

제 1 장



데이터베이스 시스템

- 1.1 데이터베이스 시스템 개요
- 1.2 화일 시스템 vs. DBMS
- 1.3 DBMS의 발전 과정
- 1.4 DBMS 언어
- 1.5 DBMS 사용자
- 1.6 ANSI/SPARC 아키텍처와 데이터 독립성
- 1.7 데이터베이스 시스템 아키텍처
 - 연습문제

1장. 데이터베이스 시스템

- ❑ 컴퓨터를 사용하여 정보를 수집하고 분석하는데 데이터베이스 기술이 활용되고 있음
- ❑ 정보와 데이터는 서로 다름
- ❑ 데이터베이스(database)의 정의

데이터베이스는 조직체의 응용 시스템들이 공유해서 사용하는 운영 데이터 (operational data)들이 구조적으로 통합된 모임이다.
데이터베이스의 구조는 사용되는 데이터 모델에 의해 결정된다.

1장. 데이터베이스 시스템(계속)

□ 데이터베이스의 예

대학에서는 데이터베이스에 학생들에 관하여 신상 정보, 수강 과목, 성적 등을 기록하고, 각 학과에 개설되어 있는 과목들에 관한 정보를 유지하고, 교수에 관해서 신상 정보, 담당 과목, 급여 정보를 유지한다.

항공기 예약 시스템에서는 여행사를 통해 항공기 좌석을 예약하면 모든 예약 정보가 데이터베이스에 기록된다.

1장. 데이터베이스 시스템(계속)

❑ 데이터베이스의 특징

- ✓ 데이터베이스는 데이터의 대규모 저장소로서, 여러 부서에 속하는 여러 사용자에게 의해 동시에 사용됨
- ✓ 모든 데이터가 중복을 최소화하면서 통합됨
- ✓ 데이터베이스는 한 조직체의 운영 데이터뿐만 아니라 그 데이터에 관한 설명(데이터베이스 스키마 또는 메타데이터(metadata))까지 포함.
- ✓ 프로그램과 데이터 간의 독립성이 제공됨
- ✓ 효율적으로 접근이 가능하고 질의를 할 수 있음

❑ 데이터베이스 관리 시스템(DBMS: Database Management System)

- ✓ 데이터베이스를 정의하고, 질의어를 지원하고, 리포트를 생성하는 등의 작업을 수행하는 소프트웨어

1.1 데이터베이스 시스템 개요

□ 데이터베이스 스키마

- ✓ 전체적인 데이터베이스 구조를 뜻하며 자주 변경되지는 않음
- ✓ 데이터베이스의 모든 가능한 상태를 미리 정의
- ✓ **내포(intension)**라고 부름

□ 데이터베이스 상태

- ✓ 특정 시점의 데이터베이스의 내용을 의미하며, 시간이 지남에 따라 계속해서 바뀜
- ✓ **외연(extension)**이라고 부름

1.1 데이터베이스 시스템 개요(계속)

데이터베이스 스키마

DEPARTMENT (DEPTNO, DEPTNAME, FLOOR)

EMPLOYEE (EMPNO, EMPNAME, TITLE, DNO, SALARY)

데이터베이스 상태

DEPARTMENT

DEPTNO	DEPTNAME	FLOOR
1	영업	8
2	기획	10
3	개발	9

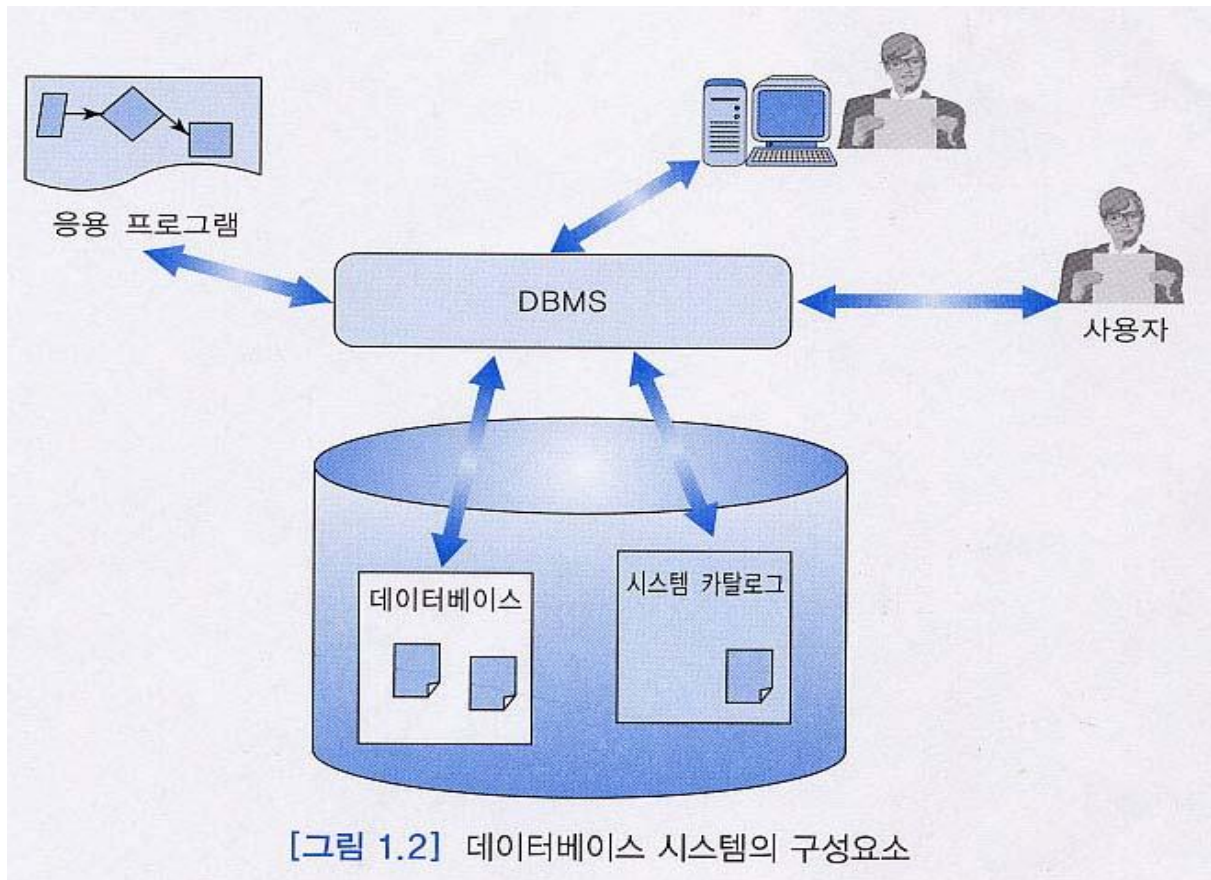
EMPLOYEE

EMPNO	EMPNAME	TITLE	DNO	SALARY
2106	김창섭	대리	2	2000000
3426	박영권	과장	3	2500000
3011	이수민	부장	1	3000000
1003	조민희	대리	1	2000000
3427	최종철	사원	3	1500000

[그림 1.1] 데이터베이스 스키마와 데이터베이스 상태

1.1 데이터베이스 시스템 개요(계속)

❑ 데이터베이스 시스템(DBS: Database System)의 구성 요소

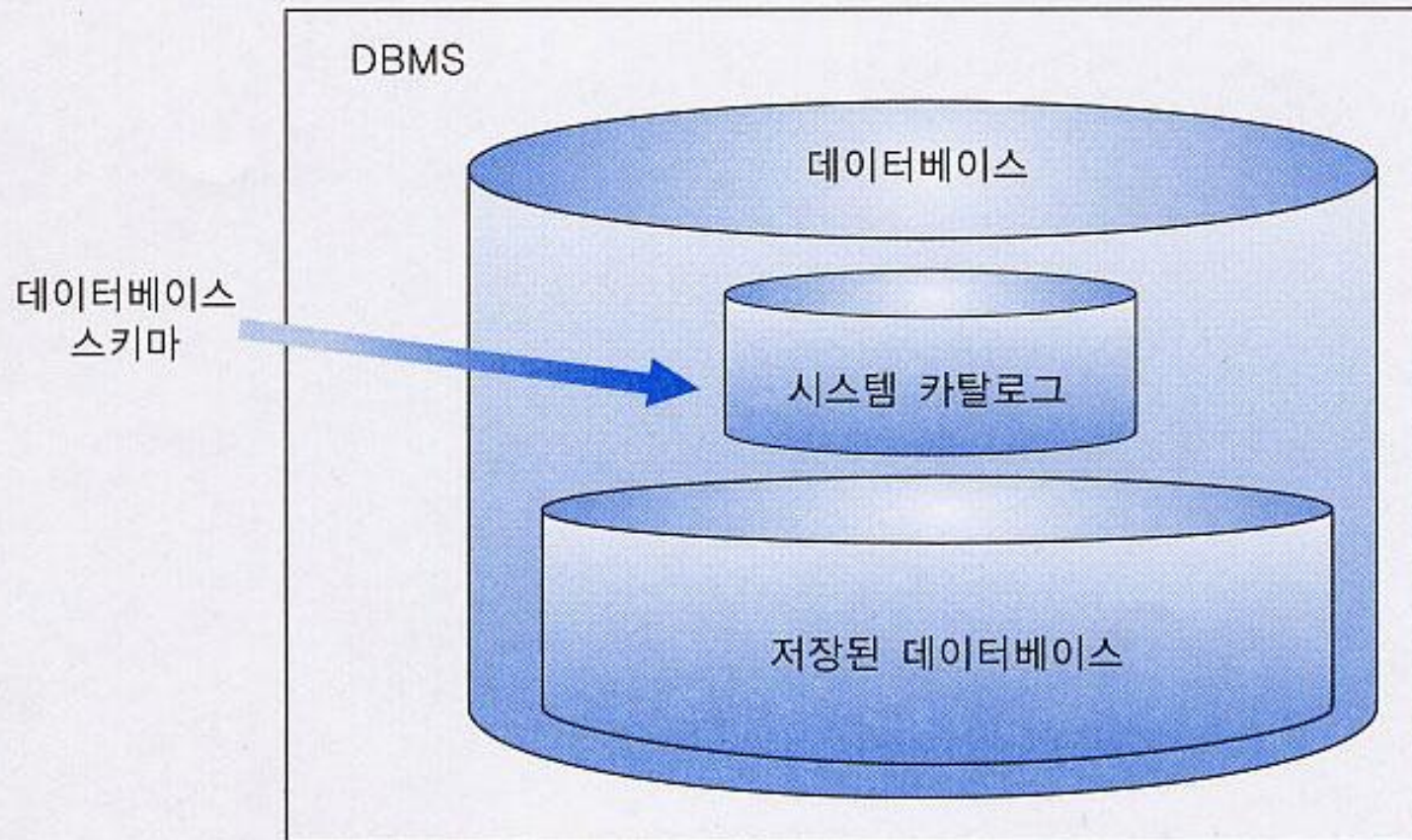


1.1 데이터베이스 시스템 개요(계속)

□ 데이터베이스

- ✓ 조직체의 응용 시스템들이 공유해서 사용하는 운영 데이터들이 구조적으로 통합된 모임
- ✓ 시스템 카탈로그(또는 데이터 사전)와 저장된 데이터베이스로 구분할 수 있음
- ✓ **시스템 카탈로그(system catalog)**는 저장된 데이터베이스의 스키마 정보를 유지

1.1 데이터베이스 시스템 개요(계속)



[그림 1.3] 시스템 카탈로그와 저장된 데이터베이스

1.1 데이터베이스 시스템 개요(계속)

□ DBMS

- ✓ 사용자가 새로운 데이터베이스를 생성하고, 데이터베이스의 구조를 명시할 수 있게 하고, 사용자가 데이터를 효율적으로 질의하고 수정할 수 있도록 하며, 시스템의 고장이나 권한이 없는 사용자로부터 데이터를 안전하게 보호하며, 동시에 여러 사용자가 데이터베이스를 접근하는 것을 제어하는 소프트웨어 패키지
- ✓ 데이터베이스 언어라고 부르는 특별한 프로그래밍 언어를 한 개 이상 제공
- ✓ SQL은 여러 DBMS에서 제공되는 사실상의 표준 데이터베이스 언어

1.1 데이터베이스 시스템 개요(계속)



[그림 1.4] 컴퓨터 시스템에서 DBMS의 위치

1.1 데이터베이스 시스템 개요(계속)

□ 사용자

- ✓ 데이터베이스 사용자는 여러 부류로 나눌 수 있음

□ 하드웨어

- ✓ 데이터베이스는 디스크와 같은 보조 기억 장치에 저장되며, DBMS에서 원하는 정보를 찾기 위해서는 디스크의 블록들을 주기억 장치로 읽어들여야 하며, 계산이나 비교 연산들을 수행하기 위해 중앙 처리 장치가 사용됨
- ✓ DBMS 자체도 주기억 장치에 적재되어 실행되어야 함

1.1 데이터베이스 시스템 개요(계속)

□ 데이터베이스 시스템의 요구사항

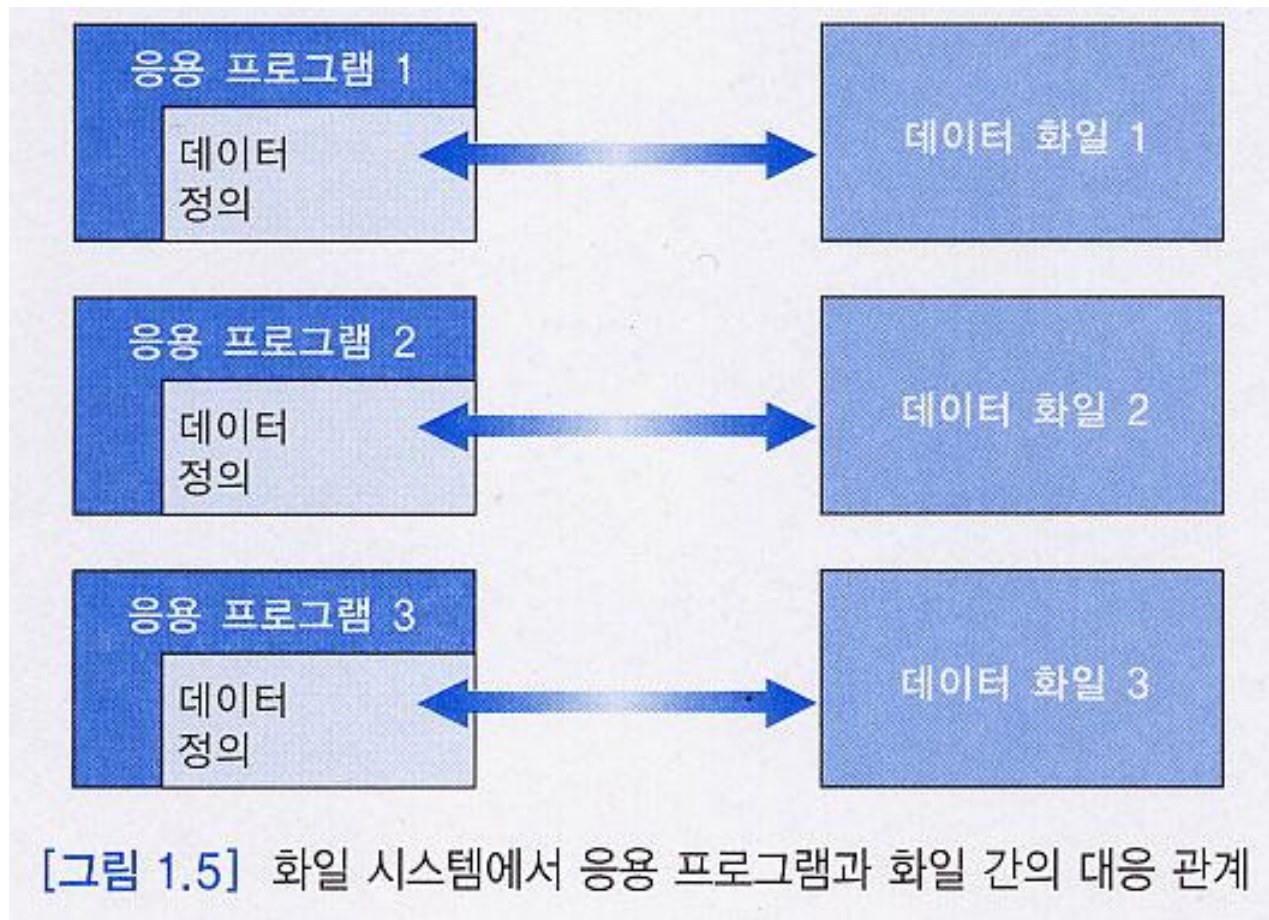
- ✓ 데이터 독립성
- ✓ 효율적인 데이터 접근
- ✓ 데이터에 대한 동시 접근
- ✓ 백업과 회복
- ✓ 중복을 줄이거나 제어하며 일관성 유지
- ✓ 데이터 무결성
- ✓ 데이터 보안
- ✓ 쉬운 질의어
- ✓ 다양한 사용자 인터페이스

1.2 화일 시스템 vs. DBMS

□ 화일 시스템을 사용한 기존의 데이터 관리

- ✓ 화일 시스템은 DBMS가 등장하지 않았을 때인 1960년대부터 사용되어 왔음
- ✓ 화일의 기본적인 구성요소는 순차적인 레코드들
- ✓ 한 레코드는 연관된 필드들의 모임
- ✓ 화일을 접근하는 방식이 응용 프로그램 내에 상세하게 표현되므로 데이터에 대한 응용 프로그램의 의존도가 높음

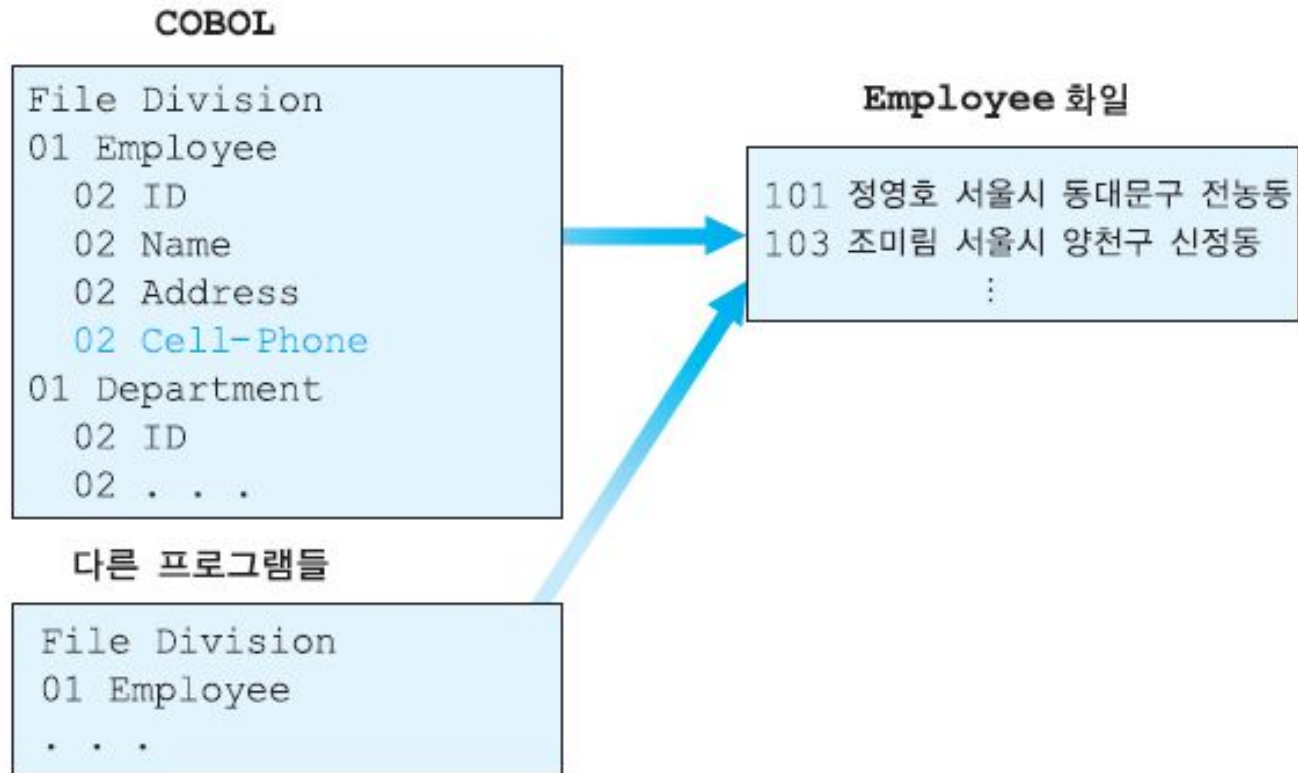
1.2 화일 시스템 vs. DBMS(계속)



1.2 화일 시스템 vs. DBMS(계속)

그림 1.6에서 두 개의 코볼 프로그램에는 Employee 화일의 필드들이 열거되어 있다. 만일 Employee 화일에 사원의 휴대폰 번호를 추가로 나타내려면 Employee 화일의 레코드를 하나씩 읽어서, 휴대폰 번호 필드를 추가한 레코드를 새로운 Employee 화일에 기록하는 프로그램을 작성해야 한다. 그 다음에 기존의 Employee 화일을 사용하던 모든 응용 프로그램들을 찾아서 휴대폰 번호 필드를 추가해야 한다.

1.2 화일 시스템 vs. DBMS(계속)



[그림 1.6] 응용 프로그램과 데이터 화일의 대응 예

1.2 화일 시스템 vs. DBMS(계속)

❑ 화일 시스템의 단점

- ✓ 데이터가 많은 화일에 중복해서 저장됨

예 :

그림 1.7에서 보는 것처럼 기업의 인사 관리 응용 프로그램에 사용되는 EMPLOYEE 화일과 사원 교육 관리 응용 프로그램에서 사용되는 ENROLLMENT 화일에 DEPARTMENT가 중복되어 나타날 수 있다. 어떤 사원의 DEPARTMENT 필드 값이 바뀔 때 두 화일에서 모두 수정하지 않으면 동일한 사원의 소속 부서가 화일마다 다르게 되어 데이터의 불일치가 발생한다.

EMPLOYEE 화일(인사 관리 프로그램용)

NAME	JUMIN-NO	DEPARTMENT	...	ADDRESS
------	----------	------------	-----	---------

ENROLLMENT 화일(교육 관리 프로그램용)

NAME	JUMIN-NO	DEPARTMENT	...	COURSE
------	----------	------------	-----	--------

[그림 1.7] 두 화일에서 DEPARTMENT가 중복됨

1.2 화일 시스템 vs. DBMS(계속)

□ 화일 시스템의 단점(계속)

- ✓ 다수 사용자들을 위한 동시성 제어가 제공되지 않음
- ✓ 검색하려는 데이터를 쉽게 명시하는 질의어가 제공되지 않음
- ✓ 보안 조치가 미흡
- ✓ 회복 기능이 없음
- ✓ 프로그램-데이터 독립성이 없으므로 유지보수 비용이 많이 소요됨
- ✓ 화일을 검색하거나 갱신하는 절차가 상대적으로 복잡하기 때문에 프로그래머의 생산성이 낮음
- ✓ 데이터의 공유와 융통성이 부족

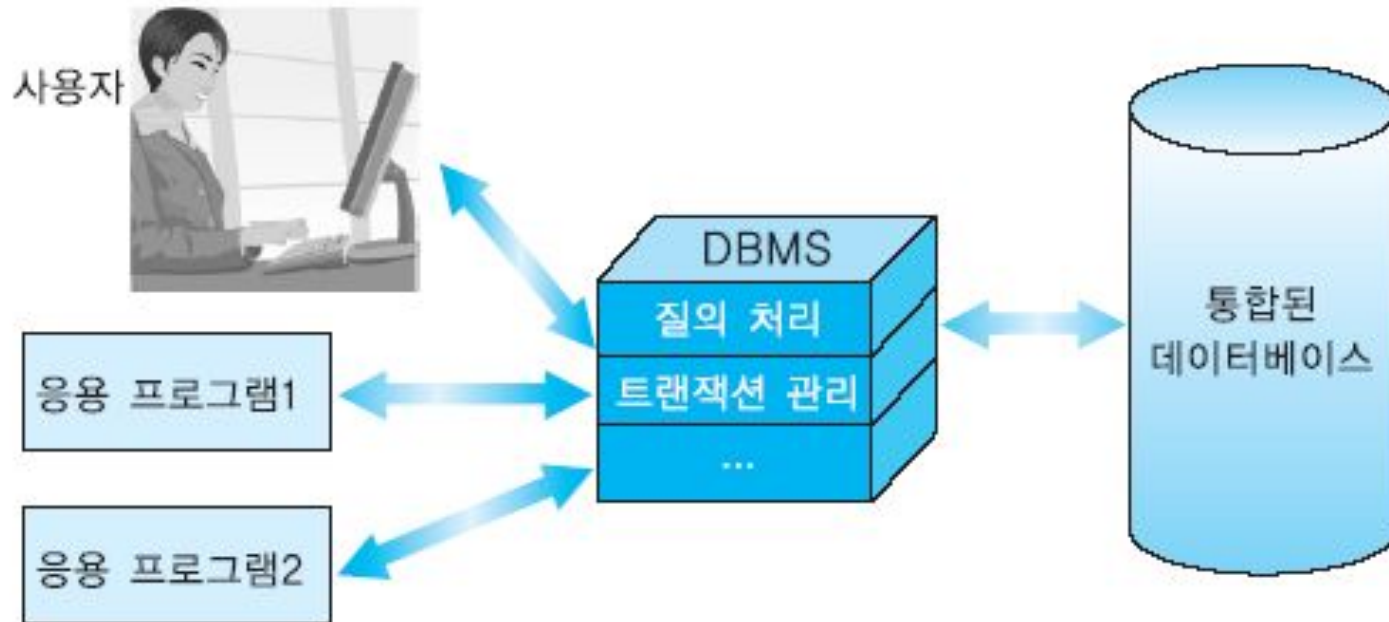
1.2 화일 시스템 vs. DBMS(계속)

□ DBMS를 사용한 데이터베이스 관리

- ✓ 여러 사용자와 응용 프로그램들이 데이터베이스를 공유
- ✓ 사용자의 질의를 빠르게 수행할 수 있는 인덱스 등의 접근 경로를 DBMS가 자동적으로 선택하여 수행
- ✓ 권한이 없는 사용자로부터 데이터베이스를 보호
- ✓ 여러 사용자에게 적합한 다양한 인터페이스를 제공
- ✓ 데이터 간의 복잡한 관계를 표현하며, 무결성 제약조건을 DBMS가 자동적으로 유지
- ✓ 시스템이 고장 나면 데이터베이스를 고장 전의 일관된 상태로 회복시킴
- ✓ 프로그램에 영향을 주지 않으면서 데이터베이스 구조를 변경할 수 있음

프로그램-데이터 독립성(program-data independence)

1.2 화일 시스템 vs. DBMS(계속)



[그림 1.8] DBMS를 사용한 데이터베이스 관리

1.2 화일 시스템 vs. DBMS(계속)

□ DBMS의 장점

- ✓ 중복성과 불일치가 감소됨
- ✓ 시스템을 개발하고 유지하는 비용이 감소됨
- ✓ 표준화를 시행하기가 용이
- ✓ 보안이 향상됨
- ✓ 무결성이 향상됨
- ✓ 조직체의 요구사항을 식별할 수 있음
- ✓ 다양한 유형의 고장으로부터 데이터베이스를 회복할 수 있음
- ✓ 데이터베이스의 공유와 동시 접근이 가능함

1.2 화일 시스템 vs. DBMS(계속)

〈표 1.1〉 화일 시스템 방식과 DBMS 방식의 비교

화일 시스템 방식	DBMS 방식
데이터에 대한 물리적 접근만 조정한다.	데이터에 대한 물리적 접근과 논리적인 접근을 모두 조정한다.
동일한 화일을 두 개 이상의 프로그램이 동시에 접근할 수 없다.	동일한 데이터를 다수 사용자가 동시에 접근할 수 있다.
데이터가 비구조적이며, 중복성과 유지보수 비용이 높다.	데이터가 구조화되어 있으며, 중복성과 유지보수 비용이 낮다.
어떤 프로그램이 기록한 데이터는 다른 프로그램에서 읽을 수 없는 경우가 많다.	접근 권한이 있는 모든 프로그램이 데이터를 공유한다.
데이터에 대한 접근은 미리 작성된 프로그램을 통해서만 가능하다.	질의어를 사용하여 데이터에 대한 융통성 있는 접근이 가능하다.
각 응용 프로그램마다 화일이 따로 있으므로 데이터가 통합되어 있지 않다.	데이터가 중복을 배제하면서 통합되어 있다.

1.2 화일 시스템 vs. DBMS(계속)

□ DBMS 선정시 고려 사항

✓ 기술적 요인

- DBMS에 사용되고 있는 데이터 모델, DBMS가 지원하는 사용자 인터페이스, 프로그래밍 언어, 응용 개발 도구, 저장 구조, 성능, 접근 방법 등

✓ 경제적 요인

- 소프트웨어와 하드웨어 구입 비용, 유지 보수 비용, 직원들의 교육 지원 등

1.2 화일 시스템 vs. DBMS(계속)

❑ DBMS의 단점

- ✓ 추가적인 하드웨어 구입 비용이 들고, DBMS 자체의 구입 비용도 상당히 비쌈
- ✓ 직원들의 교육 비용도 많이 소요됨
- ✓ 비밀과 프라이버시 노출 등의 단점이 존재할 수 있음
- ✓ 초기의 투자 비용이 너무 클 때, 오버헤드가 너무 클 때, 응용이 단순하고 잘 정의되었으며 변경되지 않을 것으로 예상될 때, 엄격한 실시간 처리 요구사항이 있을 때, 데이터에 대한 다수 사용자의 접근이 필요하지 않을 때는 DBMS를 사용하지 않는 것이 바람직할 수 있음

1.4 DBMS 언어

❑ 데이터 정의어(DDL: Data Definition Language)

- ✓ 사용자는 데이터 정의어를 사용하여 데이터베이스 스키마를 정의
- ✓ 데이터 정의어로 명시된 문장이 입력되면 DBMS는 사용자가 정의한 스키마에 대한 명세를 시스템 카탈로그 또는 데이터 사전에 저장
- ✓ 데이터 정의어의 기본적인 기능

데이터 모델에서 지원하는 데이터 구조를 생성

예, SQL에서 CREATE TABLE

데이터 구조의 변경

예, SQL에서 ALTER TABLE

데이터 구조의 삭제

예, SQL에서 DROP TABLE

데이터 접근을 위해 특정 애트리뷰트 위에 인덱스를 정의

예, SQL에서 CREATE INDEX

1.4 DBMS 언어(계속)

❑ 데이터 조작어(DML: Data Manipulation Language)

- ✓ 사용자는 데이터 조작어를 사용하여 데이터베이스 내의 원하는 데이터를 검색하고, 수정하고, 삽입하고, 삭제
- ✓ 절차적 언어(procedural language)와 비절차적 언어(non-procedural language)
- ✓ 관계 DBMS에서 사용되는 SQL은 대표적인 비절차적 언어
- ✓ 대부분의 데이터 조작어는 SUM, COUNT, AVG와 같은 내장 함수들을 갖고 있음
- ✓ 데이터 조작어는 단말기에서 대화식으로 입력되어 수행되거나 C, 코볼 등의 고급 프로그래밍 언어로 작성된 프로그램에 내포되어 사용됨

1.4 DBMS 언어(계속)

❑ 데이터 조작용의 기본적인 기능

데이터의 검색

예, SQL에서 SELECT

데이터의 수정

예, SQL에서 UPDATE

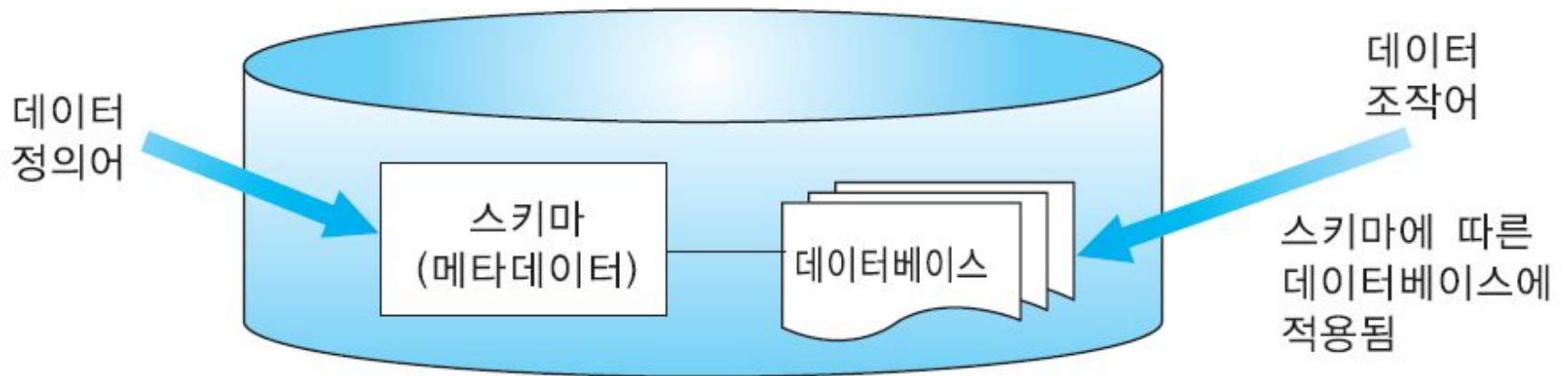
데이터의 삭제

예, SQL에서 DELETE

데이터의 삽입

예, SQL에서 INSERT

1.4 DBMS 언어(계속)



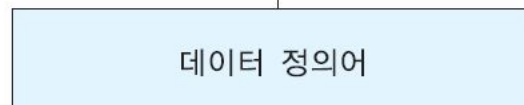
[그림 1.16] 데이터 정의어와 데이터 조작어

1.4 DBMS 언어(계속)

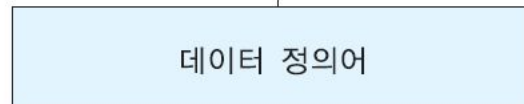
외부 단계



개념 단계



내부 단계



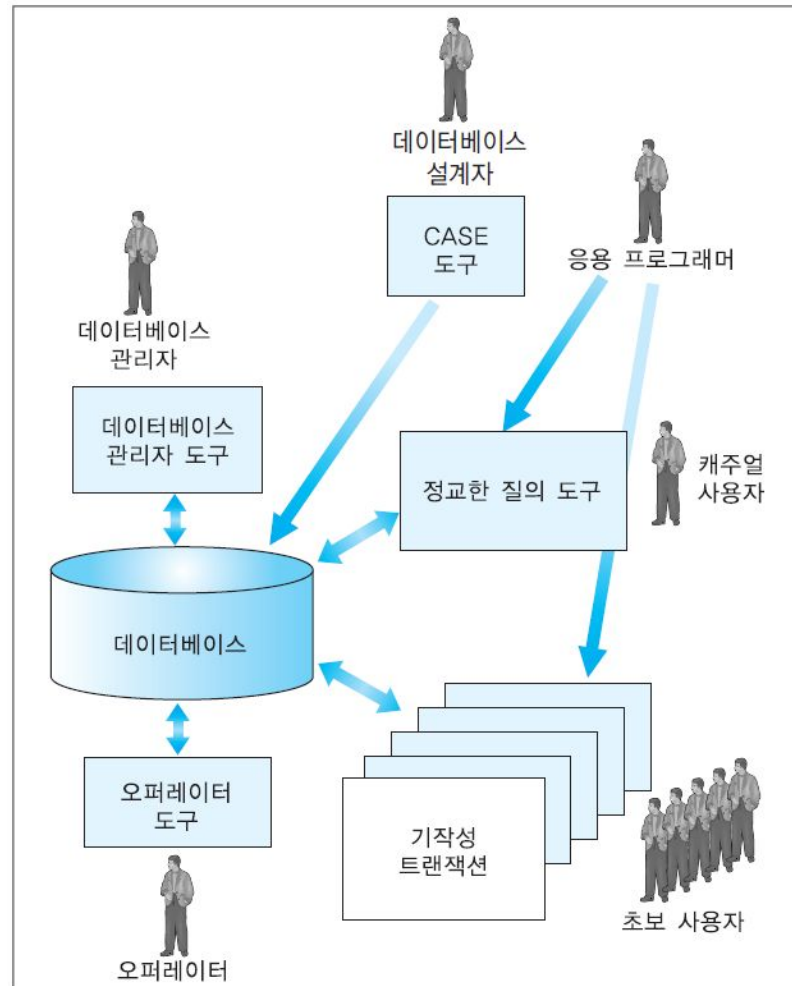
[그림 1.17] ANSI/SPARC 3단계 모델의 각 단계에서 사용되는 데이터 정의어와 데이터 조작어

1.4 DBMS 언어(계속)

❑ 데이터 제어어(DCL: Data Control Language)

- ✓ 사용자는 데이터 제어어를 사용하여 데이터베이스 트랜잭션을 명시하고 권한을 부여하거나 취소

1.5 DBMS 사용자



[그림 1.18] DBMS의 사용자들

1.5 DBMS 사용자(계속)

❑ 데이터베이스 관리자(DBA: Database Administrator)

- ✓ 데이터베이스 관리자는 조직의 여러 부분의 상이한 요구를 만족시키기 위해서 일관성 있는 데이터베이스 스키마를 생성하고 유지하는 사람(팀)
- ✓ 데이터베이스 관리자의 역할
 - 데이터베이스 스키마의 생성과 변경
 - 무결성 제약조건을 명시
 - 사용자의 권한을 허용하거나 취소하고, 사용자의 역할을 관리
 - 저장 구조와 접근 방법(물리적 스키마) 정의
 - 백업과 회복
 - 표준화 시행

1.5 DBMS 사용자(계속)

□ 응용 프로그래머

- ✓ 데이터베이스 위에서 특정 응용(예, 고객 관리, 인사 관리, 재고 관리 등)이나 인터페이스를 구현하는 사람
- ✓ 고급 프로그래밍 언어인 C, 코볼 등으로 응용 프로그램을 개발하면서 데이터베이스를 접근하는 부분은 내포된 데이터 조작어를 사용
- ✓ 이들이 작성한 프로그램은 최종 사용자들이 반복해서 수행하므로 **기작성 트랜잭션**(canned transaction)이라 부름

1.5 DBMS 사용자(계속)

❑ 최종 사용자(end user)

- ✓ 질의하거나 갱신하거나 보고서를 생성하기 위해서 데이터베이스를 사용하는 사람
- ✓ 최종 사용자는 다시 데이터베이스 질의어를 사용하여 매번 다른 정보를 찾는 캐주얼 사용자와 기작성 트랜잭션을 주로 반복해서 수행하는 초보 사용자로 구분

❑ 데이터베이스 설계자(database designer)

- ✓ ERWin 등의 CASE 도구들을 이용해서 데이터베이스 설계를 담당
- ✓ 데이터베이스의 일관성을 유지하기 위해서 정규화를 수행

❑ 오퍼레이터

- ✓ DBMS가 운영되고 있는 컴퓨터 시스템과 전산실을 관리하는 사람

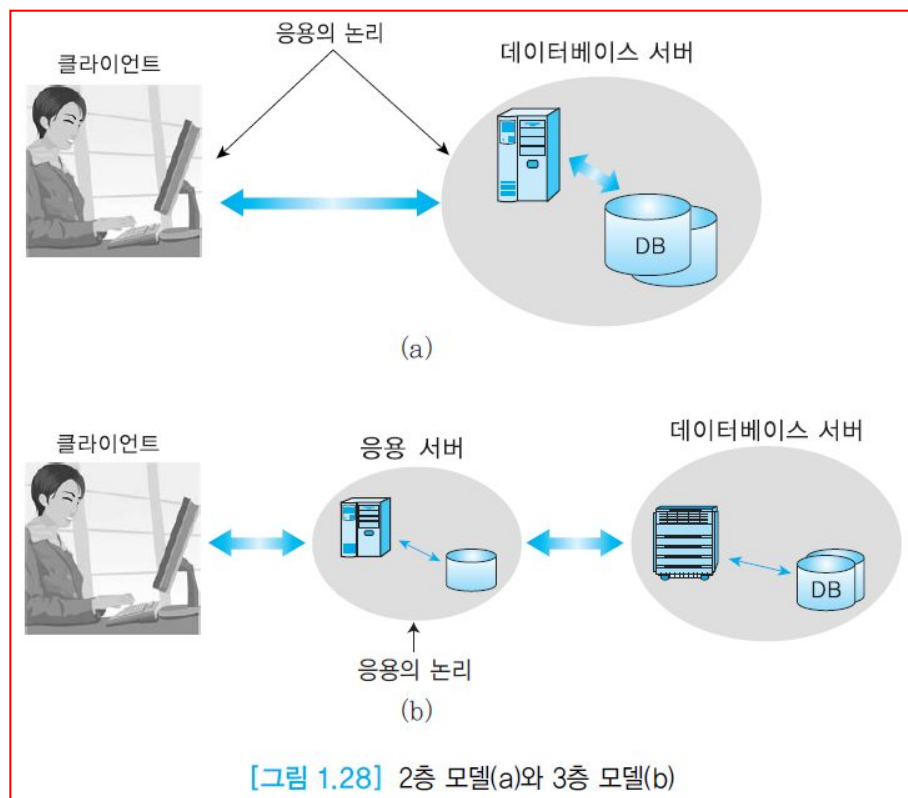
1.7 데이터베이스 시스템 아키텍처(계속)

❑ 클라이언트-서버 데이터베이스 시스템(client-server database system)

- ✓ PC 또는 워크스테이션처럼 자체 컴퓨팅 능력을 가진 클라이언트를 통해 데이터베이스 서버를 접근
- ✓ 데이터베이스가 하나의 데이터베이스 서버에 저장되어 있음
- ✓ 데이터베이스 시스템의 기능이 서버와 클라이언트에 분산됨
- ✓ 서버는 데이터베이스를 저장하고 DBMS를 운영하면서 여러 클라이언트에서 온 질의를 최적화하고, 권한 검사를 수행하고, 동시성 제어와 회복 기능을 수행하고, 데이터베이스의 무결성을 유지하고, 데이터베이스 접근을 관리
- ✓ 클라이언트는 사용자 인터페이스를 관리하고 응용들을 수행

1.7 데이터베이스 시스템 아키텍처(계속)

- ❑ 2층 모델(2-tier model)
 - ✓ 클라이언트와 데이터베이스 서버가 직접 연결됨
- ❑ 3층 모델(3-tier model)
 - ✓ 클라이언트와 데이터베이스 서버 사이에 응용 서버가 추가됨



1.7 데이터베이스 시스템 아키텍처(계속)

❑ 클라이언트-서버 데이터베이스 시스템의 장점

- ✓ 데이터베이스를 보다 넓은 지역에서 접근할 수 있음
- ✓ 다양한 컴퓨터 시스템을 사용할 수 있음

❑ 클라이언트-서버 데이터베이스 시스템의 단점

- ✓ 보안이 다소 취약할 수 있음