



Chapter 01

데이터베이스 시스템

MySQL로 배우는 데이터베이스 개론과 실습

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.
- ❖ 정보 시스템의 발전 과정을 통해 데이터베이스 시스템의 등장 배경을 이해한다.
- ❖ 파일 시스템과 DBMS를 비교하여 DBMS의 장점을 이해한다.
- ❖ 데이터베이스 시스템의 구성 요소를 알아본다.

01 데이터베이스와 데이터베이스 시스템

1. 데이터, 정보, 지식
2. 데이터베이스의 활용
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. 데이터베이스의 활용

❖ 데이터베이스

- 다음과 같이 일상생활의 거의 모든 곳에서 생성됨

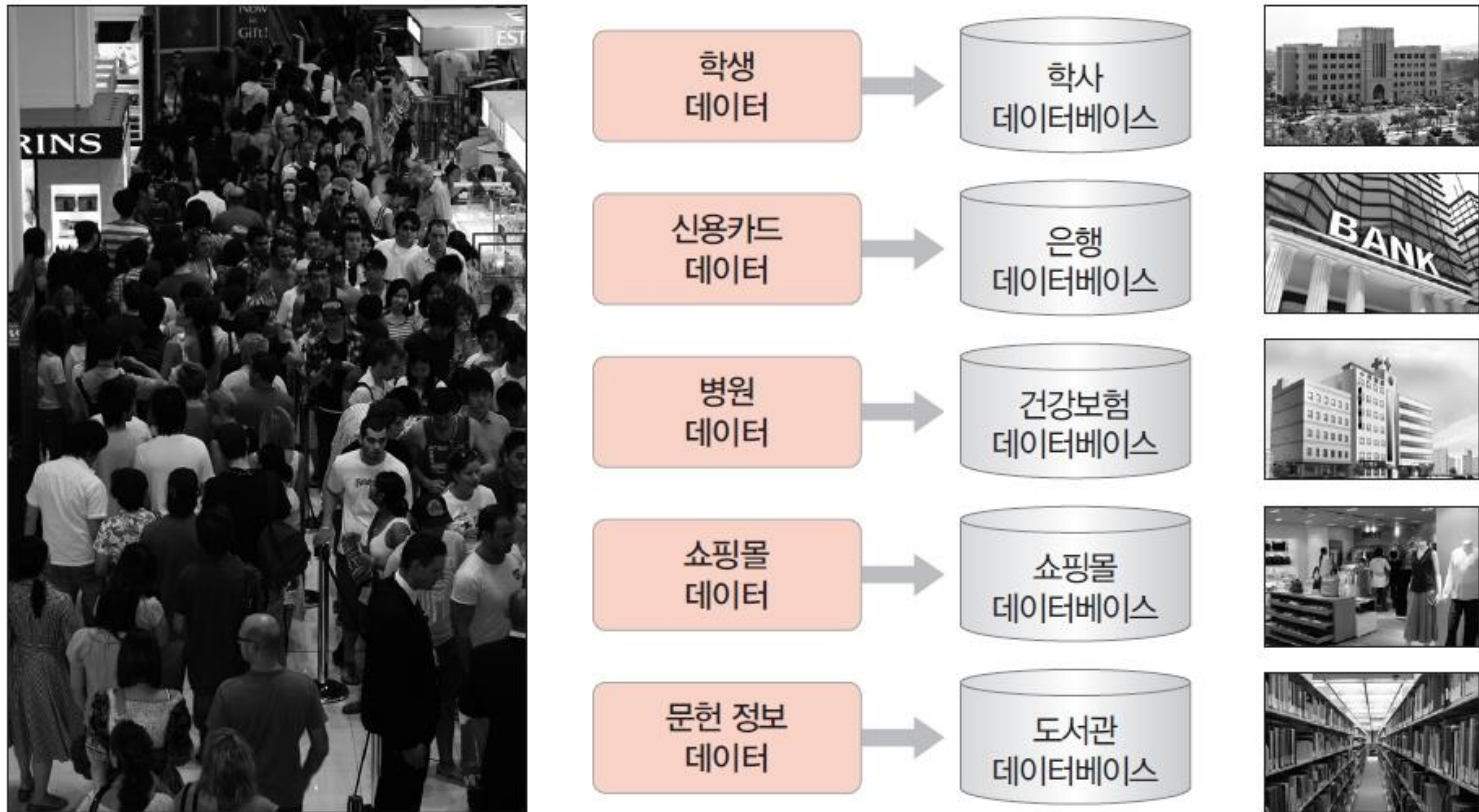


그림 1-2 일상생활에서 생성되는 데이터베이스

2. 데이터베이스의 활용

- 예) 패스트푸드 체인점에서 500원짜리 소프트아이스크림을 살 경우
 - 체인점의 판매 데이터베이스에는 체인점 이름, 판매대 번호, 판매자, 판매 시간, 금액, 결제 방법 등의 데이터가 실시간으로 저장


신대방삼거리역/ 대표 : 김 숙 (10010274) T:02-011-3335 담당 : 정 주 POS NO : 01 일사 : 2022-03-24 18:12:38		
QTY	D/C	AMT
소프트 아이스크림	1 0	500
과세금액		273
부가세		227
총매출액		500
합계		500
출금돈		500
[카 드]		500
소비자 중심 경영 인증기업(CCM)		
<< 카드 승인 >>		
[카드번호]	940951*****271	
[승인금액]	500	
[승인번호]	302517480 [일시별]	

그림 1-3 패스트푸드 체인점 데이터베이스

2. 데이터베이스의 활용

- 데이터베이스 시스템은 데이터의 검색과 변경 작업을 주로 수행
 - 변경이란 시간에 따라 변하는 데이터 값을 데이터베이스에 반영하기 위해 수행하는 삽입, 삭제, 수정 등의 작업을 말함 - 이러한 검색·변경 빈도에 따라 시스템 구축의 난이도가 결정
- 다음은 검색·변경 빈도에 따른 데이터베이스 유형

표 1-1 검색·변경 빈도에 따른 데이터베이스 유형

유형	검색 빈도	변경 빈도	데이터베이스 구축	특징
유형 1	적다	적다	 쉬움 어려움	• 검색이 많지 않아 데이터베이스를 구축할 필요 없음 • 보존 가치가 있는 경우에 구축함 예 공공 정보 데이터베이스
유형 2	많다	적다		• 사용자 수 보통임 • 검색은 많지만, 데이터에 대한 변경은 적음 예 도서 데이터베이스
유형 3	적다	많다		• 예약 변경/취소 등 데이터 변경은 많지만, 검색은 적음 • 실시간 검색 및 변경이 중요함 예 비행기 예약 데이터베이스
유형 4	많다	많다		• 사용자 수 많음 • 검색도 많고, 거래로 인한 변경도 많음 예 증권 데이터베이스

3. 데이터베이스의 개념 및 특징

❖ 데이터베이스 개념

- 여러 사람이 공용으로 사용하기 위해 통합하고 저장한 운영 데이터의 집합
- ❶ 통합된 데이터(integrated data)
- ❷ 저장된 데이터(stored data)
- ❸ 운영 데이터(operational data)
- ❹ 공용 데이터(shared data)

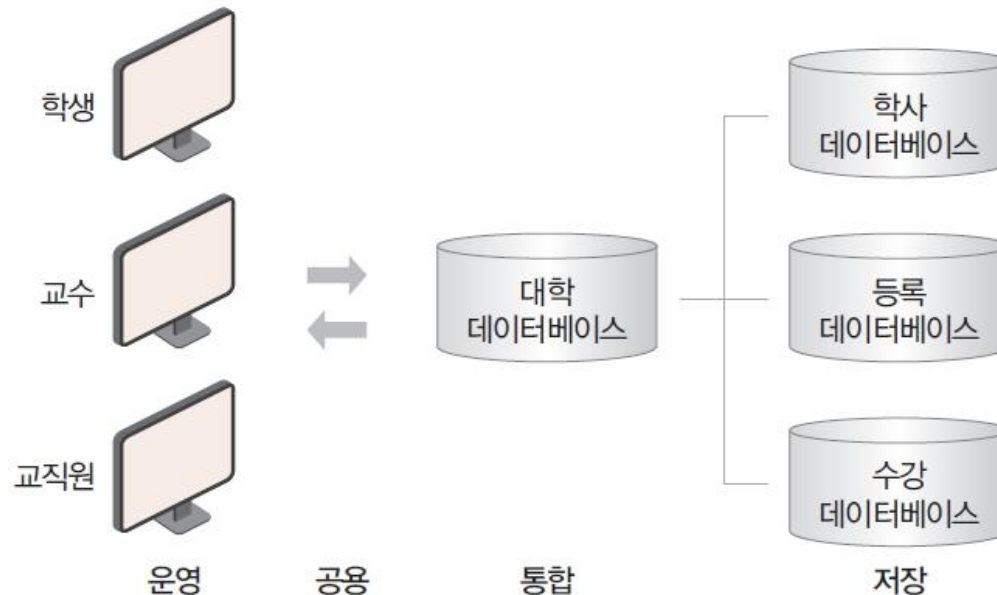


그림 1-4 데이터베이스의 개념

3. 데이터베이스의 개념 및 특징

❖ 데이터베이스 특징

❶ 실시간 접근성(real time accessibility)

데이터베이스는 실시간으로 서비스된다.

❷ 계속적인 변화(continuous change)

데이터 값은 시간에 따라 항상 바뀐다.

❸ 동시 공유(concurrent sharing)

데이터베이스는 서로 다른 업무 또는 여러 사용자에게 동시에 공유된다.

❹ 내용에 따른 참조(reference by content)

데이터베이스에 저장된 데이터는 데이터의 물리적인 위치가 아니라 데이터 값에 따라 참조된다

※ 자세한 내용은 책을 참고하세요.

4. 데이터베이스 시스템의 구성

❖ 데이터베이스 시스템

- 각 조직에서 사용하던 데이터를 통합하고 공유할 때 생기는 장점을 이용하는 시스템
- 구성 : 데이터베이스 관리 시스템(DBMS), 데이터베이스, 데이터 모델

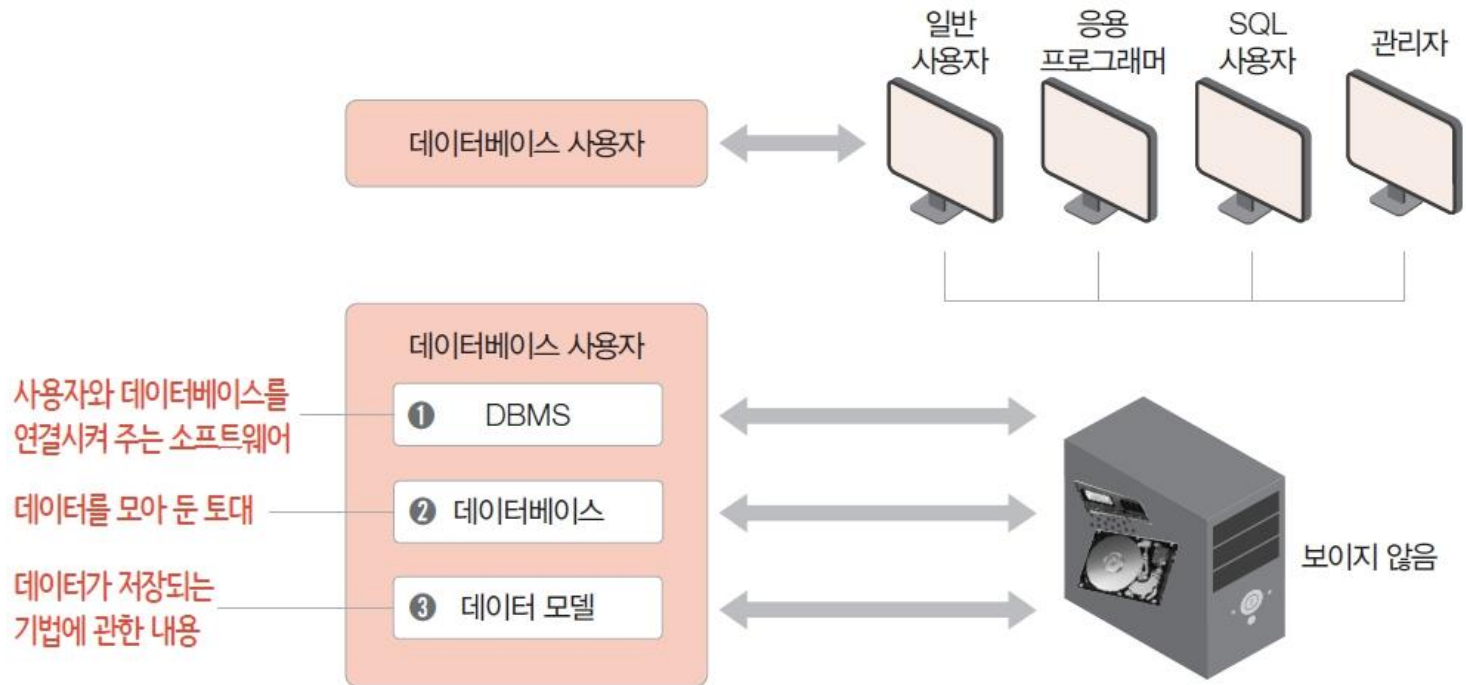


그림 1-5 데이터베이스 시스템의 구성 요소와 물리적인 위치

02 데이터베이스 시스템의 발전

1. 정보기술과 데이터베이스 시스템의 발전
2. 정보 시스템의 발전



1. 정보기술과 데이터베이스 시스템의 발전

❖ [1단계] 마당서점의 시작(동네 서점)



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

1. 정보기술과 데이터베이스 시스템의 발전

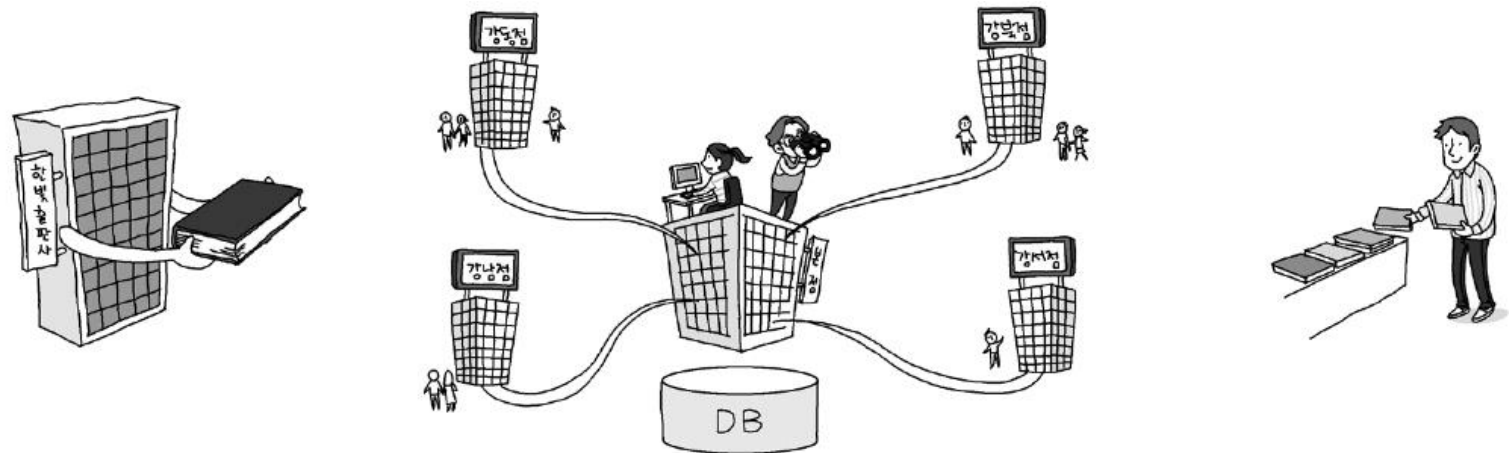
❖ [2단계] 초기 전산화(+컴퓨터)



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

1. 정보기술과 데이터베이스 시스템의 발전

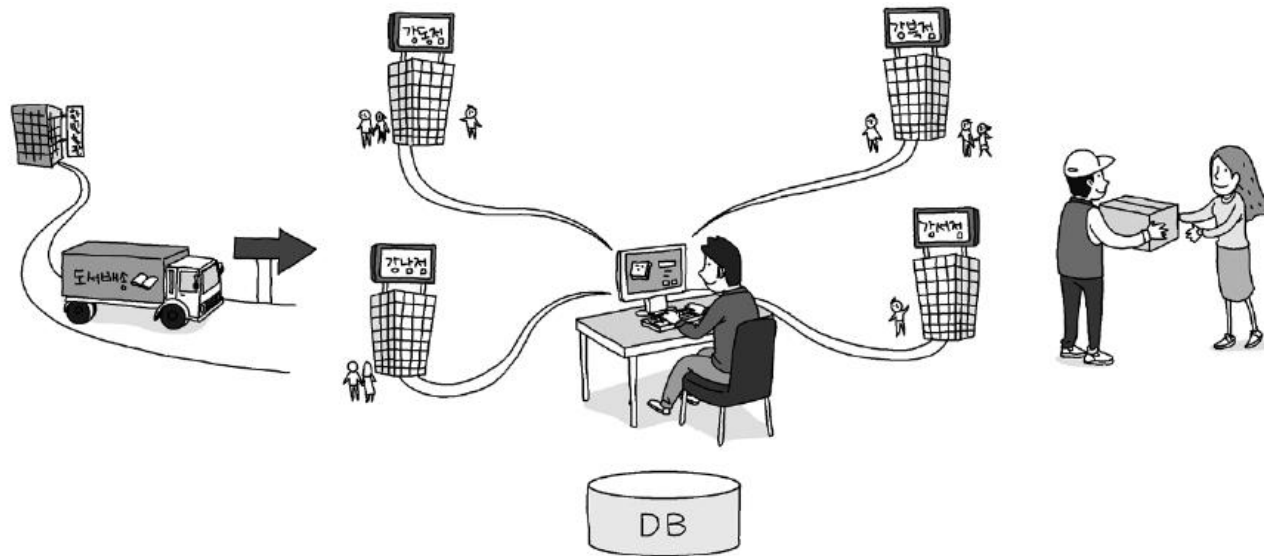
❖ [3단계] 데이터베이스 시스템 도입(+원격통신)



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

1. 정보기술과 데이터베이스 시스템의 발전

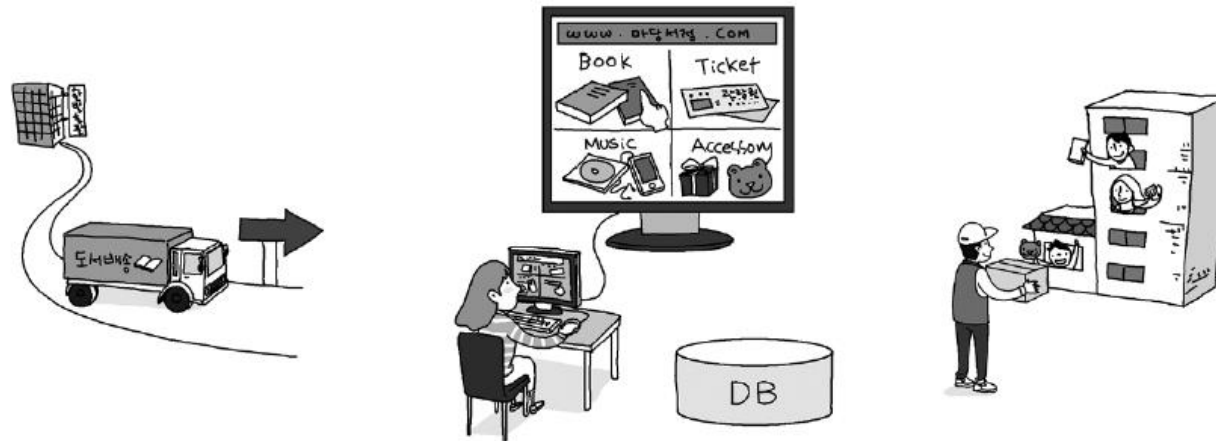
❖ [4단계] 홈페이지 구축(+인터넷)



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

1. 정보기술과 데이터베이스 시스템의 발전

❖ [5단계] 인터넷 쇼핑몰로 확장



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

1. 정보기술과 데이터베이스 시스템의 발전

- 마당서점의 성장 과정을 정보통신기술의 발전 양상과 정리한 표

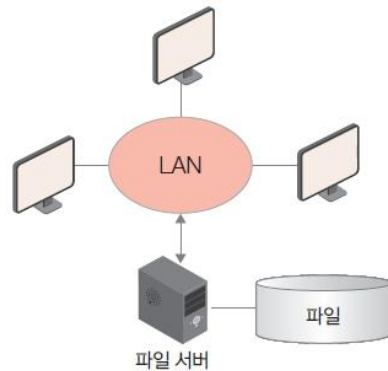
표 1-2 정보통신기술의 발전과 마당서점의 성장

단계	시기	주요 특징
	정보통신기술	
[1단계] 마당서점	1970년대	<ul style="list-style-type: none"> • 사장이 모든 도서의 제목과 가격을 기억함 • 매출과 판매가 컴퓨터 없이 관리됨 • 매출에 대한 내용이 정확하지 않음
	없음	
[2단계] 초기 전산화	1980년대	<ul style="list-style-type: none"> • 컴퓨터를 이용한 초기 응용 프로그램으로 업무를 처리함 • 파일 시스템을 사용함 • 한 대의 컴퓨터에서만 판매 및 매출을 관리함
	컴퓨터	
[3단계] 데이터베이스 구축	1990년대	<ul style="list-style-type: none"> • 지점 간 클라이언트/서버 시스템을 도입하여 업무를 처리함 • 데이터베이스 관리 시스템(DBMS)을 도입함
	컴퓨터+원격통신	
[4단계] 홈페이지 구축	2000년대	<ul style="list-style-type: none"> • 인터넷을 이용하여 도서 검색 및 주문함 • 웹 DB 시스템으로 불특정 다수 고객을 유치함 • 고객이 지리적으로 넓게 분산됨
	컴퓨터+인터넷	
[5단계] 인터넷 쇼핑몰	2010년대 이후	<ul style="list-style-type: none"> • 도서뿐 아니라 음반, 액세서리, 문구, 공연 티켓까지 판매하는 인터넷 쇼핑 물로 확대함 • 도서 외 상품의 매출 비중이 50% 이상으로 늘어남
	컴퓨터+인터넷 +스마트폰	

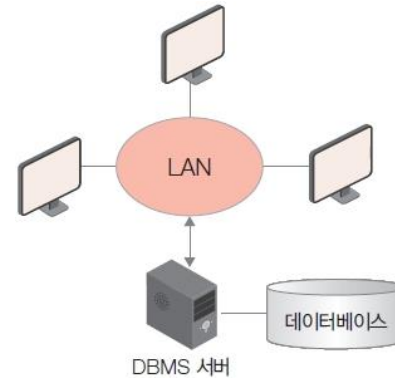
2. 정보 시스템의 발전

❖ 데이터 처리 관점에서 살펴본 기업 정보 시스템의 발전 과정

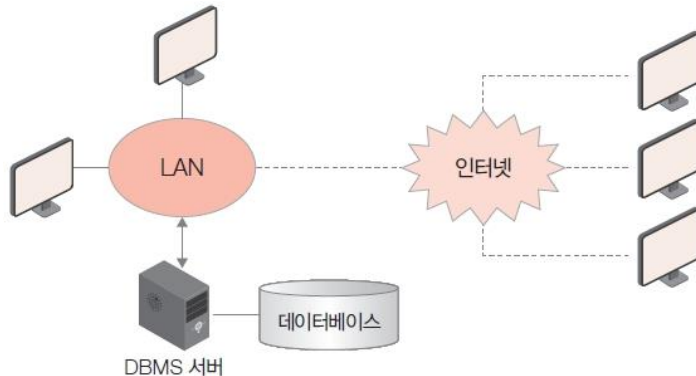
① 파일 시스템



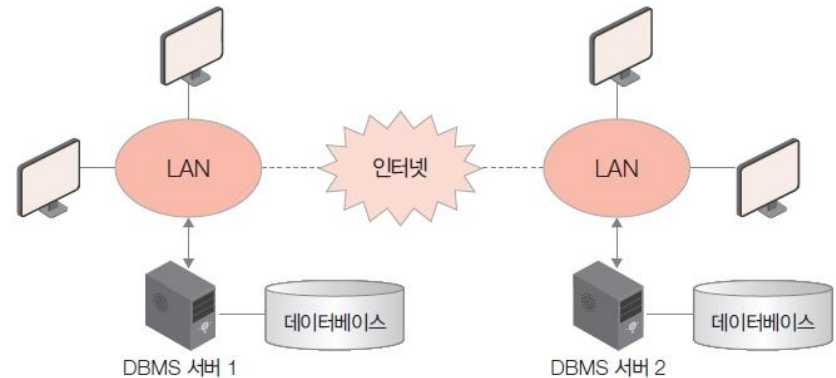
② 데이터베이스 시스템



③ 웹 데이터베이스 시스템



④ 분산 데이터베이스 시스템



2. 정보 시스템의 발전

- 정보 시스템의 발전 과정을 기업의 업무 환경과 연계하여 정리한 표

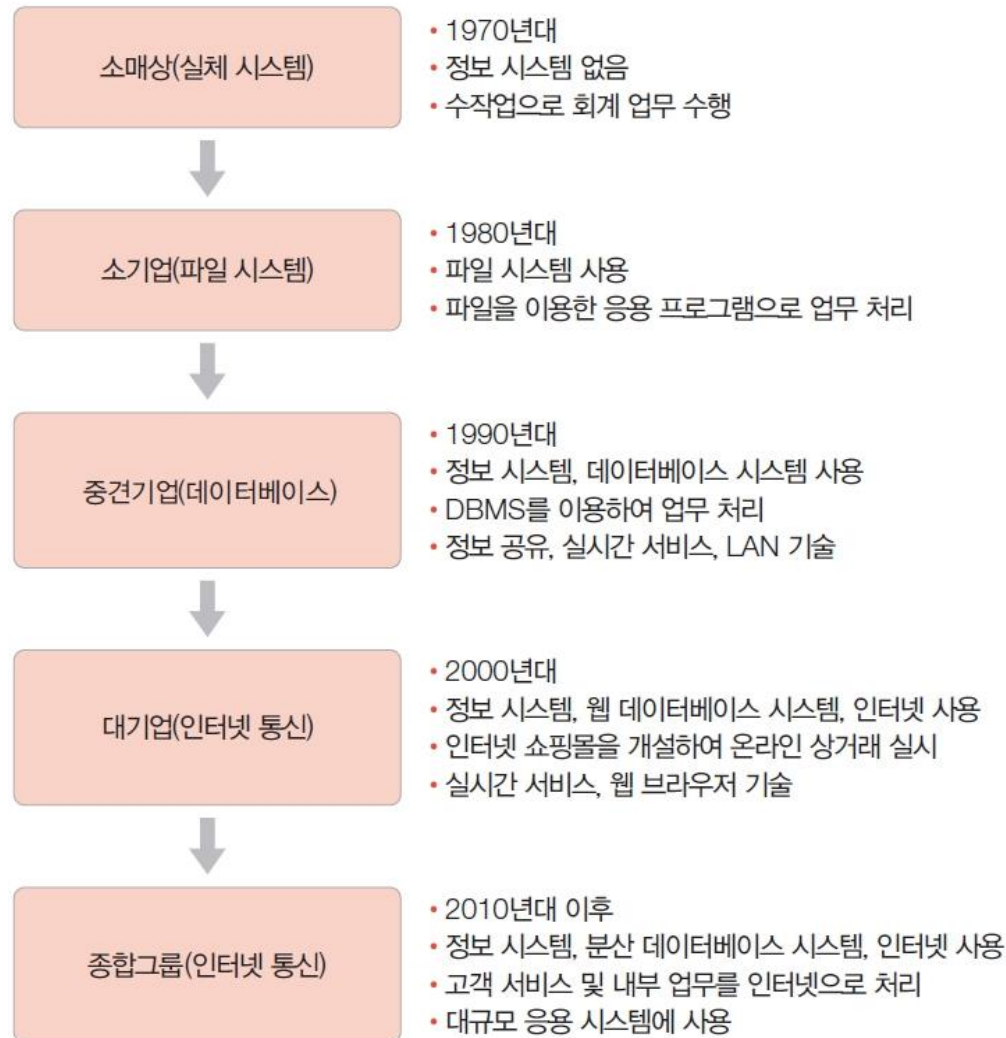


그림 1-6 정보 시스템의 발전과 기업의 업무 환경 변화

03 파일 시스템과 DBMS

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



1. 마당서점 데이터를 저장하는 방법

❖ 도서 검색 프로그램

- 마당서점 초기 : 취급하는 도서 수가 적어 컴퓨터에 따로 데이터를 저장하여 관리할 필요가 없었음
- 이후 : 도서 수가 대폭 늘어 고객이 직접 조회할 수 있도록 검색 프로그램을 제공

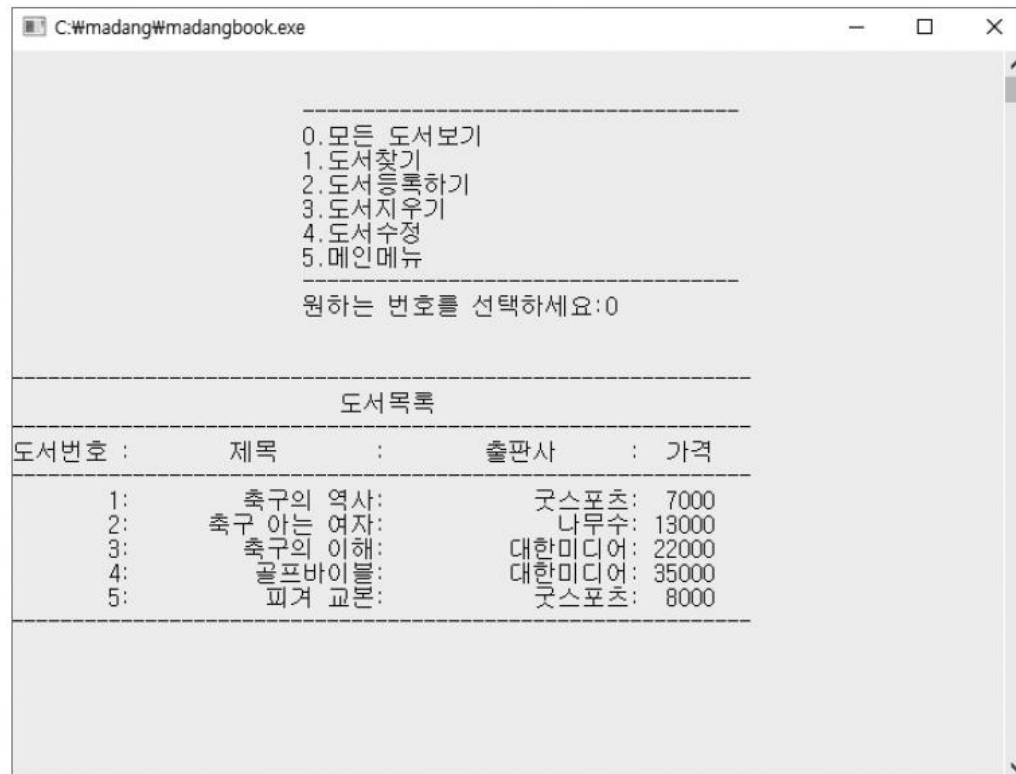


그림 1-7 도서 검색 프로그램

1. 마당서점 데이터를 저장하는 방법

❖ 데이터를 프로그램 내부에 저장하는 방법

■ 프로그램 1 : 데이터를 프로그램 내부에 저장하기

- C 언어의 구조체 BOOK을 먼저 선언하고 main() 프로그램에서 구조체 배열 변수 BOOKS[]에 데이터를 저장
- 도서 데이터는 프로그램 내 구조체 변수에 저장됨

```
/* BOOK 데이터 구조 정의 */
typedef struct {
    int    bookid;
    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, "촉구의 역사");
    strcpy(BOOKS[1].publisher, "굿스포츠");
    BOOKS[1].price = 7000;
```

1. 마당서점 데이터를 저장하는 방법

❖ 데이터를 프로그램 내부에 저장하는 방법

```
/* 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();  
  
/* 기타 프로그램 코드 */  
...(생략)...  
}
```

■ 프로그램 1의 문제점

- 새로운 데이터가 생길 때마다 프로그램을 수정한 후 다시 컴파일하고, 새로 컴파일된 프로그램을 실행시켜야 함
- 프로그램을 컴파일하여 새로운 프로그램을 가동하는 순간 검색 서비스를 중단시켜야 함

1. 마당서점 데이터를 저장하는 방법

❖ 파일 시스템을 사용하는 방법

- 파일 시스템 : 데이터를 프로그램과 분리하여 별도의 파일에 저장하는 방법
- C 언어로 작성한 도서 검색 프로그램에서 도서를 등록하는 화면
 - 새로운 도서가 추가되면 프로그램과 분리되어 별도의 파일에 저장

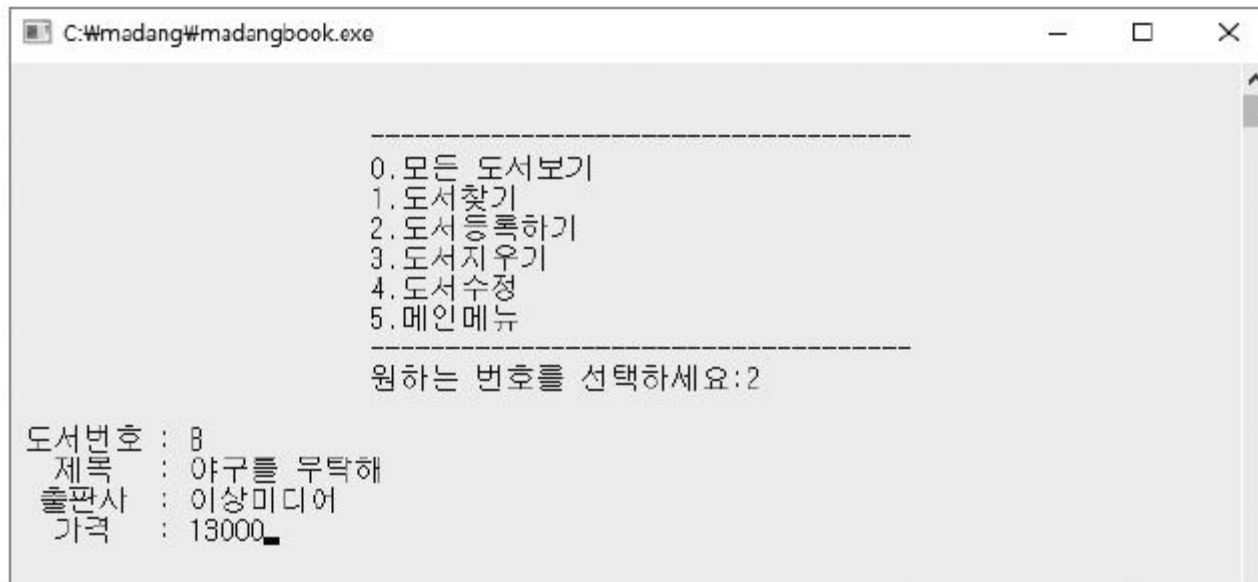


그림 1-8 도서 검색 프로그램에서 도서를 등록하는 화면

1. 마당서점 데이터를 저장하는 방법

❖ 파일 시스템을 사용하는 방법

■ 프로그램 2 : 데이터를 파일에 저장하기

- BOOK 데이터 구조를 먼저 선언하고 main() 프로그램에서 파일로부터 데이터를 불러와 구조체 배열 변수 BOOKS[]에 저장
- 새로운 도서가 입고되면 '도서등록하기' 화면에서 도서 정보를 입력 받아 파일에 저장

```
/* BOOK 데이터 구조 정의 */
```

```
typedef struct {  
    int    bookid;  
    char   bookname[20];  
    char   publisher[20];  
    int    price;  
} BOOK;
```

```
int main() {  
    BOOK BOOKS[10];  
    int i = 1;  
    insert();    /* 도서 입력 함수 */
```

1. 마당서점 데이터를 저장하는 방법

❖ 파일 시스템을 사용하는 방법

```
/* 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();

/* 기타 프로그램 코드 */
...(생략)...
}
```

■ 프로그램 2의 문제점

- 데이터 구조가 바뀌는 문제(불편함)가 있음
- 같은 파일을 두 개의 프로그램이 공유함으로써 발생하는 문제가 있음

1. 마당서점 데이터를 저장하는 방법

❖ DBMS를 사용하는 방법

■ 프로그램 3 : 데이터를 DBMS에 저장하기

```
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;
}
```

프로그램에서 데이터를 불러올 때는 EXEC SQL이라는 별도의 명령어를 이용하여 DBMS에 호출함

DBMS는 데이터를 정의하고 데이터를 관리하는 사용자 인터페이스를 따로 제공함

1. 마당서점 데이터를 저장하는 방법

❖ DBMS를 사용하는 방법

- MySQL의 데이터베이스 관리 소프트웨어 [MySQL Workbench](#)

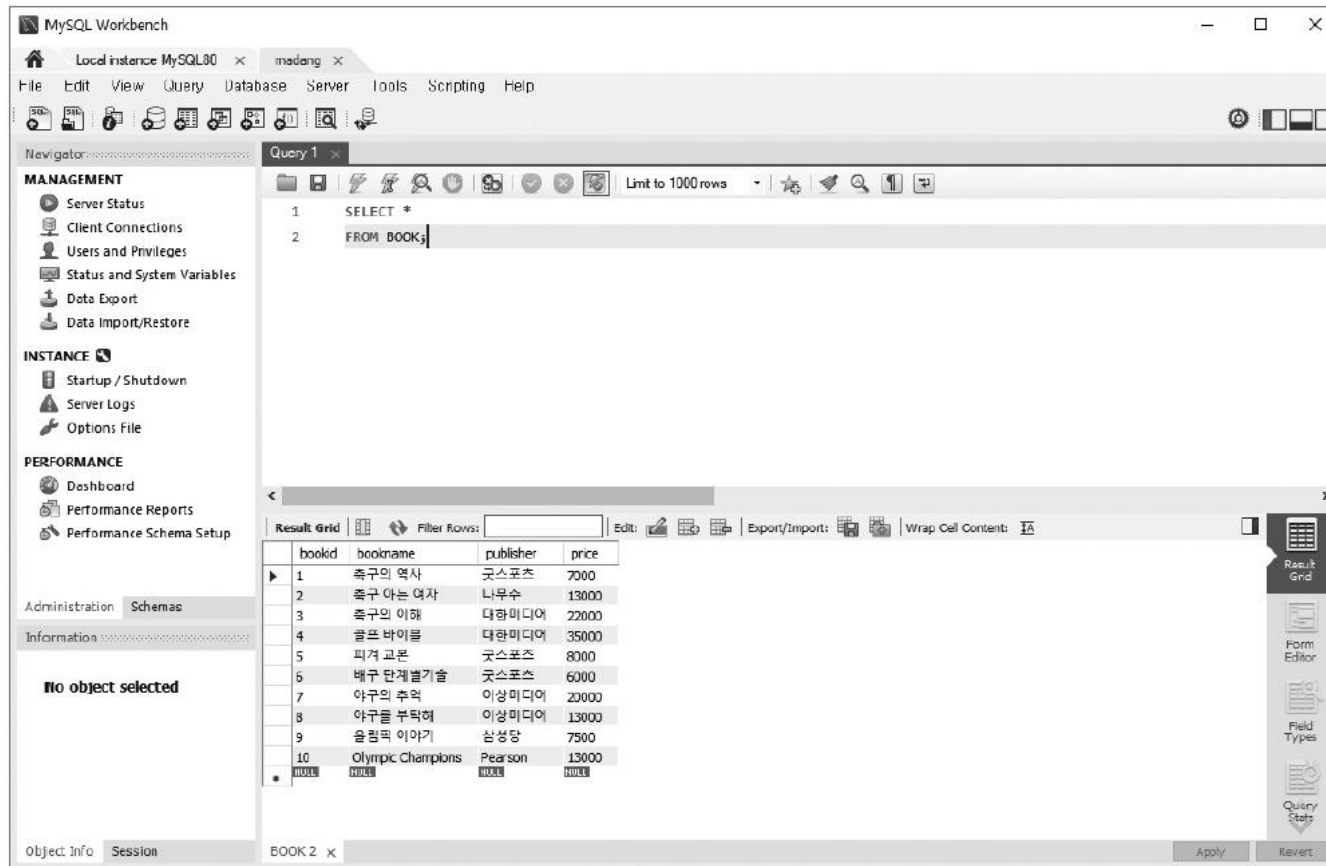


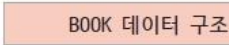
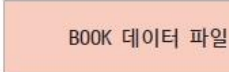
그림 1-9 MySQL Workbench의 데이터베이스 관리 화면

2. 마당서점 데이터의 저장 방법 비교

❖ [프로그램 1] 구조

표 1-3 마당서점 데이터의 저장 방법 비교

프로그램 1

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

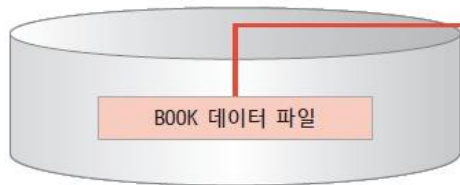
- 프로그램에 데이터 정의와 데이터 값을 모두 포함하는 방식
- 프로그램에 BOOK 데이터 구조를 정의하고 데이터 값도 직접 변수에 저장함
- 데이터 구조 혹은 데이터 값이 바뀌면 프로그램을 다시 컴파일해야 함

2. 마당서점 데이터의 저장 방법 비교

❖ [프로그램 2] 구조

프로그램 2

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



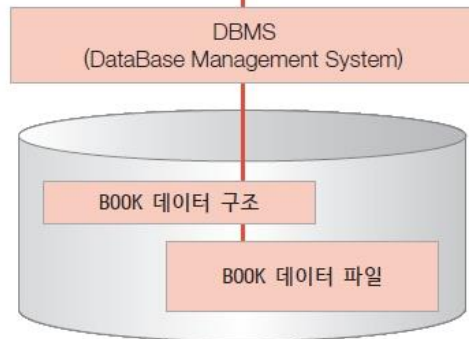
- 파일에 데이터 값, 프로그램에 데이터 정의를 포함하는 방식
- 프로그램에 BOOK 데이터 구조만 정의하고, 데이터 값은 book.dat 라는 파일에 저장함
- 데이터 값이 바뀌면 프로그램에 변경이 없지만, 데이터 구조가 바뀌면 프로그램을 다시 컴파일해야 함

2. 마당서점 데이터의 저장 방법 비교

❖ [프로그램 3] 구조

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 저장함
- 데이터 구조가 바뀌거나 데이터 값이 바뀌어도 프로그램을 다시 컴파일할 필요가 없음

3. 파일 시스템과 DBMS 비교

- ❖ 파일 시스템을 이용하는 방법(프로그램 2 방법)과 DBMS를 이용하는 방법(프로그램 3 방법)을 비교한 표

표 1-4 파일 시스템과 DBMS의 비교

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

3. 파일 시스템과 DBMS 비교

- 예) 마당서점의 구매 담당자(출판사로부터 도서 구매)와 판매 담당자(고객에게 도서 판매)가 업무에 필요한 프로그램을 각각 운영하는 경우

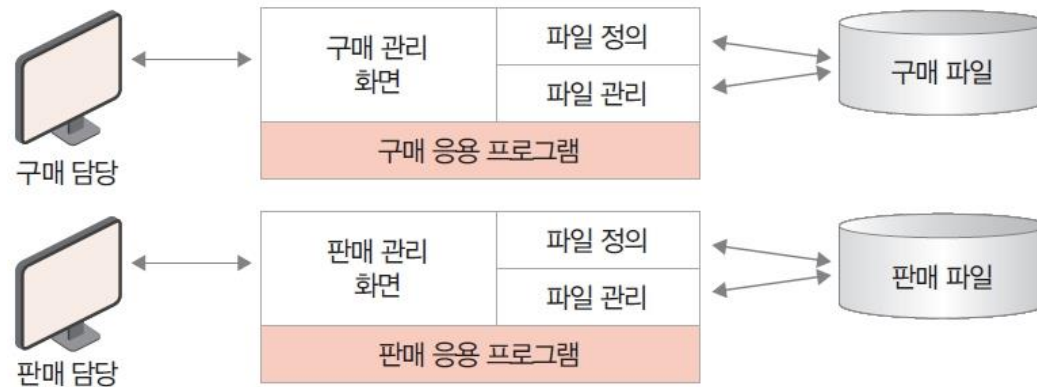


그림 1-10 파일 시스템으로 구축된 구매 및 판매 응용 프로그램

- [그림 1-11]은 DBMS로 구축된 구매 및 판매 응용 프로그램

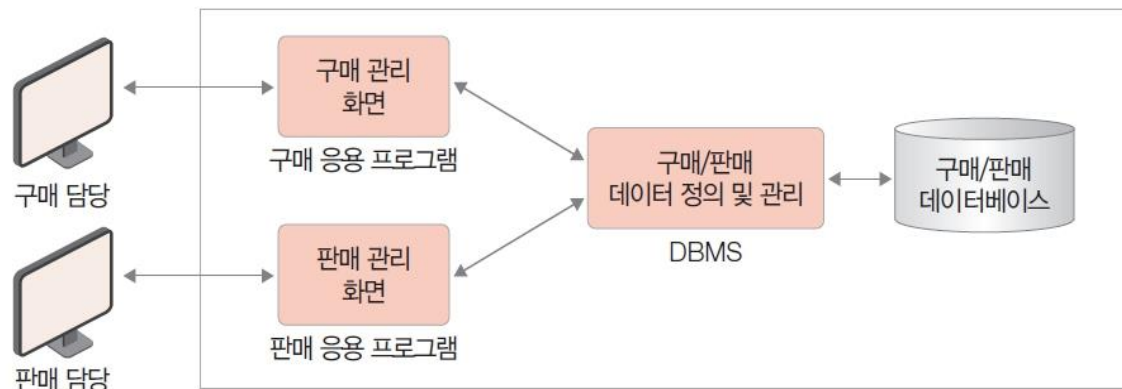


그림 1-11 DBMS로 구축된 구매 및 판매 응용 프로그램

3. 파일 시스템과 DBMS 비교

- 파일 시스템과 비교하여 DBMS의 장점 비교 표

표 1-5 DBMS의 장점

장점	설명
데이터 중복 최소화	DBMS를 이용하여 데이터를 공유하므로 중복 가능성이 낮음
데이터 일관성 유지	중복 제거로 데이터의 일관성이 유지됨
데이터 독립성 유지	데이터 정의와 프로그램 간의 독립성을 유지할 수 있음
관리 기능 제공	데이터 복구, 보안, 동시성 제어, 데이터 관리 기능 등을 수행함
프로그램 개발 생산성 향상	짧은 시간에 큰 프로그램을 개발할 수 있음
기타	데이터 무결성 유지, 데이터 표준 준수가 용이함

04 데이터베이스 시스템의 구성

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



데이터베이스 시스템의 구성

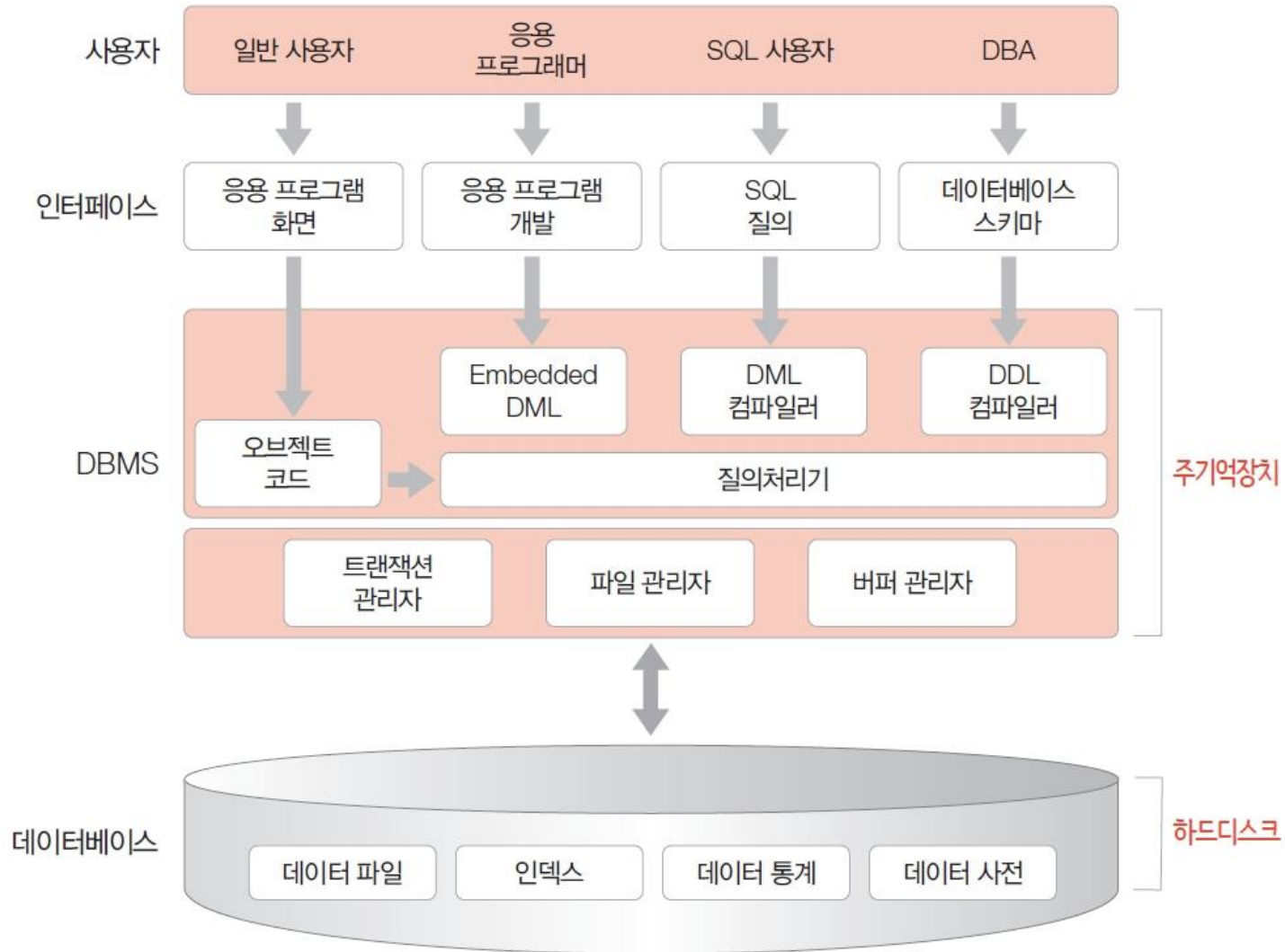


그림 1-12 데이터베이스 시스템의 구성

1. 데이터베이스 언어

❖ SQL(Structured Query Language)의 구성

- 데이터 정의어
 - DBMS에 저장된 테이블 구조를 정의하는 데 사용
- 데이터 조작어
 - 데이터를 검색·삽입·삭제·수정하는 데 사용
- 데이터 제어어
 - 데이터의 사용 권한을 관리하는 데 사용

❖ SQL의 핵심

- 데이터 조작어 중 데이터를 검색하는 질의문
- 질의문은 SELECT-FROM-WHERE 구조로 되어 있음

1. 데이터베이스 언어

- [그림 1-13]은 Book 테이블에서 질의문을 사용하는 예

bookid	bookname	publisher	price
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

(a) Book 테이블

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

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

(b) 질의문과 결과 예

Book 테이블에서 모든 bookname(도서명)과 publisher(출판사)를 검색한다.

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

bookname	publisher
축구 아는 여자	나무수
축구의 이해	대한미디어
골프 바이블	대한미디어

Book 테이블에서
price(정가)가 10,000원 이상인
bookname(도서명)과
publisher(출판사)를 검색한다.

그림 1-13 질의문 사용 예

2. 데이터베이스 사용자

- 일반 사용자
 - 은행의 창구 혹은 관공서의 민원 접수처 등에서 데이터 관련 업무를 하는 사람
 - 프로그래머가 개발한 프로그램을 이용하여 데이터베이스에 접근
 - 응용 프로그래머
 - 일반 사용자가 사용할 수 있도록 프로그램을 만드는 사람
 - SQL 사용자
 - SQL을 사용해 업무를 처리하는 IT 부서의 담당자
 - 응용 프로그램으로 구현되지 않은 업무를 SQL을 사용해 처리
 - 데이터베이스 관리자
 - 데이터베이스 운영 조직의 데이터베이스 시스템을 총괄하는 사람
- 데이터베이스 사용자별로 갖추어야 할 지식수준

표 1-6 데이터베이스 사용자별로 갖추어야 할 지식수준(×: 없음, ○: 보통, ◎: 높음)

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

3. DBMS

❖ DBMS 기능

표 1-7 DBMS의 기능

기능	설명
데이터 정의	<ul style="list-style-type: none">• 데이터 구조를 정의하고 데이터 구조에 대한 삭제 및 변경 기능을 수행함
데이터 조작	<ul style="list-style-type: none">• 데이터를 조작하는 소프트웨어(응용 프로그램)가 요청하는 데이터의 검색 · 삽입 · 수정 · 삭제 작업을 지원함
데이터 추출	<ul style="list-style-type: none">• 사용자가 조회하는 데이터 혹은 응용 프로그램의 데이터를 추출함
데이터 제어	<ul style="list-style-type: none">• 데이터베이스 사용자를 생성하고 모니터링하며 접근을 제어함• 백업과 회복, 동시성 제어 등의 기능을 지원함

4. 데이터 모델

❖ 데이터 모델의 개념

- 데이터베이스 시스템에서 데이터를 저장하는 이론적인 방법에 관한 것
- 데이터베이스에 데이터가 어떻게 구조화되어 저장되는지를 결정함
- 현재 가장 많이 사용되는 데이터 모델은 관계 데이터 모델
- 데이터 모델을 구분하는 가장 큰 기준은 데이터 간 관계를 표현하는 방법임

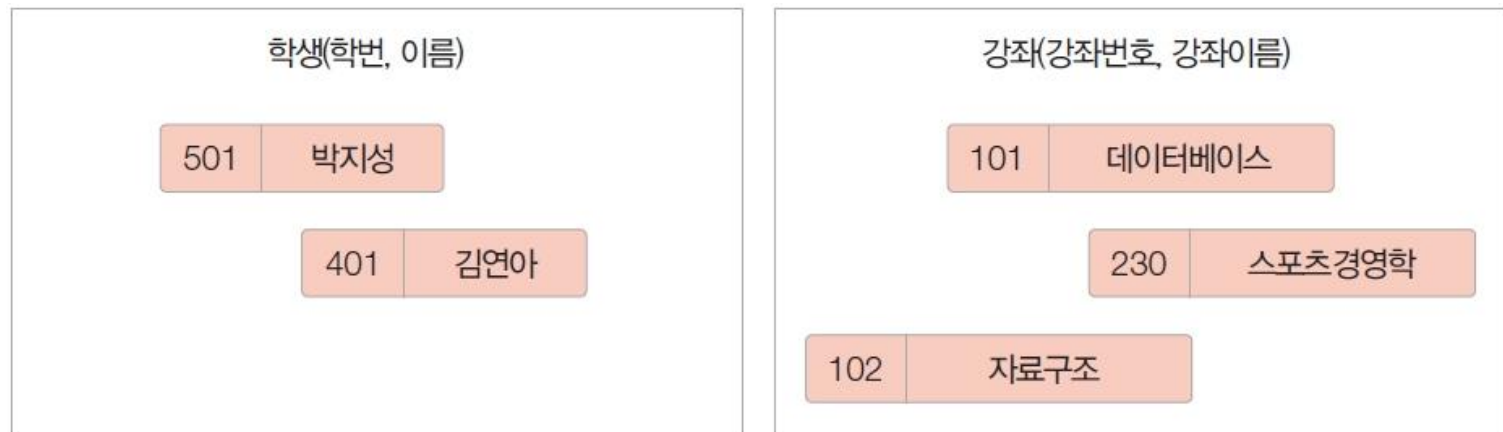
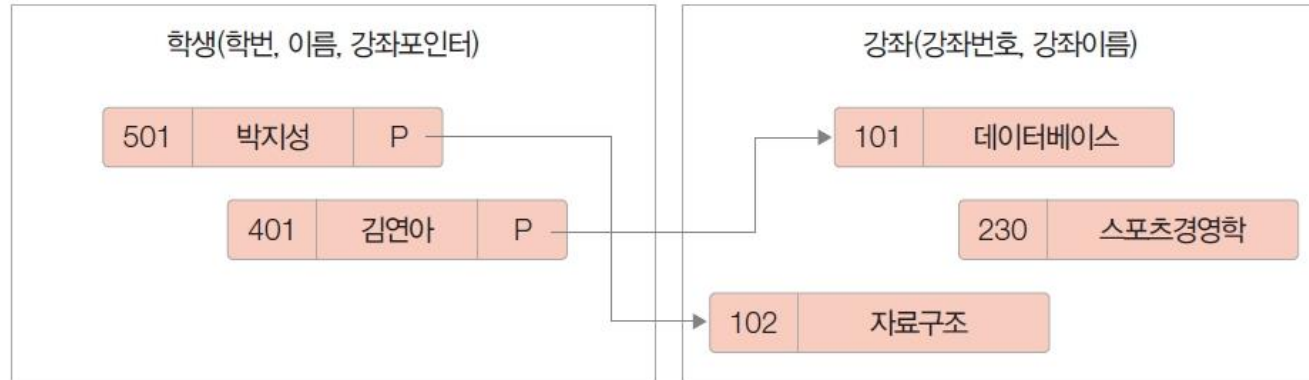


그림 1-14 관계 표현을 위한 테이블 예

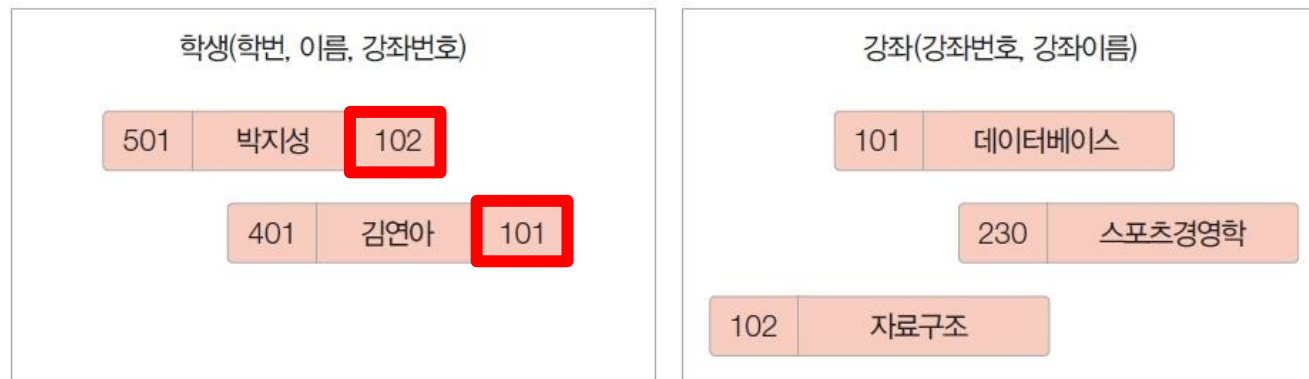
4. 데이터 모델

❶ 포인터 사용 : 계층 데이터 모델, 네트워크 데이터 모델



(a) 포인터 사용

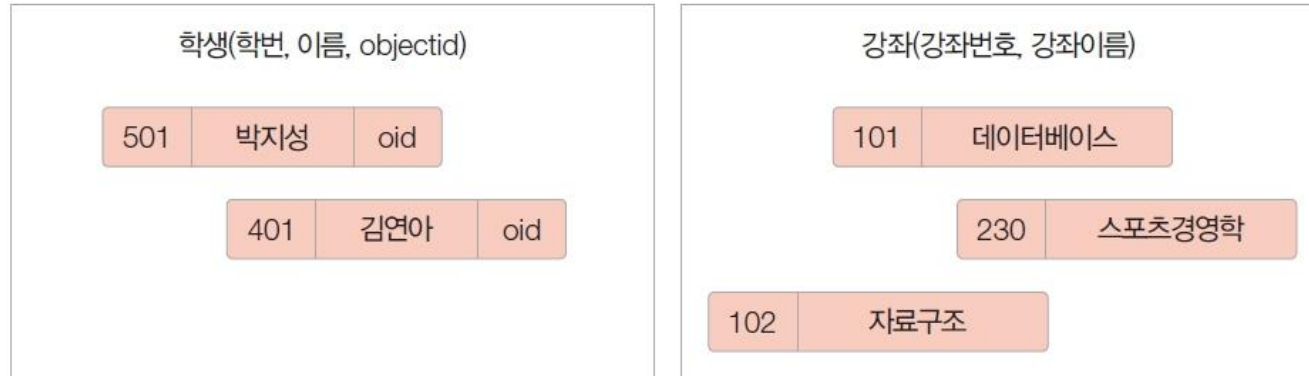
❷ 속성값 사용 : 관계 데이터 모델



(b) 속성값 사용

4. 데이터 모델

③ 객체 식별자 사용 : 객체 데이터 모델



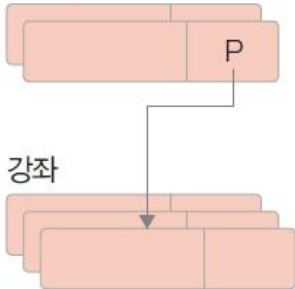
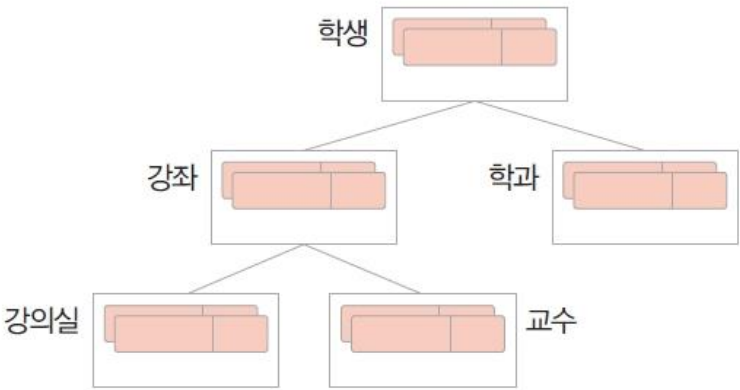
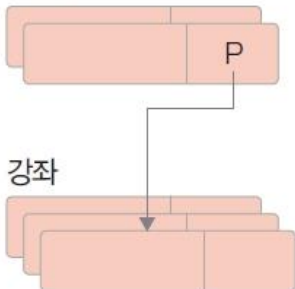
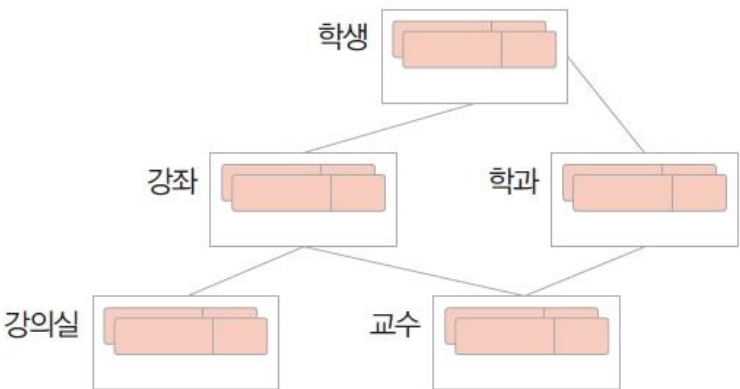
(c) 객체 식별자

그림 1-15 학생-강좌 관계 표현 예

4. 데이터 모델

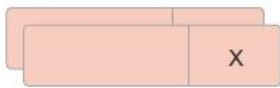
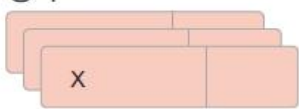
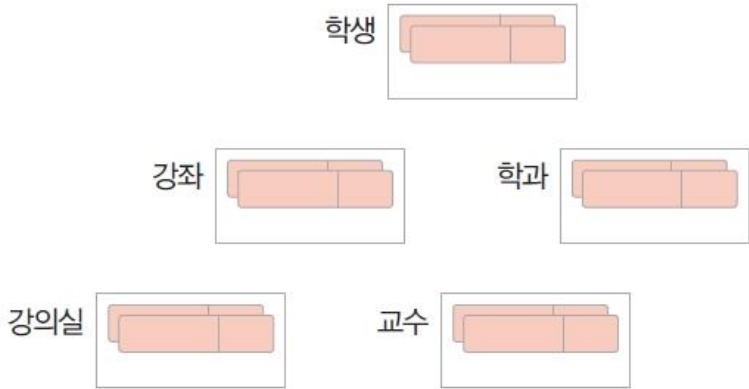
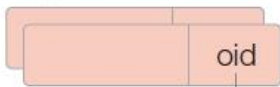
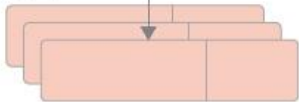
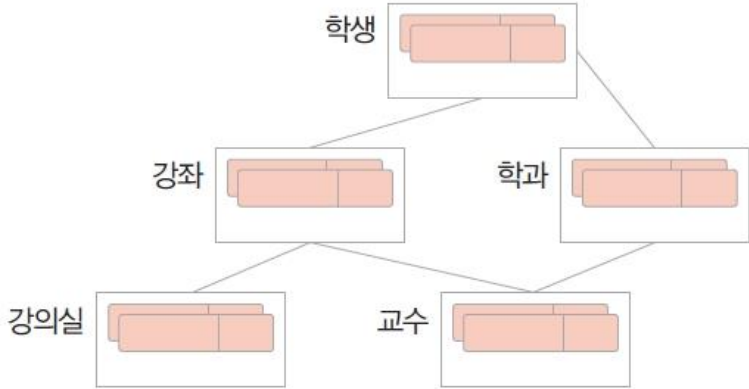
❖ 데이터 모델에서 관계 표현 방법

표 1-8 데이터 모델별 관계 표현 방법

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. 데이터 모델

❖ 데이터 모델에서 관계 표현 방법

데이터 모델	관계의 표현	데이터 구성
관계 데이터 모델 (속성값 사용)	<p>학생</p>  <p>강좌</p> 	
객체 데이터 모델 (객체 식별자 사용)	<p>학생</p>  <p>강좌</p>  <p>객체 식별자 oid</p>	

4. 데이터 모델

❖ 데이터 모델을 사용 시기별로 나타낸 것

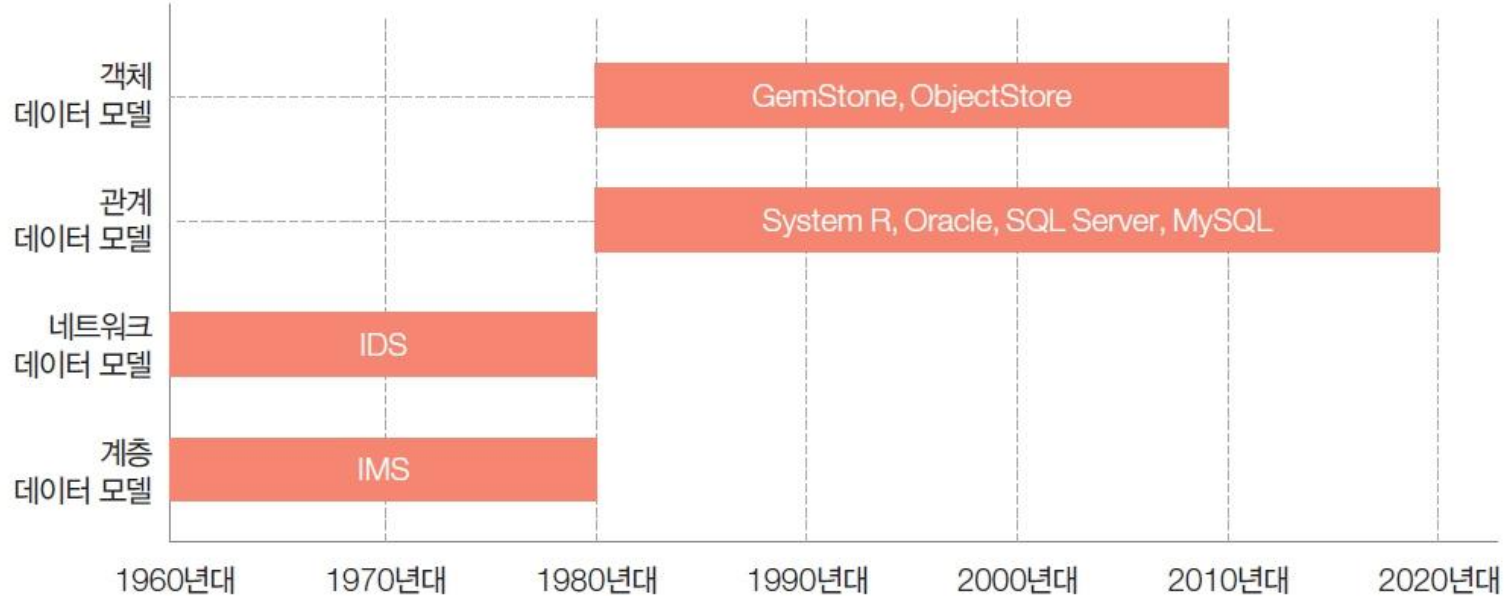


그림 1-16 데이터 모델과 제품의 역사

5. 데이터베이스의 개념적 구조

❖ 3단계 데이터베이스 구조

- 3단계 데이터베이스 구조는 외부 단계, 개념 단계, 내부 단계로 나뉨

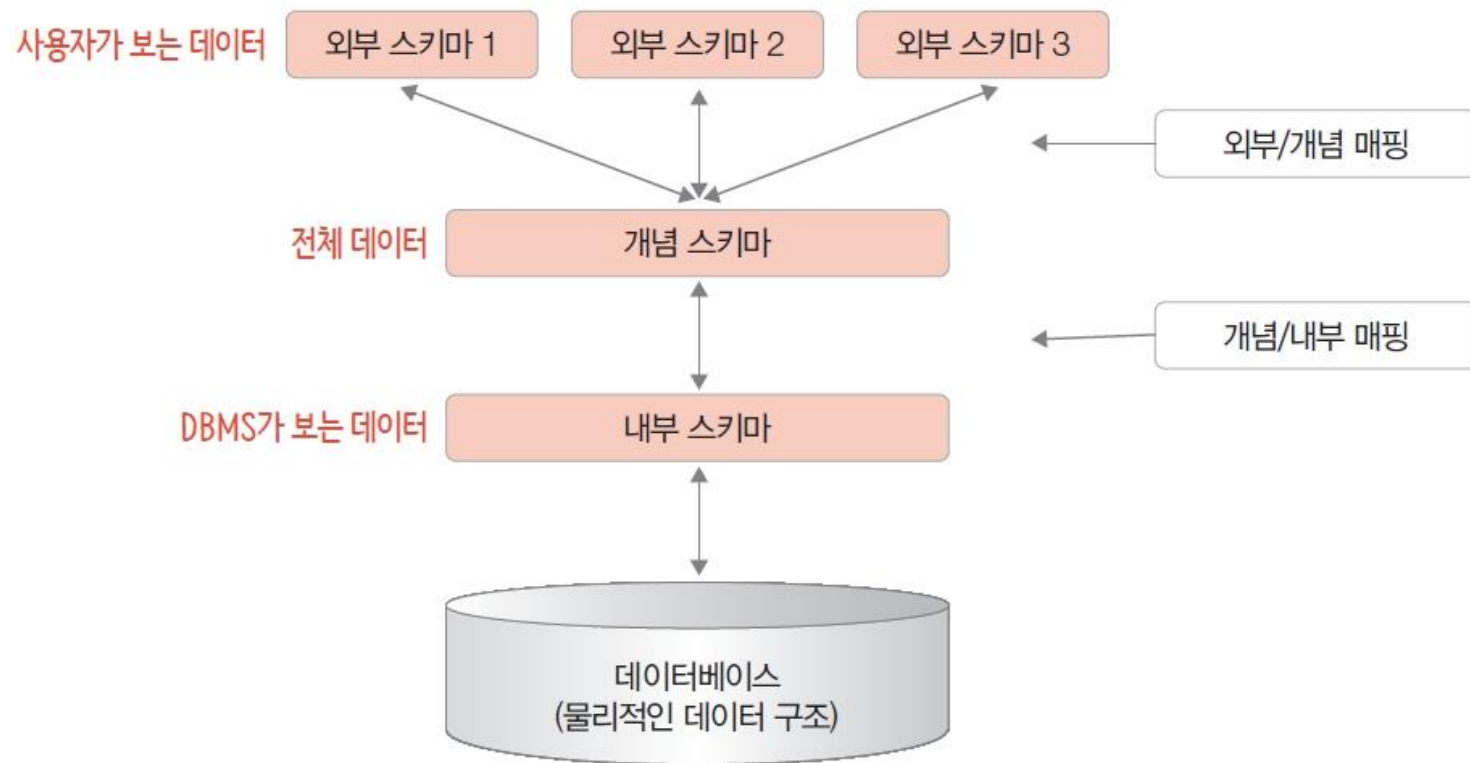


그림 1-17 ANSI의 3단계 데이터베이스 구조

'스키마(schema)'는 그리스어에서 유래된 단어로 조직이나 구조를 의미

5. 데이터베이스의 개념적 구조

❖ 외부 단계

- 일반 사용자나 응용 프로그래머가 접근하는 계층으로 전체 데이터베이스 중에서 하나의 논리적인 부분을 의미
- 여러 개의 외부 스키마가 있을 수 있음

❖ 개념 단계

- 전체 데이터베이스의 정의를 의미
- 통합 조직별로 하나만 존재하며 DBA가 관리함

❖ 내부 단계

- 물리적 저장 장치에 데이터베이스가 실제로 저장되는 방법을 표현한 것
- 내부 스키마는 하나만 존재
- 인덱스, 데이터 레코드의 배치 방법, 데이터 압축 등에 관한 사항이 포함됨

5. 데이터베이스의 개념적 구조

❖ DBMS는 매핑 사상을 통하여 각 단계 간 대응 관계를 정의함

- 외부/개념 매핑, 개념/내부 매핑
- 예) 대학의 수강신청을 예로 들어 3단계 데이터베이스 구조를 살펴보기

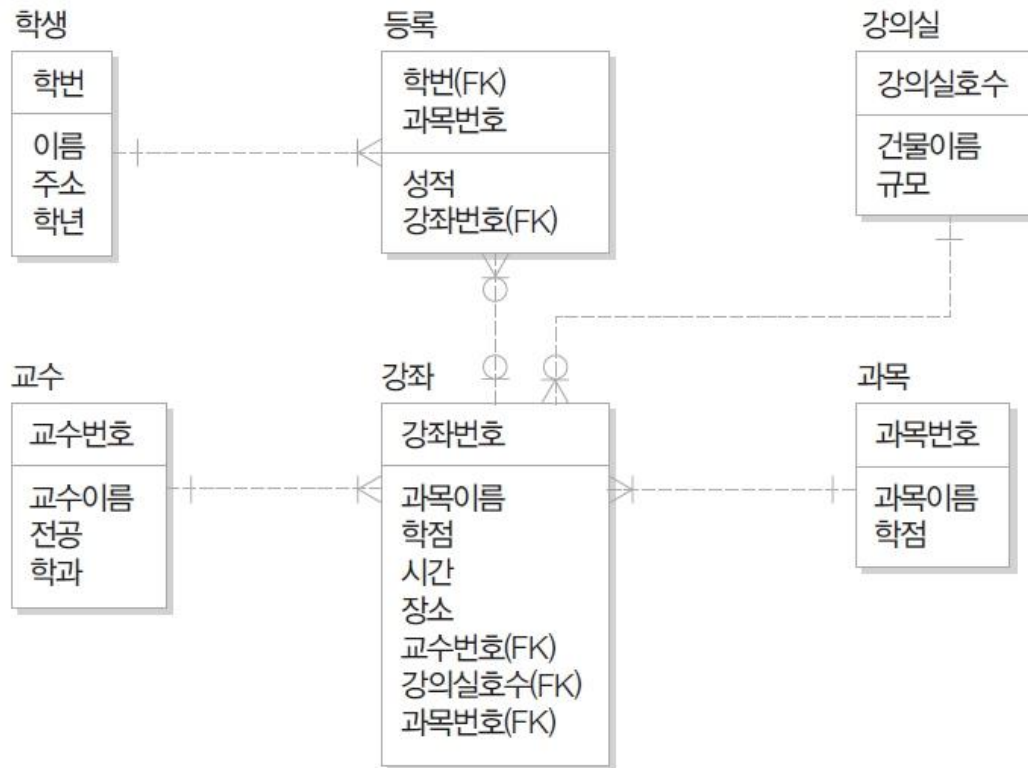


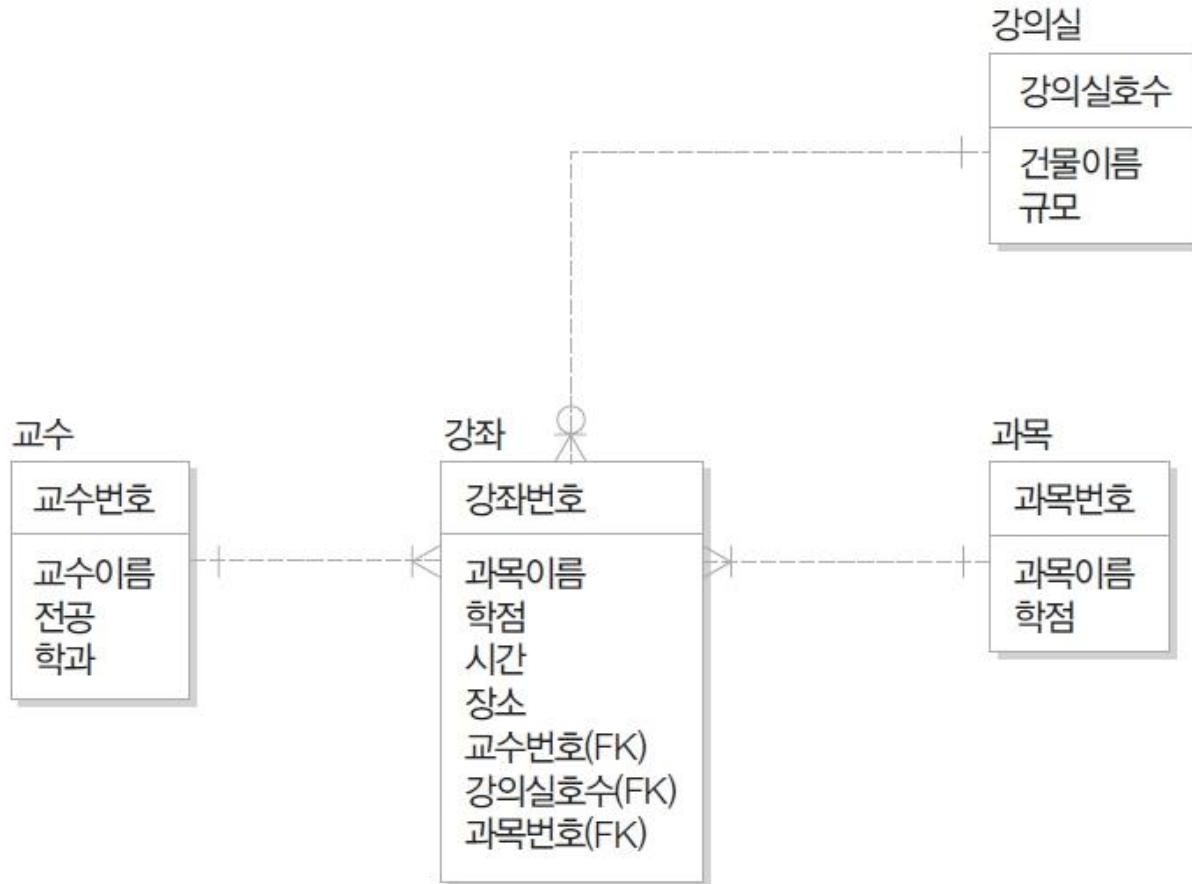
그림 1-18 수강신청 데이터베이스의 개념 스키마

5. 데이터베이스의 개념적 구조



(a) 외부 스키마 1: 수강등록 업무를 하는 학사관리과에 필요한 데이터베이스

5. 데이터베이스의 개념적 구조



(b) 외부 스키마 2: 시간표 작성 업무를 하는 수업관리과에 필요한 데이터베이스

그림 1-19 수강신청 데이터베이스의 외부 스키마

5. 데이터베이스의 개념적 구조

- 다음은 [그림 1-18]의 수강신청 데이터베이스의 내부 스키마를 그림으로 나타낸 것

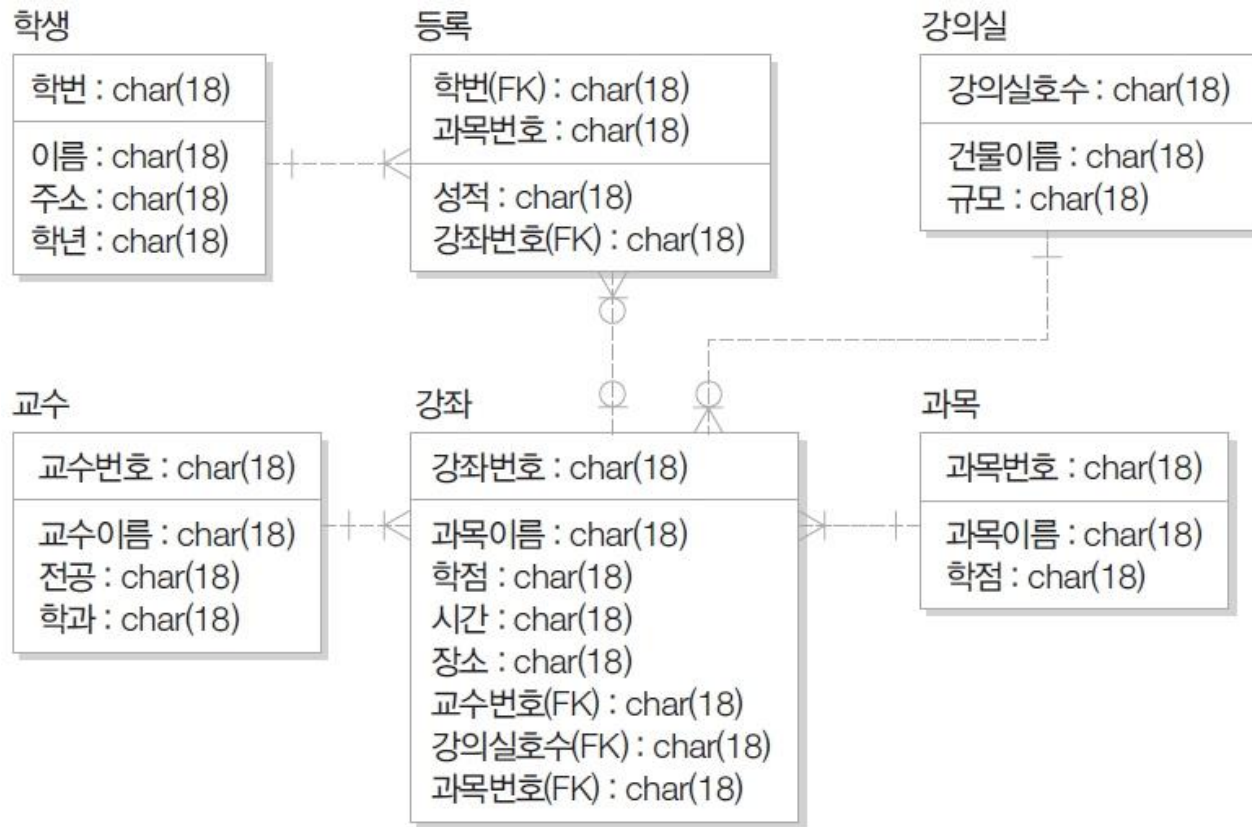
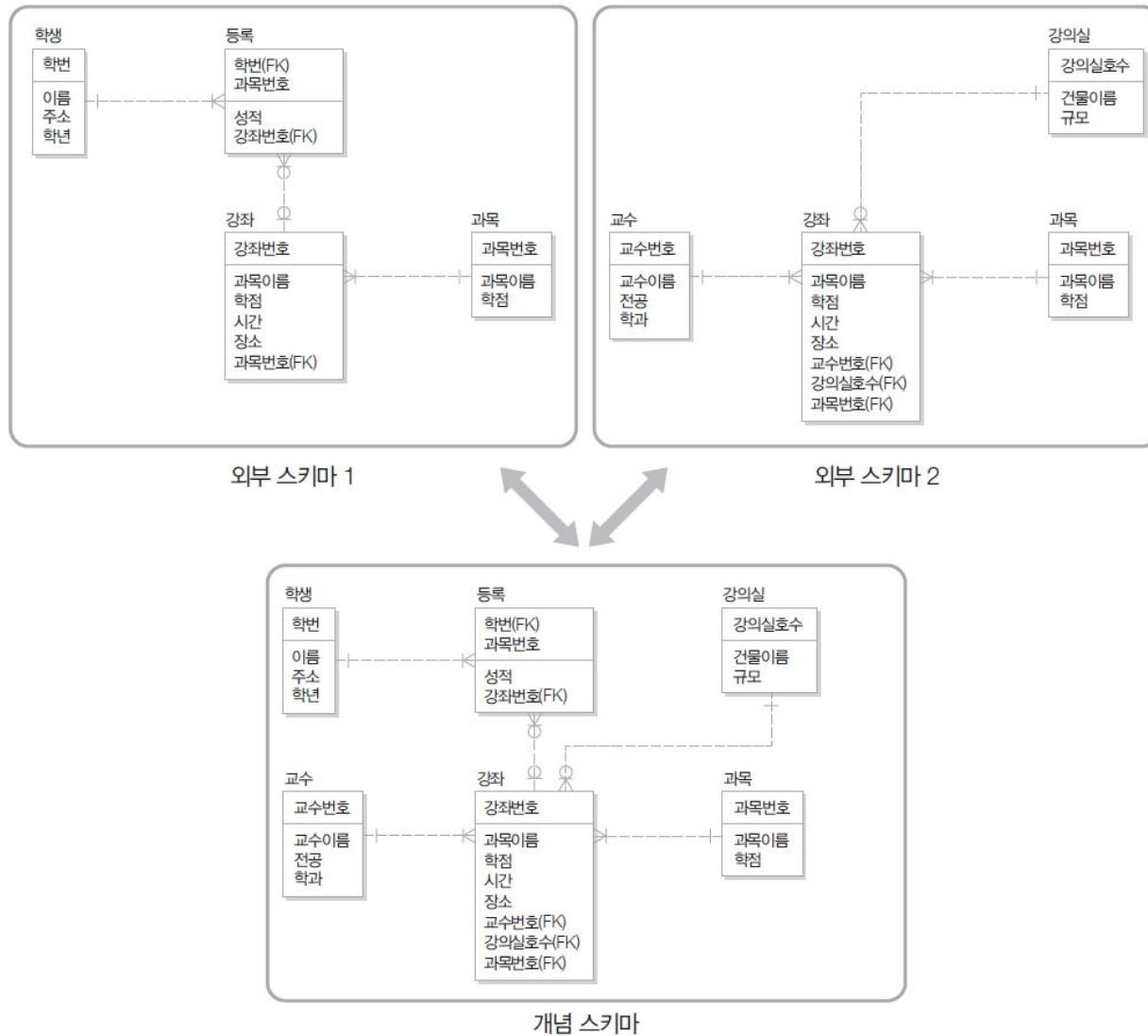


그림 1-20 수강신청 데이터베이스의 내부 스키마

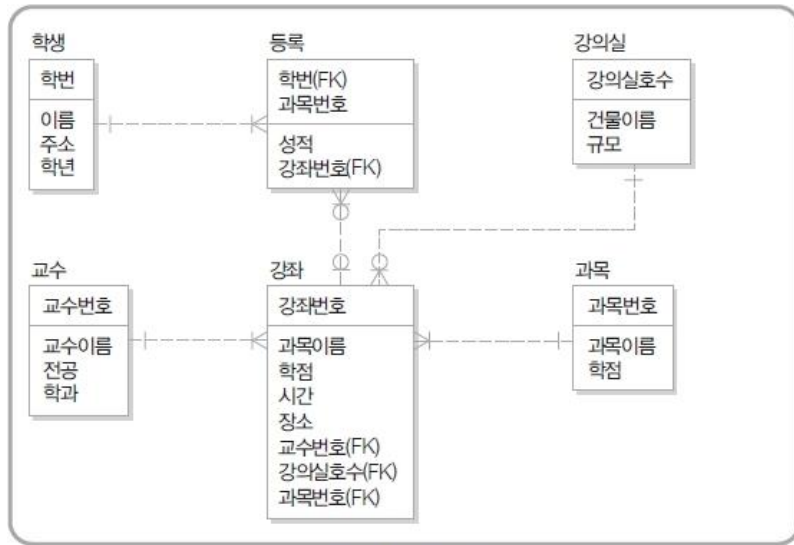
5. 데이터베이스의 개념적 구조

■ 수강신청 데이터베이스의 3단계 구조

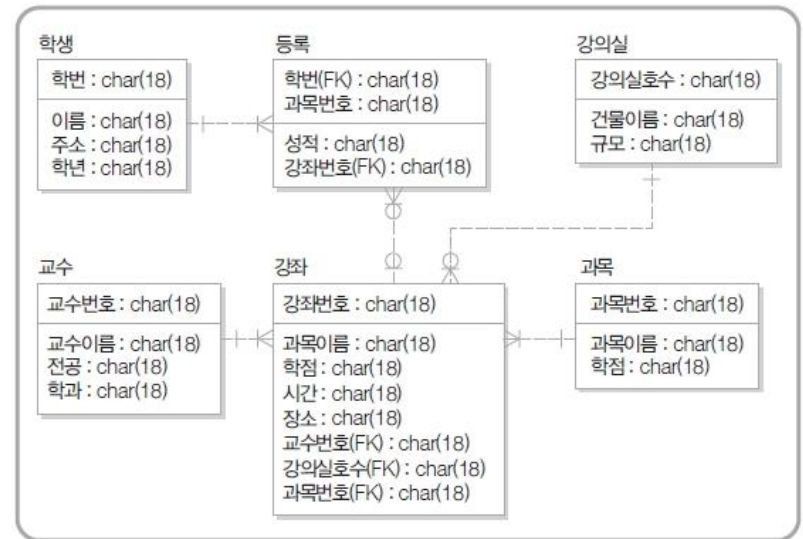


5. 데이터베이스의 개념적 구조

■ 수강신청 데이터베이스의 3단계 구조



개념 스키마



내부 스키마

5. 데이터베이스의 개념적 구조

❖ 데이터 독립성

- 하위 단계의 내용을 추상화하여 상위 단계에 그 세부 사항을 숨김으로써 한 단계 내의 변경에 대해서 다른 단계와 상호 간섭이 없도록 하는 것

❖ 2가지 데이터 독립성

- **논리적 데이터 독립성**(logical data independence)
 - 외부 단계와 개념 단계 사이의 독립성
 - 개념 스키마가 변경되어도 외부 스키마에는 영향을 미치지 않도록 지원함
 - 논리적 구조가 변경되어도 응용 프로그램에는 영향이 없도록 하는 개념
- **물리적 데이터 독립성**(physical data independence)
 - 개념 단계와 내부 단계 사이의 독립성
 - 저장 장치 구조 변경과 같이 내부 스키마가 변경되어도 개념 스키마에 영향을 미치지 않도록 지원함

요약

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