레이션의 차수의 합이며, 카디널리티는 두 릴레이션의 카디널리티의 곱임
- 형식 :  $R \times S$

질의 2-6 고객 릴레이션과 주문 릴레이션의 카티전 프로덕트를 구하시오  
(결과가 많으므로 튜플을 일부 삭제한 릴레이션을 사용함).

고객  $\times$  주문

### 3. 집합 연산

고객

고객번호	이름	주소	핸드폰
1	박지성	영국 맨체스타	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	장미란	대한민국 강원도	000-7000-0001

×

주문

주문번호	고객번호	도서번호	판매가격	주문일자
1	2	1	7000	2014-07-01
2	1	2	13000	2014-07-03
3	2	5	8000	2014-07-03
4	1	2	13000	2014-07-04



고객번호	이름	주소	핸드폰	주문번호	고객번호	도서번호	판매가격	주문일자
1	박지성	영국 맨체스타	000-5000-0001	1	2	1	7000	2014-07-01
1	박지성	영국 맨체스타	000-5000-0001	2	1	2	13000	2014-07-03
1	박지성	영국 맨체스타	000-5000-0001	3	2	5	8000	2014-07-03
1	박지성	영국 맨체스타	000-5000-0001	4	1	2	13000	2014-07-04
2	김연아	대한민국 서울	000-6000-0001	1	2	1	7000	2014-07-01
2	김연아	대한민국 서울	000-6000-0001	2	1	2	13000	2014-07-03
2	김연아	대한민국 서울	000-6000-0001	3	2	5	8000	2014-07-03
2	김연아	대한민국 서울	000-6000-0001	4	1	2	13000	2014-07-04
3	장미란	대한민국 강원도	000-7000-0001	1	2	1	7000	2014-07-01
3	장미란	대한민국 강원도	000-7000-0001	2	1	2	13000	2014-07-03
3	장미란	대한민국 강원도	000-7000-0001	3	2	5	8000	2014-07-03
3	장미란	대한민국 강원도	000-7000-0001	4	1	2	13000	2014-07-04

그림 2-23 카티전 프로덕트의 예

## 4. 조인

- 두 릴레이션의 공통 속성을 기준으로 속성 값이 같은 튜플을 수평으로 결합하는 연산임. 조인을 수행하기 위해서는 두 릴레이션의 조인에 참여하는 속성이 서로 동일한 도메인으로 구성되어야 함. 조인 연산의 결과는 공통 속성의 속성 값이 동일한 튜플만을 반환함.
- 형식 :  $R \bowtie S = \sigma_c(R \times S)$  (R과 S는 릴레이션, c 는 조인조건)
- 조인 연산의 구분
  - 기본연산 : 세타조인( $\bowtie_\theta$ ), 동등조인( $\bowtie$ ), 자연조인( $\bowtie_N$ )
  - 확장된 조인 연산 : 세미조인( $\ltimes, \rtimes$ ), 외부조인( $\ltimes, \bowtie, \rtimes$ )

## 4. 조인

### ❖ 세타조인과 동등조인

#### ■ 세타조인(theta join, $\theta$ )

- 조인에 참여하는 두 릴레이션의 속성 값을 비교하여 조건을 만족하는 튜플만 반환함
- 세타조인의 조건은  $\{=, \neq, \leq, \geq, <, >\}$  중 하나가 됨
- 형식 :  $R \bowtie_{(r \text{ 조건 } s)} S$  ( $R$ 과  $S$ 는 릴레이션이며  $r$ 은  $R$ 의 속성,  $s$ 는  $S$ 의 속성)

#### ■ 동등조인(equi join)

- 세타조인에서  $=$  연산자를 사용한 조인을 말함. 보통 조인 연산이라고 하면 동등조인을 지칭함
- 형식 :  $R \bowtie_{(r = s)} S$

질의 2-7 고객과 고객의 주문 사항을 모두 보이시오.

고객  $\bowtie_{\text{고객.고객번호=주문.고객번호}}$  주문


## 4. 조인

고객

고객번호	이름	주소	핸드폰
1	박지성	영국 맨체스타	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	장미란	대한민국 강원도	000-7000-0001
4	추신수	미국 클리블랜드	000-8000-0001

주문

주문번호	고객번호	도서번호	판매가격	주문일자
1	2	1	7000	2014-07-01
2	1	2	13000	2014-07-03
3	2	5	8000	2014-07-03
4	1	2	13000	2014-07-04
5	4	4	35000	2014-07-05
6	5	3	22000	2014-07-07
7	4	3	22000	2014-07-07

고객  고객.고객번호=주문.고객번호 주문

고객번호	이름	주소	핸드폰	주문번호	고객번호	도서번호	판매가격	주문일자
1	박지성	영국 맨체스타	000-5000-0001	2	1	2	13000	2014-07-03
1	박지성	영국 맨체스타	000-5000-0001	4	1	2	13000	2014-07-04
2	김연아	대한민국 서울	000-6000-0001	1	2	1	7000	2014-07-01
2	김연아	대한민국 서울	000-6000-0001	3	2	5	8000	2014-07-03
4	추신수	미국 클리블랜드	000-8000-0001	5	4	4	35000	2014-07-05
4	추신수	미국 클리블랜드	000-8000-0001	7	4	3	22000	2014-07-07

그림 2-24 동등조인의 예

## 4. 조인

### ❖ 자연조인(natural join)

- 동등조인에서 조인에 참여한 속성이 두 번 나오지 않도록 두 번째 속성을 제거한 결과를 반환함.
- 형식 :  $R \bowtie_{N(r, s)} S$

질의 2-8    고객과 고객의 주문 사항을 모두 보여주되 같은 속성은 한 번만 표시하시오.

고객  $\bowtie_{N(\text{고객.고객번호}=\text{주문.고객번호})}$  주문




## 4. 조인

고객

고객번호	이름	주소	핸드폰
1	박지성	영국 맨체스타	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	장미란	대한민국 강원도	000-7000-0001
4	추신수	미국 클리블랜드	000-8000-0001

주문

주문번호	고객번호	도서번호	판매가격	주문일자
1	2	1	7000	2014-07-01
2	1	2	13000	2014-07-03
3	2	5	8000	2014-07-03
4	1	2	13000	2014-07-04
5	4	4	35000	2014-07-05
6	5	3	22000	2014-07-07
7	4	3	22000	2014-07-07

고객  N(고객.고객번호=주문.고객번호) 주문

고객번호	이름	주소	핸드폰	주문번호	도서번호	판매가격	주문일자
1	박지성	영국 맨체스타	000-5000-0001	2	2	13000	2014-07-03
1	박지성	영국 맨체스타	000-5000-0001	4	2	13000	2014-07-04
2	김연아	대한민국 서울	000-6000-0001	1	1	7000	2014-07-01
2	김연아	대한민국 서울	000-6000-0001	3	5	8000	2014-07-03
4	추신수	미국 클리블랜드	000-8000-0001	5	4	35000	2014-07-05
4	추신수	미국 클리블랜드	000-8000-0001	7	3	22000	2014-07-07

그림 2-25 자연조인의 예

## 4. 조인

### ❖ 외부조인과 세미조인

#### ■ 외부조인(outer join)

- 자연조인 시 조인에 실패한 튜플을 모두 보여주되 값이 없는 대응 속성에는 NULL 값을 채워서 반환
- 모든 속성을 보여주는 기준 릴레이션 위치에 따라 왼쪽(left) 외부조인, 오른쪽(right) 외부조인, 완전(full, 양쪽) 외부조인으로 나뉨

- 형식 : 왼쪽(left) 외부조인 -  $R \bowtie_{(r, s)} S$

완전(full) 외부조인 -  $R \bowtie_{(r, s)} S$

오른쪽(right) 외부조인 -  $R \bowtie_{(r, s)} S$

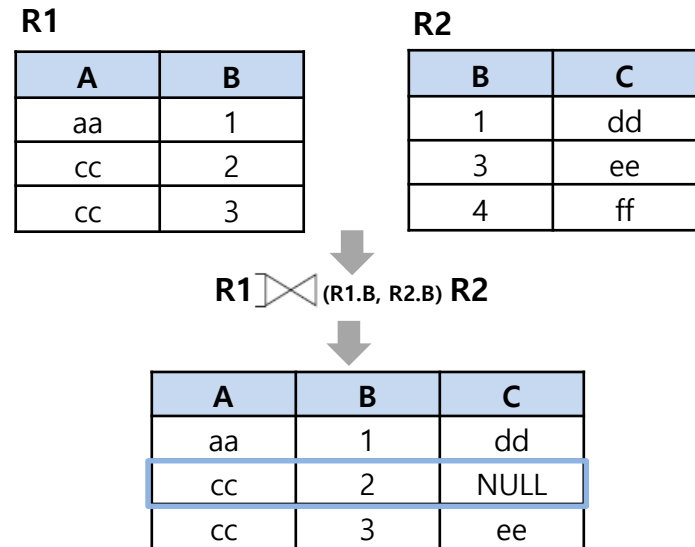





그림 2-26 왼쪽 외부조인의 예

## 4. 조인

### ❖ 외부조인과 세미조인

질의 2-9 마당서점의 고객과 고객의 주문 내역을 보이시오.

- ① 고객 기준으로 주문내역이 없는 고객도 모두 보이시오.
- ② 주문내역이 없는 고객과, 고객 릴레이션에 고객번호가 없는 주문을 모두 보이시오.
- ③ 주문내역 기준으로 고객 릴레이션에 고객번호가 없는 주문도 모두 보이시오.

- ① 고객  (고객.고객번호, 주문.고객번호) 주문
- ② 고객  (고객.고객번호, 주문.고객번호) 주문
- ③ 고객  (고객.고객번호, 주문.고객번호) 주문

## 4. 조인

고객

고객번호	이름
1	박지성
2	김연아
3	장미란
4	추신수

주문

주문번호	고객번호	판매가격
1	2	7000
2	1	13000
3	2	8000
4	1	13000
5	4	35000
6	5	22000
7	4	22000

고객 ⋈ 고객.고객번호=주문.고객번호 주문

고객번호	이름	주문번호	판매가격
1	박지성	2	13000
1	박지성	4	13000
2	김연아	1	7000
2	김연아	3	8000
3	장미란	NULL	NULL
4	추신수	5	35000
4	추신수	7	22000

① 왼쪽 외부조인

고객 ⋈ 고객.고객번호=주문.고객번호 주문

고객번호	이름	주문번호	판매가격
1	박지성	2	13000
1	박지성	4	13000
2	김연아	1	7000
2	김연아	3	8000
3	장미란	NULL	NULL
4	추신수	5	35000
4	추신수	7	22000
5	NULL	6	22000

② 완전 외부조인

고객 ⋈ 고객.고객번호=주문.고객번호 주문

이름	주문번호	고객번호	판매가격
김연아	1	2	7000
박지성	2	1	13000
김연아	3	2	8000
박지성	4	1	13000
추신수	5	4	35000
NULL	6	5	22000
추신수	7	4	22000

③ 오른쪽 외부조인

그림 2-27 외부조인의 예

## 4. 조인

### ❖ 외부조인과 세미조인

#### ■ 세미조인(semi join)

- 자연조인을 한 후 두 릴레이션 중 한쪽 릴레이션의 결과만 반환하며, 기호에서 닫힌 쪽 릴레이션의 튜플만 반환함.
- 형식 :  $R \bowtie_{(r, s)} S$

질의 2-10 마당서점의 고객 중 주문 내역이 있는 고객의 고객 정보를 보이시오.

고객  $\bowtie_{\text{고객.고객번호, 주문.고객번호}}$  주문

## 4. 조인

### ❖ 외부조인과 세미조인

고객

고객번호	이름	주소	핸드폰
1	박지성	영국 맨체스타	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	장미란	대한민국 강원도	000-7000-0001
4	추신수	미국 클리블랜드	000-8000-0001

주문

주문번호	고객번호	도서번호	판매가격	주문일자
1	2	1	7000	2014-07-01
2	1	2	13000	2014-07-03
3	2	5	8000	2014-07-03
4	1	2	13000	2014-07-04
5	4	4	35000	2014-07-05
6	5	3	22000	2014-07-07
7	4	3	22000	2014-07-07



고객 ⋈ (고객.고객번호, 주문.고객번호) 주문

고객번호	이름	주소	핸드폰
1	박지성	영국 맨체스타	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
4	추신수	미국 클리블랜드	000-8000-0001

그림 2-28 세미조인(왼쪽이 닫힌 경우)의 예

## 5. 디비전

- 릴레이션의 속성 값의 집합으로 연산을 수행함
- 형식 :  $R \div S$

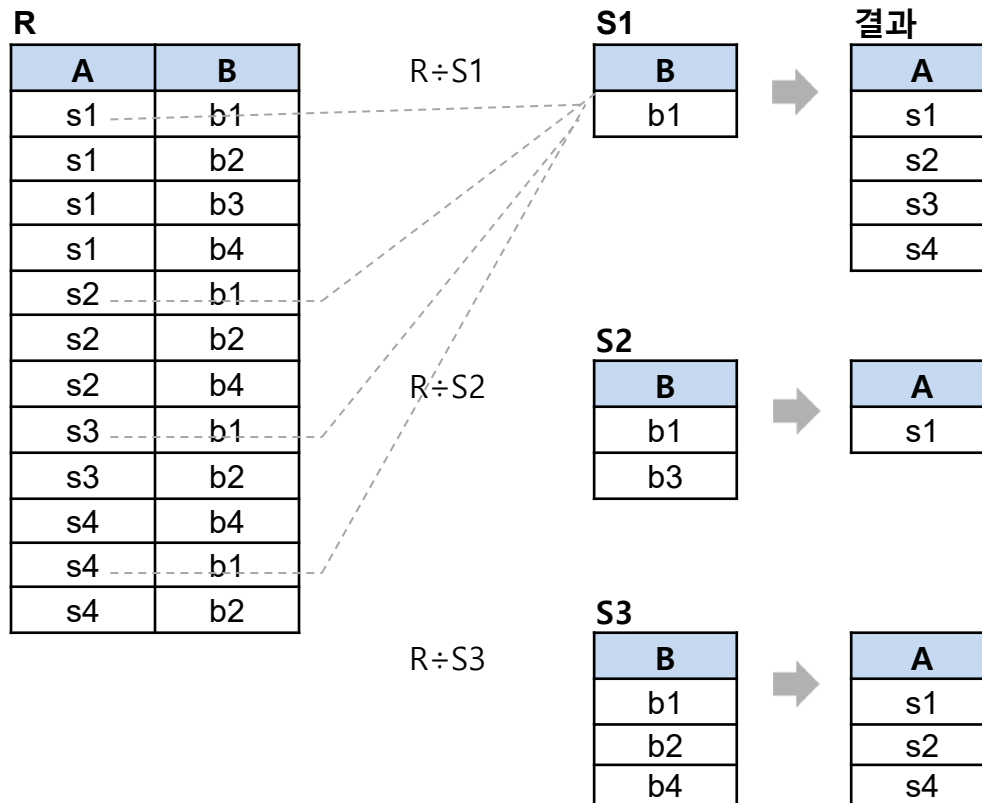


그림 2-29 디비전 연산의 예

## 6. 관계대수 예제

### ❖ 셀렉션, 프로젝션, 집합연산의 복합 사용

질의 2-11 마당서점의 도서 중 가격이 8,000원 이하인 도서이름과 출판사를 보이시오.

#### ■ 마당서점의 지점이 하나인 경우

$\pi_{\text{도서이름, 출판사}} (\sigma_{\text{가격} \leq 8000} \text{도서})$

도서

도서번호	도서이름	출판사	가격
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



$\sigma_{\text{가격} \leq 8000} (\text{도서})$

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
5	피겨 교본	굿스포츠	8000



$\pi_{\text{도서이름, 출판사}}$

도서이름	출판사
축구의 역사	굿스포츠
피겨 교본	굿스포츠

그림 2-30 단일 릴레이션에서 셀렉션, 프로젝션 연산의 복합 사용



## 6. 관계대수 예제

### ■ 마당서점의 지점이 둘 이상인 경우

$\pi_{\text{도서이름, 출판사}} ((\sigma_{\text{가격} \leq 8000} \text{도서A}) \cup (\sigma_{\text{가격} \leq 8000} \text{도서B}))$

도서A

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구아는 여자	나무수	13000
3	축구의 이해	대한미디어	22000

도서B

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
4	골프 바이블	대한미디어	35000
5	피겨 교본	굿스포츠	8000

$\sigma_{\text{가격} \leq 8000} (\text{도서A})$

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000

$\sigma_{\text{가격} \leq 8000} (\text{도서B})$

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
5	피겨 교본	굿스포츠	8000

U



도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
5	피겨 교본	굿스포츠	8000



$\pi_{\text{도서이름, 출판사}}$

도서이름	출판사
축구의 역사	굿스포츠
피겨 교본	굿스포츠

그림 2-31 두 개 이상의 릴레이션에서 선택, 프로젝션, 합집합 연산의 복합 사용

## 6. 관계대수 예제

### ❖ 카티전 프로덕트를 사용한 연산과 조인을 사용한 연산

질의 2-12 마당서점의 박지성 고객의 거래 내역 중 주문번호, 이름, 가격을 보이시오.

#### ■ 카티전 프로덕트를 사용한 연산

$\pi_{\text{주문.주문번호, 고객.이름, 주문.판매가격}} (\sigma_{\text{고객.고객번호=주문.고객번호 AND 고객.이름='박지성'}} (\text{고객} \times \text{주문}))$

고객

고객번호	이름	주민번호	주소	핸드폰
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001
2	김연아	900905-2222222	대한민국 서울	000-6000-0001
3	장미란	831009-2333333	대한민국 강원도	000-7000-0001
4	추신수	820713-1444444	미국 클리블랜드	000-8000-0001

주문

주문번호	고객번호	도서번호	판매가격	주문일자
1	1	1	7000	2014-07-01
2	1	2	13000	2014-07-03
3	2	5	8000	2014-07-03
4	3	2	13000	2014-07-04
5	4	4	35000	2014-07-05
6	1	3	22000	2014-07-07
7	4	3	22000	2014-07-07

고객 × 주문

고객번호	이름	주민번호	주소	핸드폰	주문번호	고객번호	도서번호	판매가격	주문일자
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001	1	1	1	7000	2014-07-01
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001	2	1	2	13000	2014-07-03
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001	3	2	5	8000	2014-07-03
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001	4	3	2	13000	2014-07-04
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001	5	4	4	35000	
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001	6	1	3	22000	
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001	7	4	3	22000	
2	김연아	900905-2222222	대한민국 서울	000-6000-0001	1	1	1	7000	2014-07-01
4	추신수	820713-1444444	미국 클리블랜드	000-8000-0001	5	4	4	35000	2014-07-05
4	추신수	820713-1444444	미국 클리블랜드	000-8000-0001	6	1	3	22000	2014-07-07
4	추신수	820713-1444444	미국 클리블랜드	000-8000-0001	7	4	3	22000	2014-07-07

결과  
생략

$\sigma$  고객.고객번호=주문.고객번호 AND 고객.이름='박지성'

고객번호	이름	주민번호	주소	핸드폰	주문번호	고객번호	도서번호	판매가격	주문일자
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001	1	1	1	7000	2014-07-01
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001	2	1	2	13000	2014-07-03
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001	6	1	3	22000	2014-07-07

$\pi$  주문.주문번호,고객.이름,주문.판매가격

주문번호	이름	판매가격
1	박지성	7000
2	박지성	13000
6	박지성	22000

그림 2-32 카티전 프로젝트를 사용한 연산

## 6. 관계대수 예제

### ❖ 카티전 프로덕트를 사용한 연산과 조인을 사용한 연산

#### ■ 조인을 사용한 연산

$\pi_{\text{주문번호, 이름, 판매가격}} (\sigma_{\text{이름='박지성'}} (\text{고객} \bowtie_{\text{고객.고객번호=주문.고객번호}} \text{주문}))$

#### ■ 카티전 프로덕트를 사용한 연산 (위 연산식과 동일함)

$\pi_{\text{주문.주문번호, 고객.이름, 주문.판매가격}} (\sigma_{\text{고객.고객번호=주문.고객번호 AND 고객.이름='박지성'}} (\text{고객} \times \text{주문}))$

고객

고객번호	이름	주민번호	주소	핸드폰
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001
2	김연아	900905-2222222	대한민국 서울	000-6000-0001
3	장미란	831009-2333333	대한민국 강원도	000-7000-0001
4	추신수	820713-1444444	미국 클리블랜드	000-8000-0001

주문

주문번호	고객번호	도서번호	판매가격	주문일자
1	1	1	7000	2014-07-01
2	1	2	13000	2014-07-03
3	2	5	8000	2014-07-03
4	3	2	13000	2014-07-04
5	4	4	35000	2014-07-05
6	1	3	22000	2014-07-07
7	4	3	22000	2014-07-07

고객  $\bowtie$  고객.고객번호=주문.고객번호 주문

고객번호	이름	주민번호	주소	핸드폰	주문번호	고객번호	도서번호	판매가격	주문일자
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001	1	1	1	7000	2014-07-01
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001	2	1	2	13000	2014-07-03
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001	6	1	3	22000	2014-07-07
2	김연아	900905-2222222	대한민국 서울	000-6000-0001	3	2	5	8000	2014-07-03
3	장미란	831009-2333333	대한민국 강원도	000-7000-0001	4	3	2	13000	2014-07-04
4	추신수	820713-1444444	미국 클리블랜드	000-8000-0001	5	4	4	35000	2014-07-05
4	추신수	820713-1444444	미국 클리블랜드	000-8000-0001	7	4	3	22000	2014-07-07

$\sigma_{\text{이름}='박지성'}$

고객번호	이름	주민번호	주소	핸드폰	주문번호	고객번호	도서번호	판매가격	주문일자
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001	1	1	1	7000	2014-07-01
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001	2	1	2	13000	2014-07-03
1	박지성	810225-1111111	영국 맨체스타	000-5000-0001	6	1	3	22000	2014-07-07

$\pi_{\text{주문번호,이름,판매가격}}$

주문번호	이름	판매가격
1	박지성	7000
2	박지성	13000
6	박지성	22000

그림 2-33 조인을 사용한 연산

# 연습문제

**6 릴레이션에서 특정 속성에 해당하는 열을 선택하는 데 사용하며, 릴레이션의 수직적 부분 집합을 반환하는 관계대수 연산자는?**

- ① projection    ② join
- ③ division        ④ selection

**7 릴레이션 C가 릴레이션 A(X, Y)와 B(Y, Z)를 자연조인한 결과일 때 다음 중 맞는 설명을 모두 고르시오.**

- ① C의 카디널리티는 A의 카디널리티보다 많다.
- ② C의 카디널리티는 A의 카디널리티보다 적다.
- ③ C의 차수는 A의 차수보다 많다.
- ④ C의 차수는 A의 차수보다 적다.
- ⑤ 모두 틀리다.

# 연습문제

12 다음 릴레이션에서 관계대수 식의 결과를 작성하시오.

R

A	B	C
a1	b1	c1
a2	b1	c1
a3	b1	c2
a4	b2	c3

S

C	D	E
c1	d2	e1
c1	d1	e2
c2	d3	e3
c3	d3	e3

(1)  $\sigma_{A=a2}(R)$

(2)  $\pi_{A, B}(R)$

(3)  $R \bowtie_{R.C=S.C} S$

13 다음 수강신청 관련 릴레이션에 대한 질의문을 관계대수식으로 표현하시오.

학생(학번, 이름, 전공, 학년)

수강(과목코드, 학번, 수강학기, 성적)

과목(과목코드, 과목이름, 강의실, 요일, 담당교수)

- (1) 과목코드가 1234이고, 성적이 A인 모든 학생의 학번을 보이시오.
- (2) 과목코드가 1234인 과목을 등록한 학생의 이름과 전공을 보이시오.
- (3) 과목 1234에 등록하지 않은 학생의 이름을 보이시오.(모든 학생이 수강신청에 참여했다고 가정)
- (4) 모든 과목에 등록한 학생의 이름을 보이시오. (디비전 연산자 사용)

# 연습문제

## 15 [판매원 데이터베이스] 다음 릴레이션을 보고 물음에 답하시오.

Salesperson(name, age, salary)

Order(number, custname, salesperson, amount)

Customer(name, city, industrytype)

- (1) 모든 판매원(Salesperson)의 이름을 보이시오.
- (2) 고객 '홍길동'의 주문을 수주한 판매원의 이름을 보이시오.
- (3) 주문이 있는 판매원의 이름을 보이시오.
- (4) 주문이 없는 판매원의 이름을 보이시오.
- (5) 고객 '홍길동'의 주문을 수주한 판매원의 나이를 보이시오.
- (6) 나이가 25살인 판매원에게 주문한 고객의 city 값을 보이시오.
- (7) 판매원의 이름과 그 판매원에게 주문을 한 고객의 이름을 보이시오. 단 주문이 없는 판매원도 포함하여 구한다.



1. 릴레이션
2. 릴레이션 스키마
3. 릴레이션 인스턴스
4. 관계 데이터베이스 시스템
5. 키
6. 무결성 제약조건
7. 참조 무결성 제약조건의 옵션
8. 관계대수
9. 선택
10. 프로젝션
11. 집합 연산
12. 조인
13. 디비전