

---

# DataBase



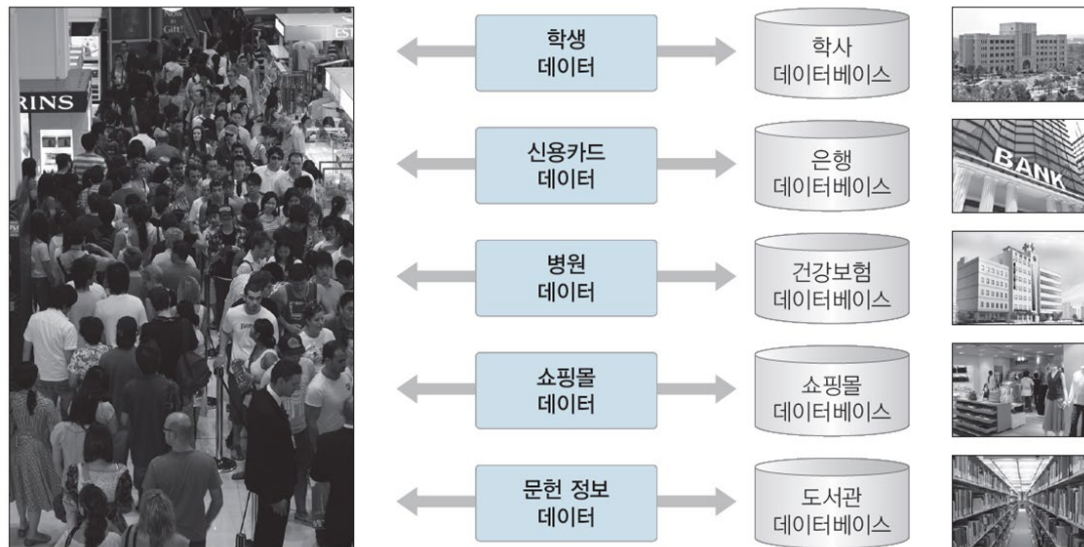
# DataBase

DataBase에 대한 이해

1. DataBase 개요Big Data 정의
2. 정보 시스템의 발전
3. DataBase 시스템의 구성
4. Data Model
5. DataBase 구조
6. 관계 Data Model의 개념

## 1. DataBase 개요

- ❖ 데이터 : 관찰의 결과로 나타난 정량적 혹은 정성적인 실제 값
- ❖ 정보 : 데이터에 의미를 부여한 것
- ❖ 지식 : 사물이나 현상에 대한 이해
- ❖ 데이터베이스 : 조직에 필요한 정보를 얻기 위해 논리적으로 연관된 데이터를 모아 구조적으로 통합해 놓은 것



## 1. DataBase 개요

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
<b>총매출액</b>			<b>500</b>
<b>합계</b>			<b>500</b>
<b>받은 돈</b>			<b>500</b>

---

[ 카 드 ] 500

---

소비자 중심 경영 인증기업(CCM)

---

« 카드 승인 »

---

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

### 이용안내

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

- 문의전화 ☎ 1588-7786 / 1544-7786

승차일 2013년 03월 10일(일)

**부산** → **서울**

Busan → Seoul

14:45 → 17:47


KTX 제310 열차 일반실 11호차 10C석

운임요금 52,900 할인금액 2,600 영수액 50,300

어른 신용 940040

승차자 변소현 예약자 변소현

인쇄일시 2013/03/08 12:17 반환번호 80003-0308-12248-10



## 1. DataBase 개요

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

## 1. DataBase 개요

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

구축이 쉬움	유형	검색 빈도	변경 빈도	데이터베이스 예	특징
↑ ↓ 구축이 어려움	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>

## 1. DataBase 개요

### ❶ 통합된 데이터(integrated data)

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

### ❷ 저장된 데이터(stored data)

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

### ❸ 운영 데이터(operational data)

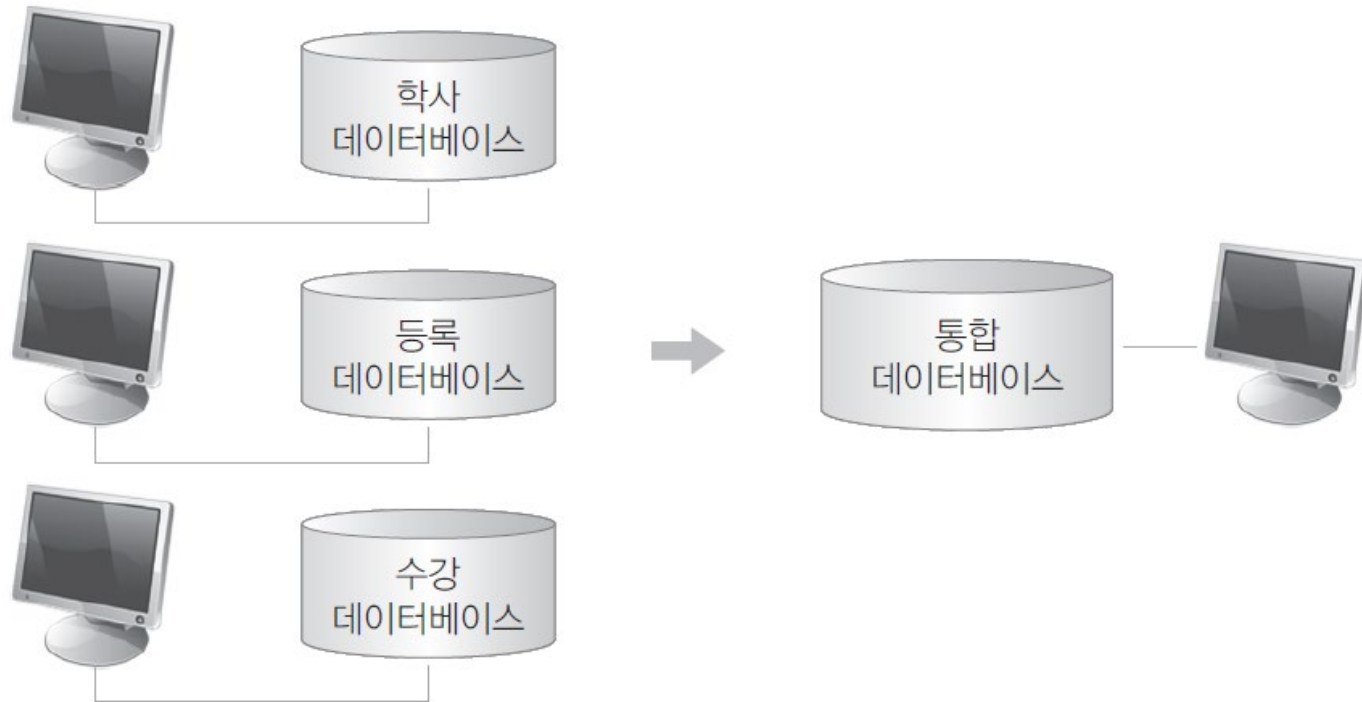
조직의 목적을 위해 사용되는 데이터를 의미한다. 즉 업무를 위한 검색을 할 목적으로 저장된 데이터

### ❹ 공용 데이터(shared data)

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



## 1. DataBase 개요



## 1. DataBase 개요

### 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.

### ❷ 계속적인 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.

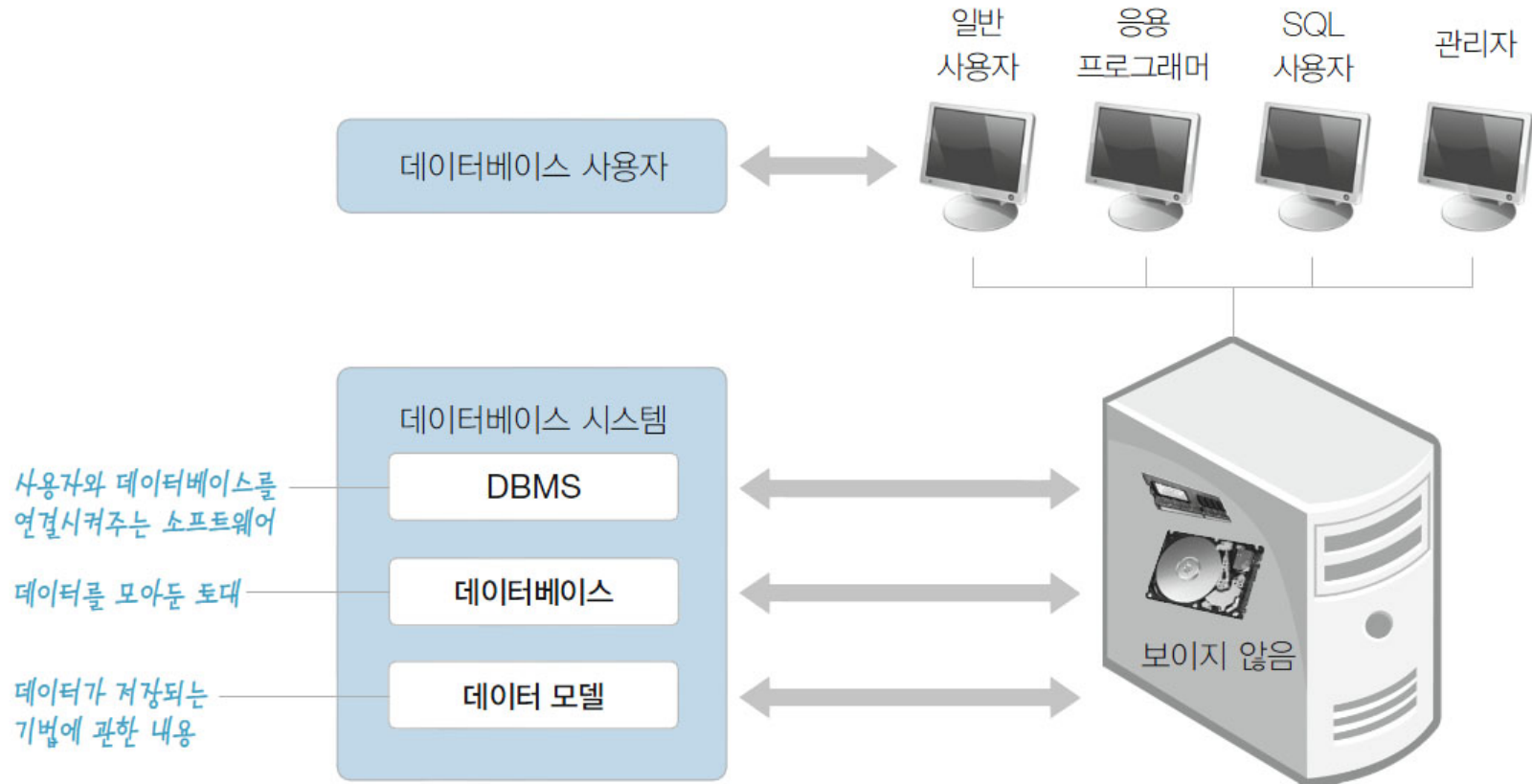
### ❸ 동시 공유(concurrent sharing)

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

### ❹ 내용에 따른 참조(reference by content)

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

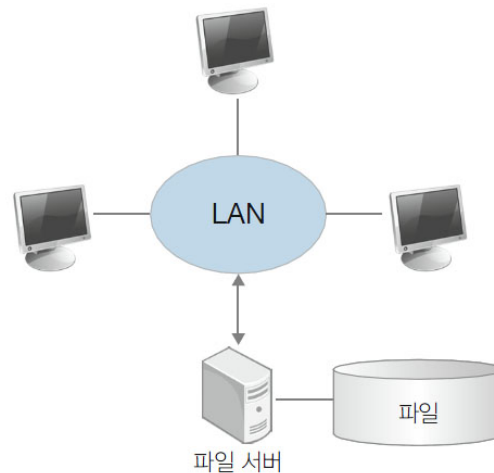
## 1. DataBase 개요



## 2. 정보 시스템의 발전

### ❶ 파일 시스템

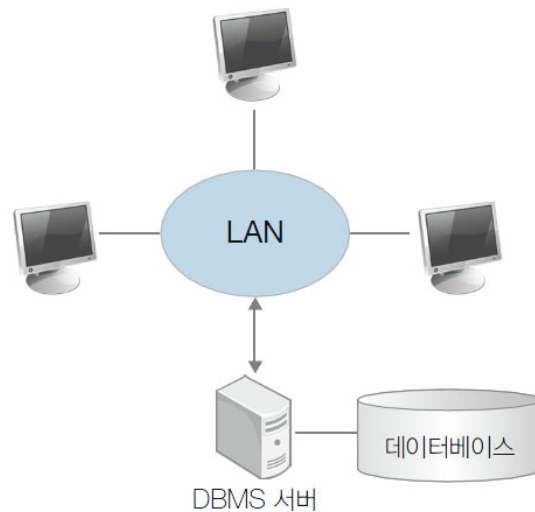
- 데이터를 파일 단위로 파일 서버에 저장
- 각 컴퓨터는 LAN을 통하여 파일 서버에 연결되어 있고, 파일 서버에 저장된 데이터를 사용하기 위해 각 컴퓨터의 응용 프로그램에서 열기/닫기(open/close)를 요청
- 각 응용 프로그램이 독립적으로 파일을 다루기 때문에 데이터가 중복 저장될 가능성이 있음
- 동시에 파일을 다루기 때문에 데이터의 일관성이 훼손될 수 있음



## 2. 정보 시스템의 발전

### ② 데이터베이스 시스템

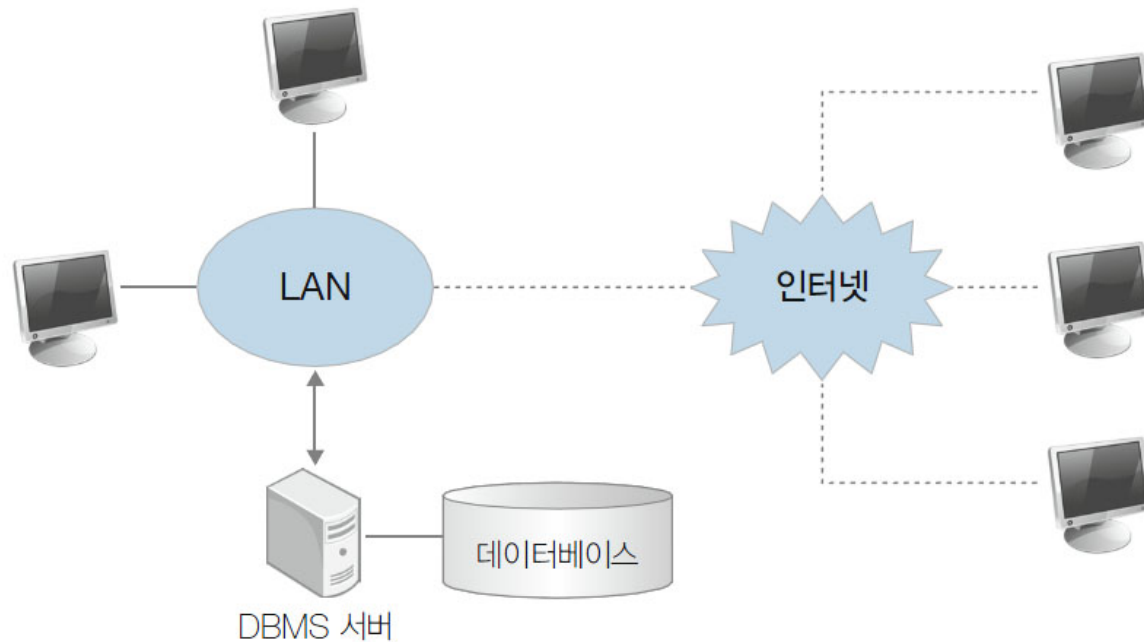
- DBMS를 도입하여 데이터를 통합 관리하는 시스템
- DBMS가 설치되어 데이터를 가진 쪽을 서버(server), 외부에서 데이터 요청하는 쪽을 클라이언트(client)라고 함
- DBMS 서버가 파일을 다루며 데이터의 일관성 유지, 복구, 동시 접근 제어 등의 기능을 수행
- 데이터의 중복을 줄이고 데이터를 표준화하며 무결성을 유지함



## 2. 정보 시스템의 발전

### ③ 웹 데이터베이스 시스템

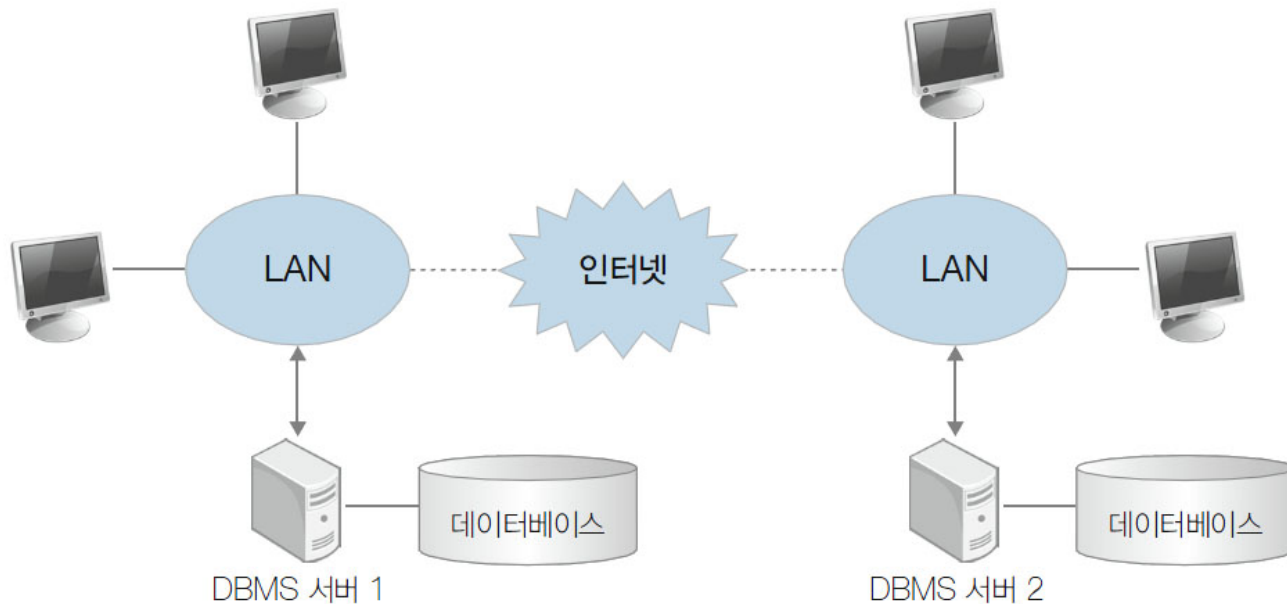
- 데이터베이스를 웹 브라우저에서 사용할 수 있도록 서비스하는 시스템
- 불특정 다수 고객을 상대로 하는 온라인 상거래나 공공 민원 서비스 등에 사용됨



## 2. 정보 시스템의 발전

### ④ 분산 데이터베이스 시스템

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



## 2. 정보 시스템의 발전

소매상(실체 시스템)



소기업(파일 시스템)



중견기업(데이터베이스)



대기업(인터넷 통신)



종합그룹(인터넷 통신)

- 1970년대
- 정보 시스템 없음
- 수작업으로 회계 업무

- 1980년대
- 파일 시스템 사용
- 파일을 이용한 응용 프로그램으로 업무 처리

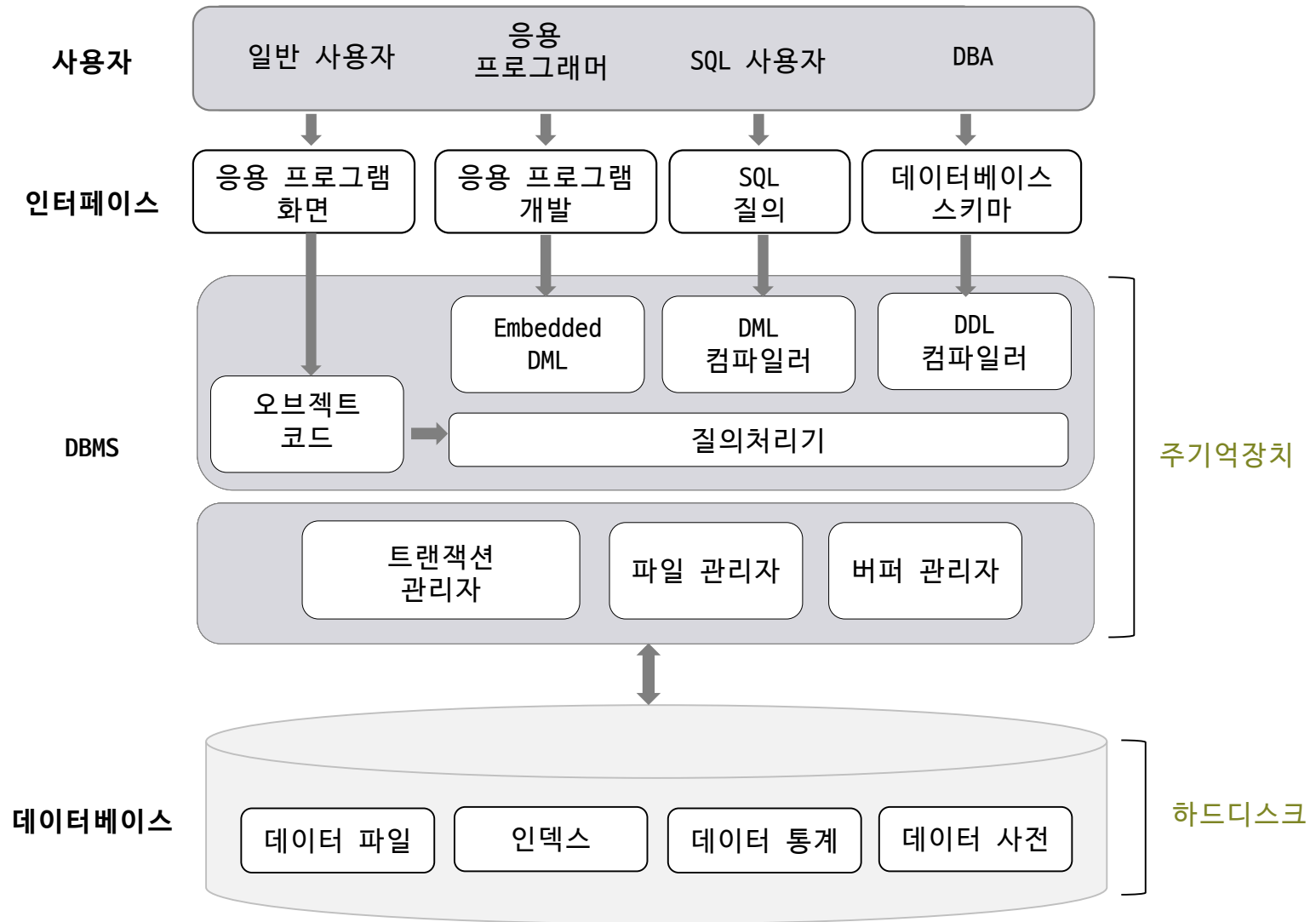
- 1990년대
- 정보 시스템, 데이터베이스 시스템 사용
- DBMS를 이용하여 업무 처리
- 정보 공유, 실시간 서비스, LAN 기술

- 2000년대
- 정보 시스템, 웹 데이터베이스 시스템, 인터넷 사용
- 인터넷 쇼핑몰을 개설하여 온라인 상거래 실시
- 실시간 서비스, 웹 브라우저 기술

- 2010년대
- 정보 시스템, 분산 데이터베이스 시스템, 인터넷 사용
- 고객 서비스 및 내부 업무를 인터넷으로 처리
- 대규모 응용 시스템에 사용



### 3. DataBase 시스템의 구성



### 3. DataBase 시스템의 구성

#### ■ 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

### 3. DataBase 시스템의 구성

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

### 3. DataBase 시스템의 구성

#### ■ 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

### 3. DataBase 시스템의 구성

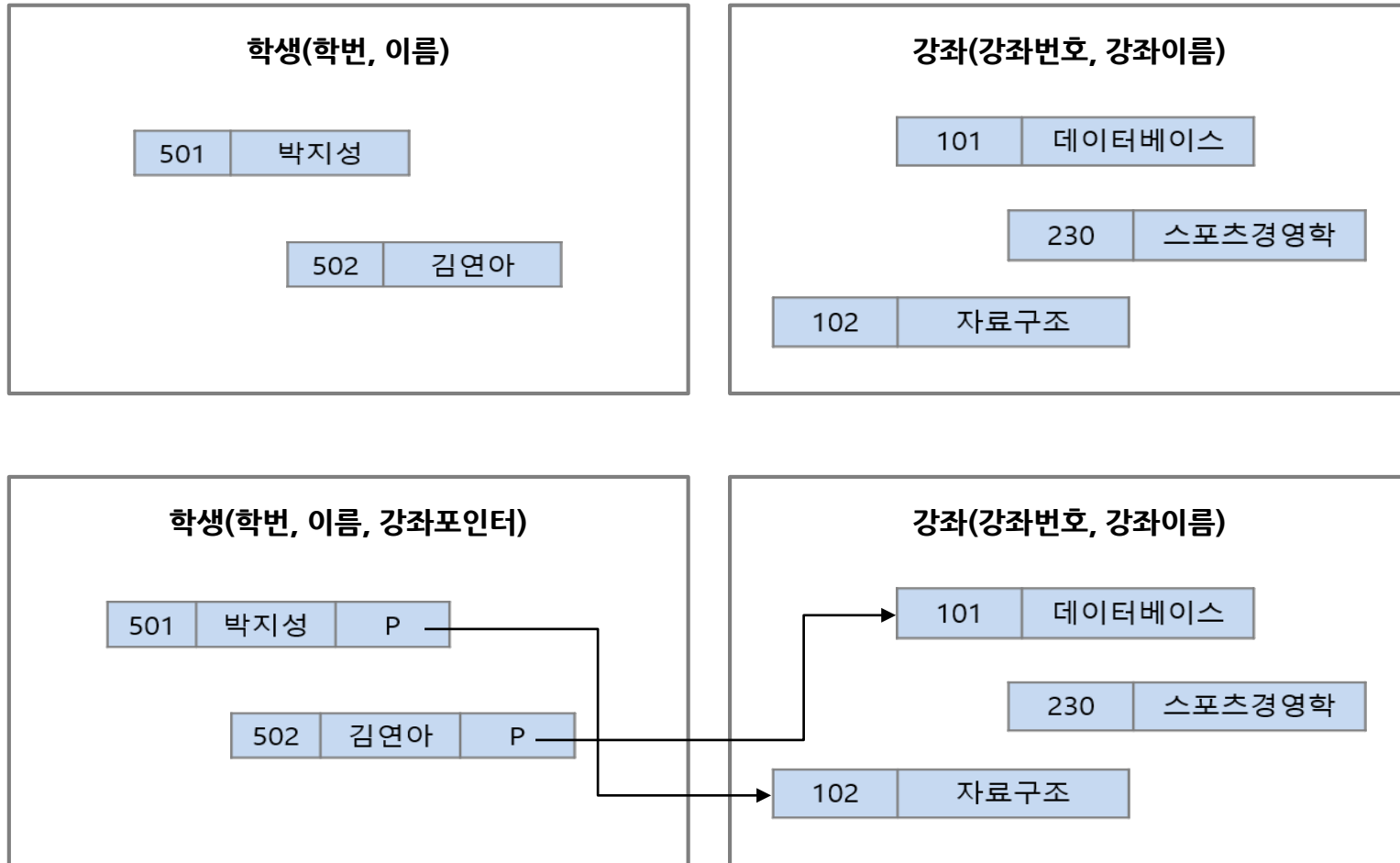
	SQL 언어	프로그래밍 능력	DBMS 지식	데이터 구성
일반 사용자	×	×	×	×
SQL 사용자	◎	×	○	○
응용 프로그래머	◎	◎	○	○
데이터베이스 관리자	◎	○	◎	◎

### 3. DataBase 시스템의 구성

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. Data Model

### ① 포인터 사용 : 계층 데이터 모델, 네트워크 데이터 모델



## 4. Data Model

### ② 속성 값 사용 : 관계 데이터 모델

학생(학번, 이름)

501	박지성
-----	-----

502	김연아
-----	-----

강좌(강좌번호, 강좌이름)

101	데이터베이스
-----	--------

230	스포츠경영학
-----	--------

102	자료구조
-----	------

학생(학번, 이름, 강좌번호)

501	박지성	102
-----	-----	-----

401	김연아	101
-----	-----	-----

강좌(강좌번호, 강좌이름)

101	데이터베이스
-----	--------

230	스포츠경영학
-----	--------

102	자료구조
-----	------



## 4. Data Model

### ③ 객체식별자 사용 : 객체 데이터 모델

학생(학번, 이름)

501	박지성
-----	-----

502	김연아
-----	-----

강좌(강좌번호, 강좌이름)

101	데이터베이스
-----	--------

230	스포츠경영학
-----	--------

102	자료구조
-----	------

학생(학번, 이름, objectid)

501	박지성	oid
-----	-----	-----

401	김연아	oid
-----	-----	-----

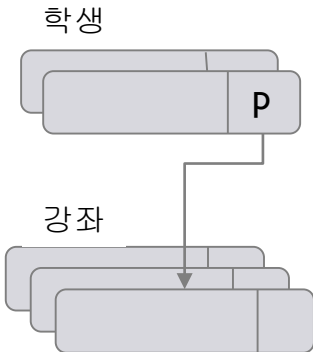
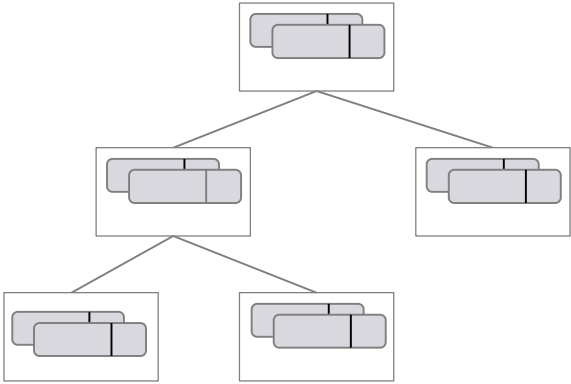
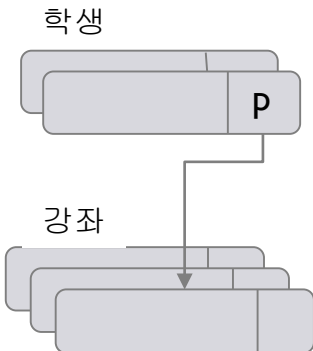
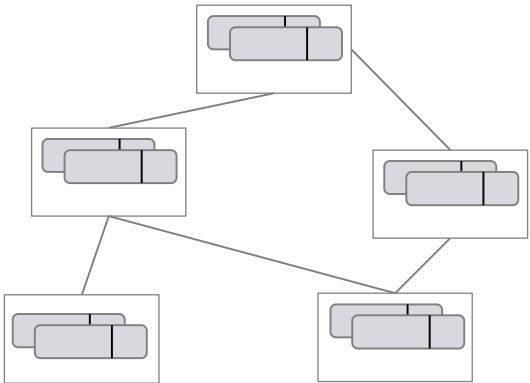
강좌(강좌번호, 강좌이름)

101	데이터베이스
-----	--------

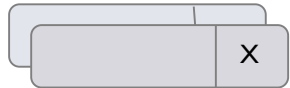
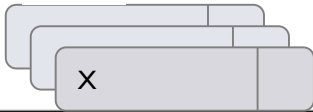
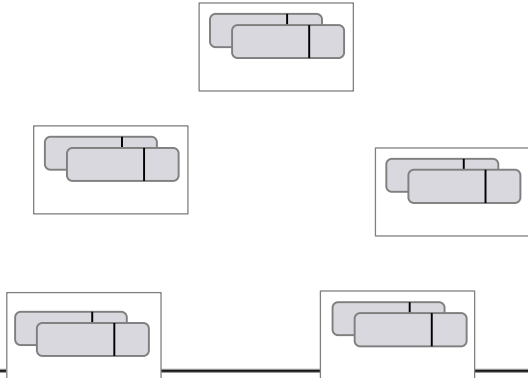
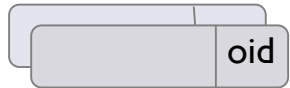

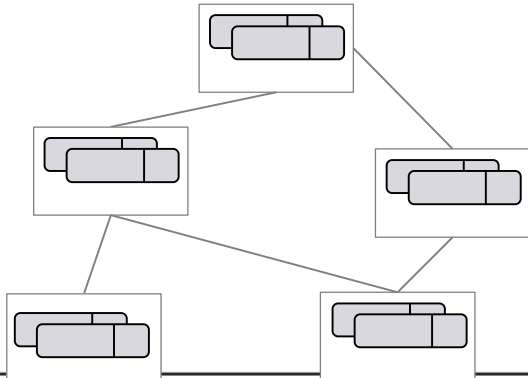
230	스포츠경영학
-----	--------

102	자료구조
-----	------

## 4. Data Model

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

## 4. Data Model

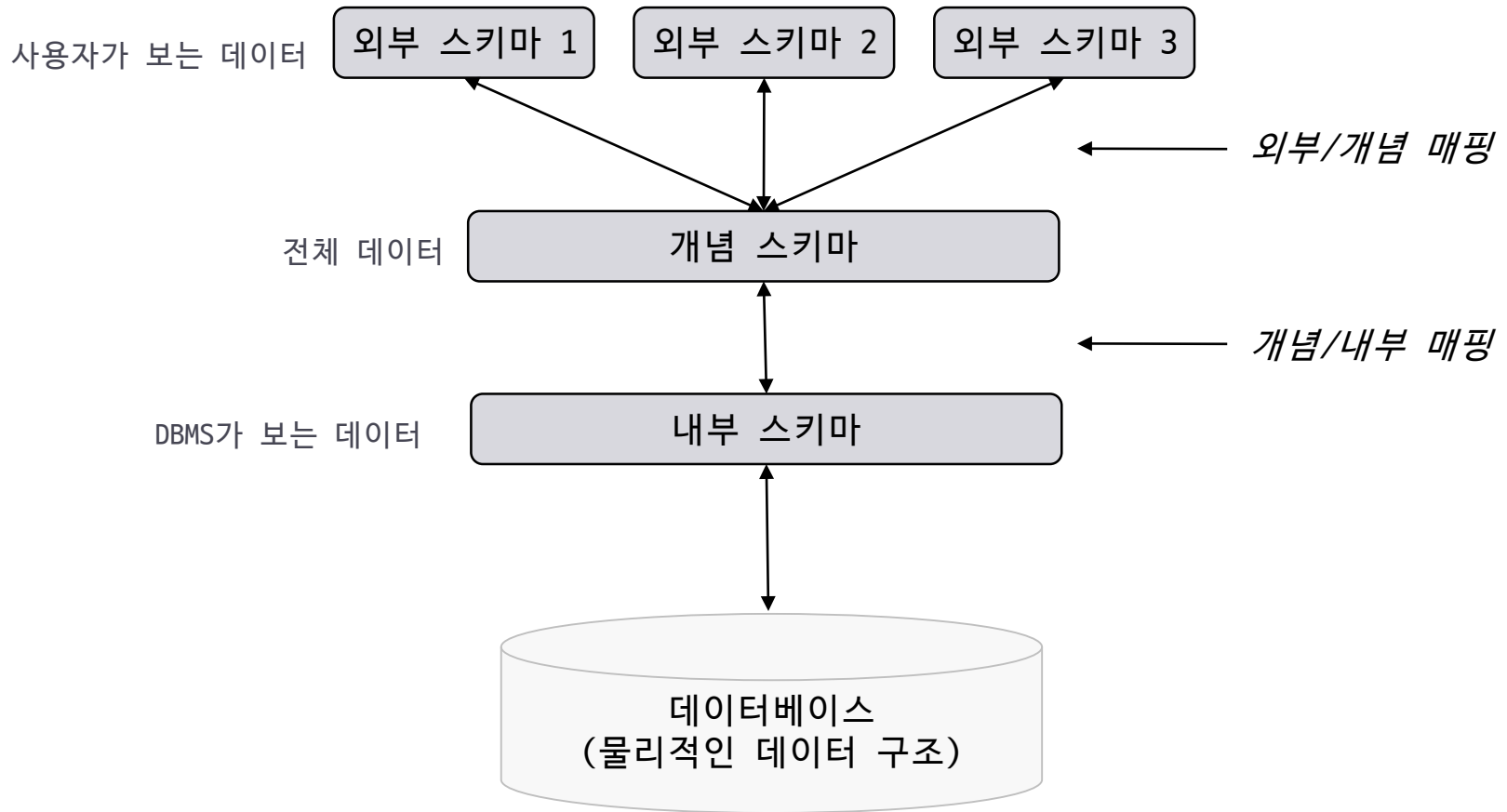
데이터 모델	관계의 표현	데이터 구성
관계 데이터 모델 (속성 값 사용)	<p>학생</p>  <p>강좌</p> 	
객체 데이터 모델 (객체식별자 사용)	<p>학생</p>  <p>강좌</p>  <p>객체 번호 oid</p>	

## 4. Data Model

데이터 모델	1970년대	1980년대	1990년대	2000년대	2010년대
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. DataBase 구조



**ANSI의 3단계 데이터베이스 구조**

## 5. DataBase 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. DataBase 구조

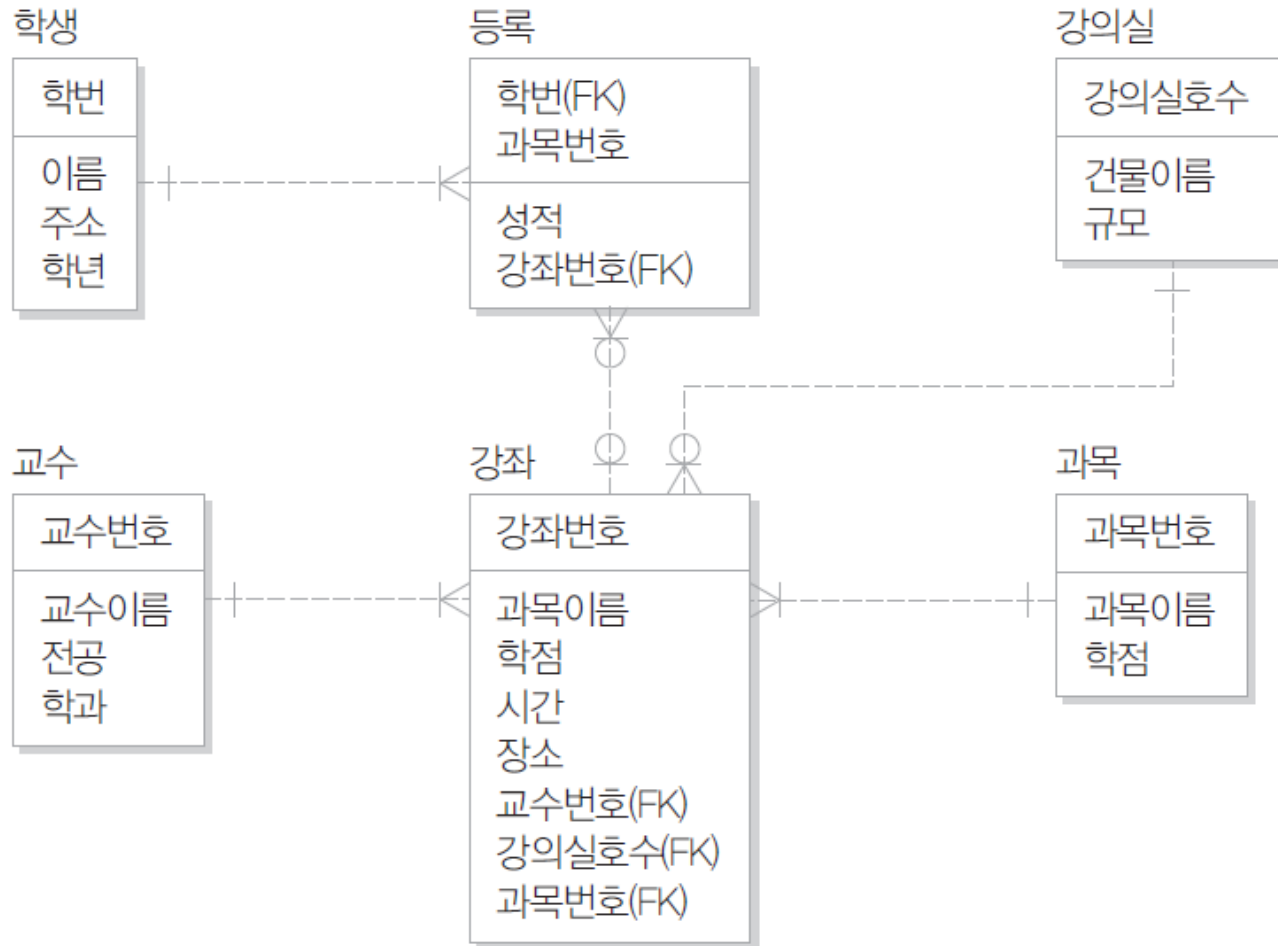
### ■ 외부/개념 매핑

- 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. DataBase 구조



수강신청 데이터베이스의 개념 스키마

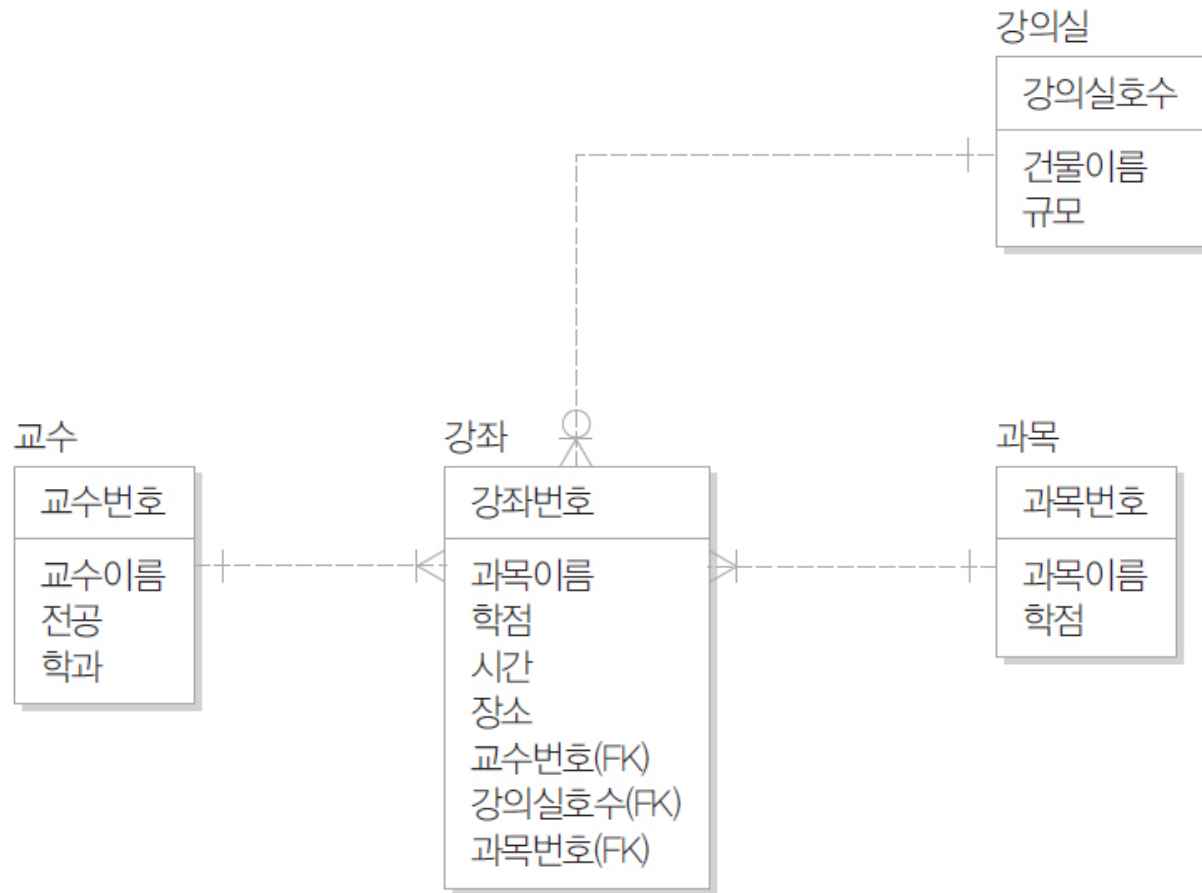


## 5. DataBase 구조



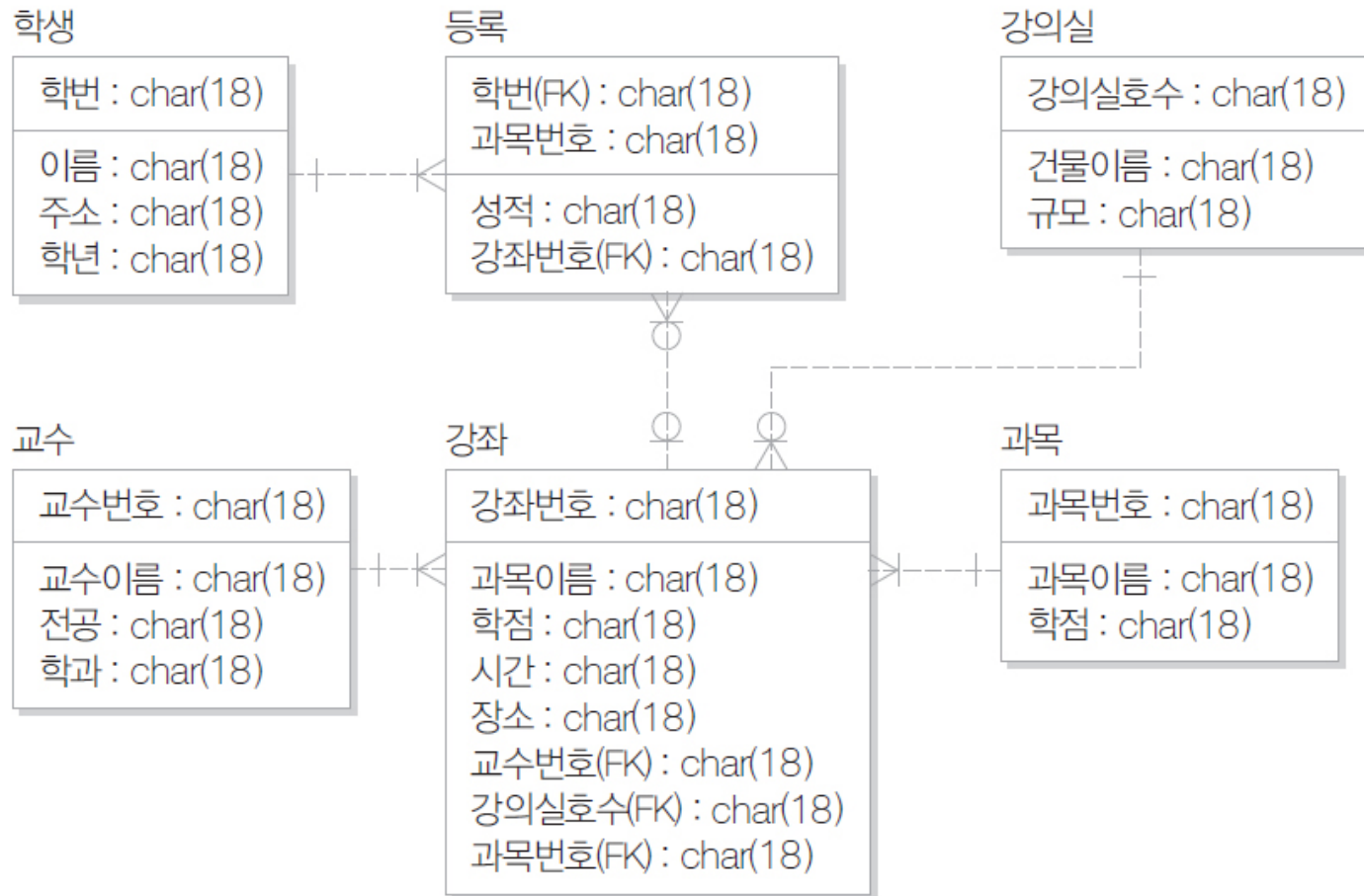
수강등록 담당 부서에서 필요한 데이터베이스(외부 스키마1)

## 5. DataBase 구조



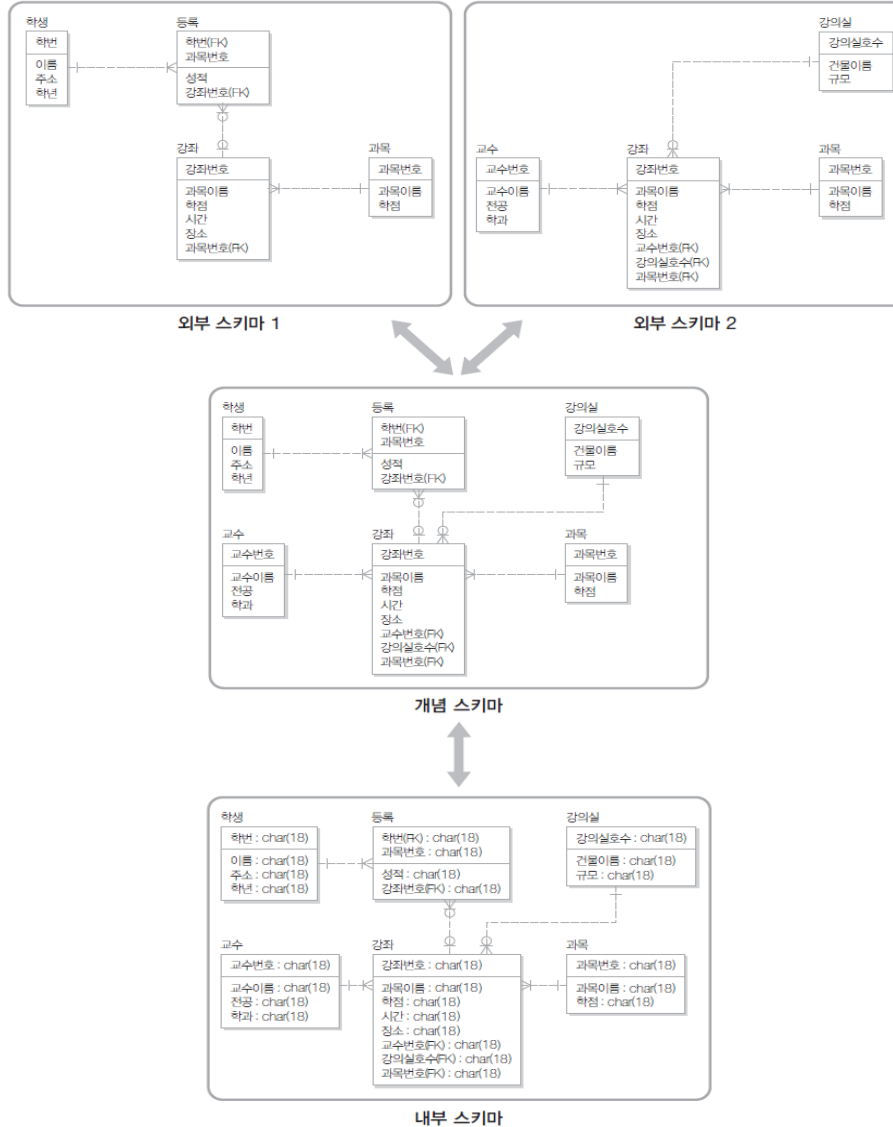
시간표 담당 부서에서 필요한 데이터베이스(외부 스키마2)

## 5. DataBase 구조



### 수강신청 데이터베이스(내부 스키마)

## 5. DataBase 구조



수강신청 데이터베이스의 3단계 구조

## 5. DataBase 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

## 6. 관계 Data Model의 개념


### ■ 릴레이션(relation) : 행과 열로 구성된 테이블

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	

## 6. 관계 Data Model의 개념

### ■ 릴레이션이란?

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

### 데이터와 테이블(릴레이션)

도서번호 = {1,2,3,4,5}

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

출판사 = {굿스포츠, 나무수, 대한미디어}

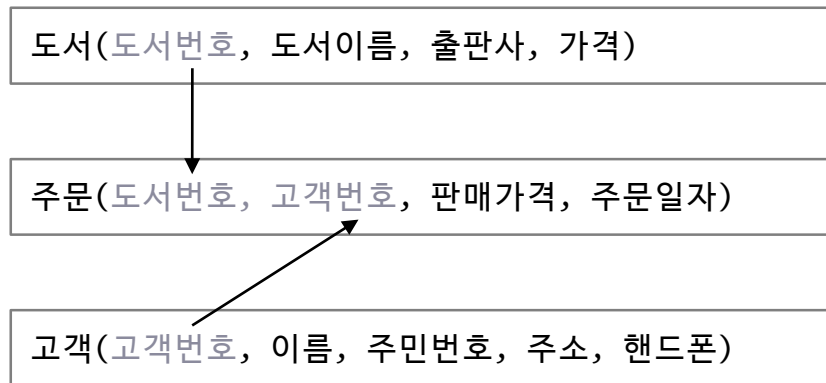
가격 = {7000, 13000, 22000, 35000, 8000}

→ 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.

## 6. 관계 Data Model의 개념

### ■ 관계(relationship)

- ① 릴레이션 내에서 생성되는 관계 : 릴레이션 내 데이터들의 관계
- ② 릴레이션 간에 생성되는 관계 : 릴레이션 간의 관계



릴레이션 간의 관계



## 6. 관계 Data Model의 개념

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)

도서 릴레이션

## 6. 관계 Data Model의 개념

### ■ 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)

## 6. 관계 Data Model의 개념

### ■ 인스턴스 요소

- 튜플(tuple) : 릴레이션의 행 → 튜플이 가지는 속성의 개수는 릴레이션 스키마의 차수와 동일하고, 릴레이션 내의 모든 튜플들은 서로 중복되지 않아야 함
- 카디널리티(cardinality) : 튜플의 수

### 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)

## 6. 관계 Data Model의 개념

- 속성은 단일 값을 가진다

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

- 속성은 서로 다른 이름을 가진다

속성은 한 릴레이션에서 서로 다른 이름을 가져야만 함.

- 한 속성의 값은 모두 같은 도메인 값을 가진다

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

- 속성의 순서는 상관없다

속성의 순서가 달라도 릴레이션 스키마는 같음.

예) 릴레이션 스키마에서 (이름, 주소) 순으로 속성을 표시하거나 (주소, 이름) 순으로 표시하여도 상관 없음.

- 릴레이션 내의 중복된 튜플은 허용하지 않는다

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

- 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.

## 6. 관계 Data Model의 개념

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

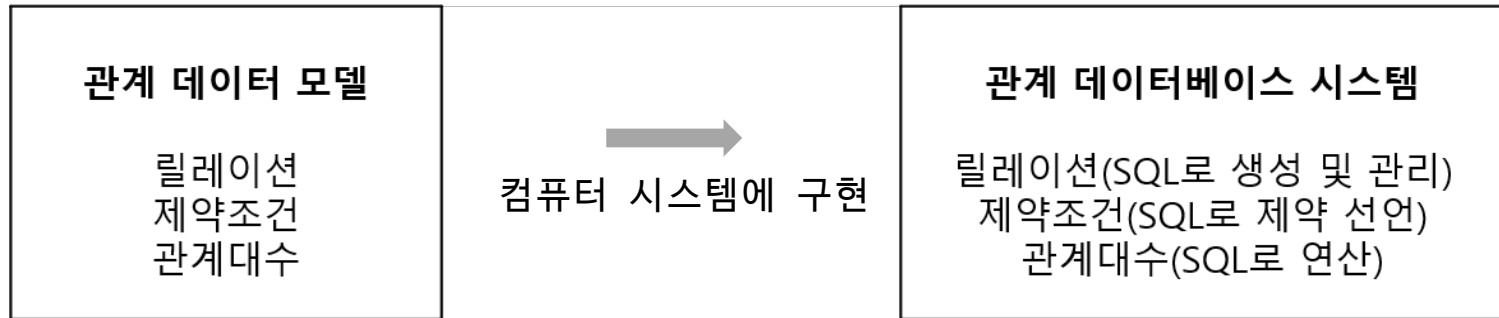
릴레이션의 특징에 위배된 경우

## 6. 관계 Data Model의 개념

- 관계 데이터 모델은 데이터를 2차원 테이블 형태인 릴레이션으로 표현함.

릴레이션에 대한 제약조건(constraints)과 관계 연산을 위한 관계대수 (relational

algebra)를 정의함.



관계 데이터베이스 시스템

## 6. 관계 Data Model의 개념

- 특정 튜플을 식별할 때 사용하는 속성 혹은 속성의 집합임.
- 릴레이션은 중복된 튜플을 허용하지 않기 때문에 각각의 튜플에 포함된 속성들 중 어느 하나(혹은 하나 이상)는 값이 달라야 함. 즉 키가 되는 속성(혹은 속성의 집합)은 반드시 값이 달라서 튜플들을 서로 구별할 수 있어야 함.
- 키는 릴레이션 간의 관계를 맺는 데도 사용됨.

## 6. 관계 Data Model의 개념

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



## 6. 관계 Data Model의 개념

### ■ 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

## 6. 관계 Data Model의 개념

### ■ 튜플을 유일하게 식별할 수 있는 속성의 최소 집합

#### (주문 릴레이션 예)

- 고객번호 : 한 명의 고객이 여러 권의 도서를 구입할 수 있으므로 후보키가 될 수 없음. 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. 참고로 이렇게 2개 이상의 속성으로 이루어진 키를 복합키(composite key)라고 함.

## 6. 관계 Data Model의 개념

- 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)

## 6. 관계 Data Model의 개념

- 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

대리키를 사용하도록 변경된 주문 릴레이션

## 6. 관계 Data Model의 개념

- 대체키(alternate key)는 기본키로 선정되지 않은 후보키를 말함.
- 고객 릴레이션의 경우 고객번호와 주민번호 중 고객번호를 기본키로 정하면 주민번호가 대체키가 됨.
- 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.

## 6. 관계 Data Model의 개념

고객

고객번호	이름	주민번호	주소	핸드폰
1	박지성	810101-1111111	영국 맨체스터	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

릴레이션 간의 참조 관계

## 6. 관계 Data Model의 개념

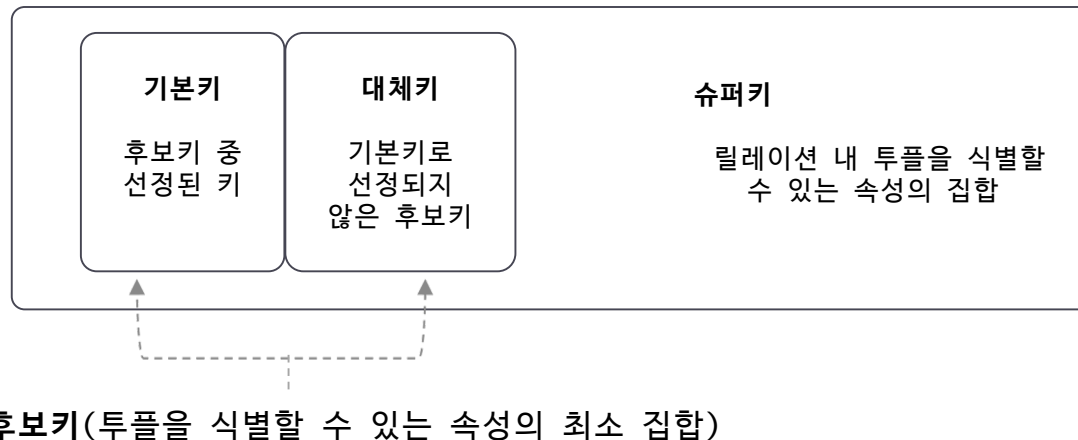
- 외래키 사용 시 참조하는 릴레이션과 참조되는 릴레이션이 꼭 다른 릴레이션일 필요는 없음. 즉 자기 자신의 기본키를 참조할 수도 있음.

참조

기본키      외래키

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

### 멘토 릴레이션



## 6. 관계 Data Model의 개념

- 데이터 무결성(integrity, 無缺性)은 데이터베이스에 저장된 데이터의 일관성과 정확성을 지키는 것을 말함.

- 도메인 무결성 제약조건

도메인 제약(domain constraint)이라고도 하며, 릴레이션 내의 튜플들이 각 속성의 도메인에 지정된 값만을 가져야 한다는 조건임. 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.



## 6. 관계 Data Model의 개념

구분	도메인	키	
	도메인 무결성 제약조건	개체 무결성 제약조건	참조 무결성 제약조건
제약 대상	속성	튜플	속성과 튜플
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>• 튜플 삽입, 수정 시 제약 사항 우선 확인</li> </ul>	<ul style="list-style-type: none"> <li>• 튜플 삽입/수정 시 제약 사항 우선 확인</li> </ul>	<ul style="list-style-type: none"> <li>• 튜플 삽입/수정 시 제약사항 우선 확인</li> <li>• 부모 릴레이션의 튜플 수정/삭제 시 제약사항 우선 확인</li> </ul>

## 6. 관계 Data Model의 개념

- 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

### 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

개체 무결성 제약조건의 수행 예(기본키 충돌 및 NULL 값 삽입)

## 6. 관계 Data Model의 개념

### ■ 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

학생관리 데이터베이스

## 6. 관계 Data Model의 개념

### ■ 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

- ① 즉시 작업을 중지
- ② 자식 릴레이션의 관련 튜플을 삭제
- ③ 초기에 설정된 다른 어떤 값으로 변경
- ④ NULL 값으로 설정

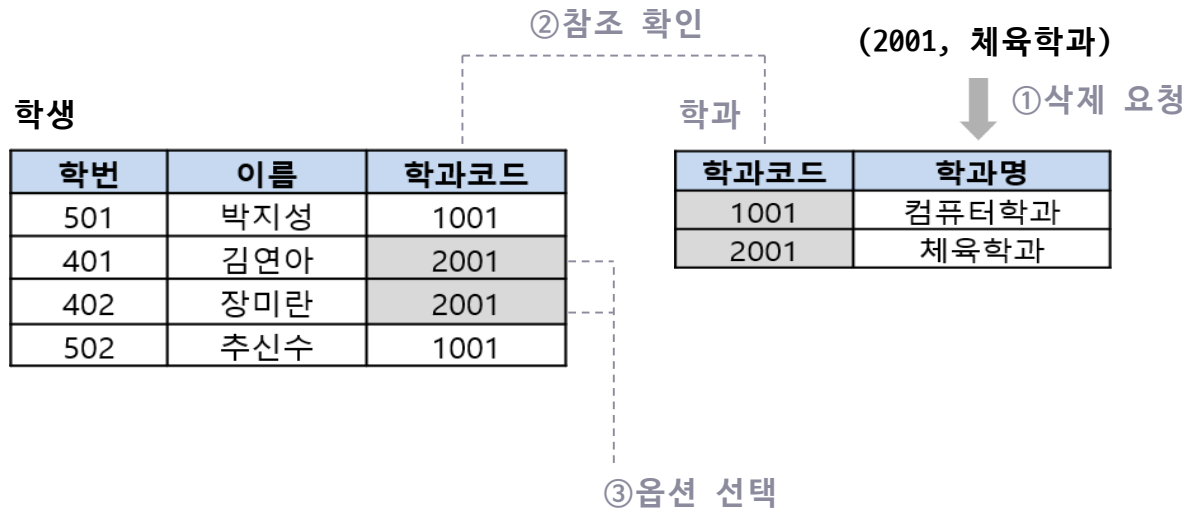
### ■ 수정

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

## 6. 관계 Data Model의 개념

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

## 6. 관계 Data Model의 개념



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

참조 무결성 제약조건에서 부모 릴레이전의 튜플을 삭제할 경우

---

## 참고

- 데이터베이스 개론과 실습(박우창, 남송휘, 이현룡) - 한빛아카데미