

데이터베이스

자바 강의실

I. 데이터베이스 개념

II. 데이터베이스 관리 시스템

III. 데이터베이스 설계

IV. 데이터 모델링

1. 데이터베이스 개념

- 컴퓨터를 사용하여 정보를 수집하고 분석하는데 데이터베이스 기술이 활용되고 있음
- **정보**와 **데이터**는 서로 다름

정보는 관찰이나 측정을 통해 수집된 데이터를 실제 문제에 도움이 될 수 있도록 해석하고 정리한 지식.

데이터는 이론을 세우는 데 기초가 되는 사실 · 자료.

- **데이터베이스(database)의 정의**

데이터베이스는 조직체의 응용 시스템들이 **공유**해서 사용하는 운영 데이터 (operational data)들이 **구조적으로 통합**된 모임이다. 데이터베이스의 구조는 사용되는 데이터 모델에 의해 결정된다.

1. 데이터베이스 개념

- 데이터베이스의 예

예: 데이터베이스 1

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.

예: 데이터베이스 2

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

1. 데이터베이스 개념

- 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

2. 데이터베이스 관리 시스템

- 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

2. 데이터베이스 관리 시스템

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

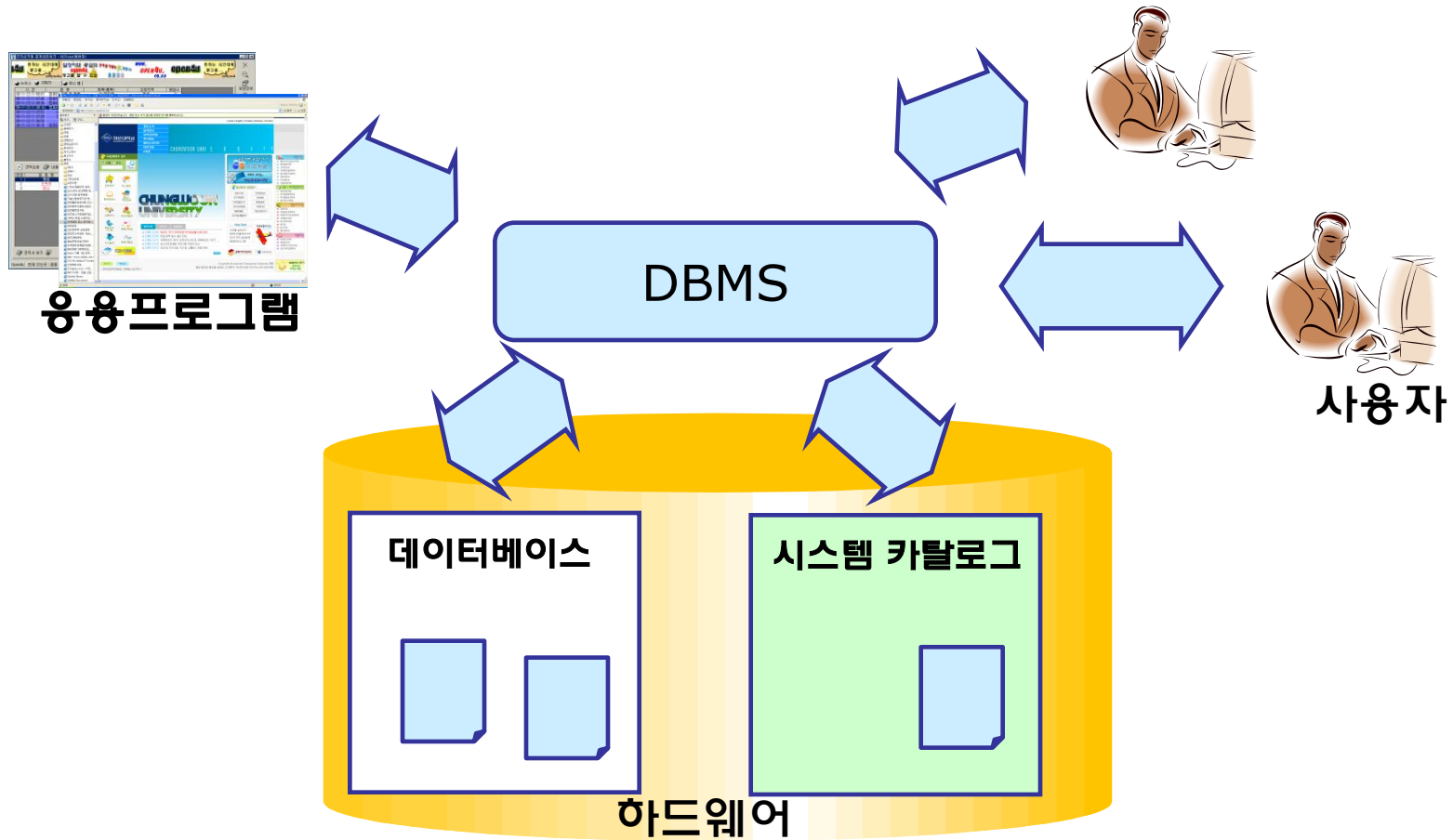
EMPLOYEE

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] 데이터베이스 스키마와 데이터베이스 상태

2. 데이터베이스 관리 시스템

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

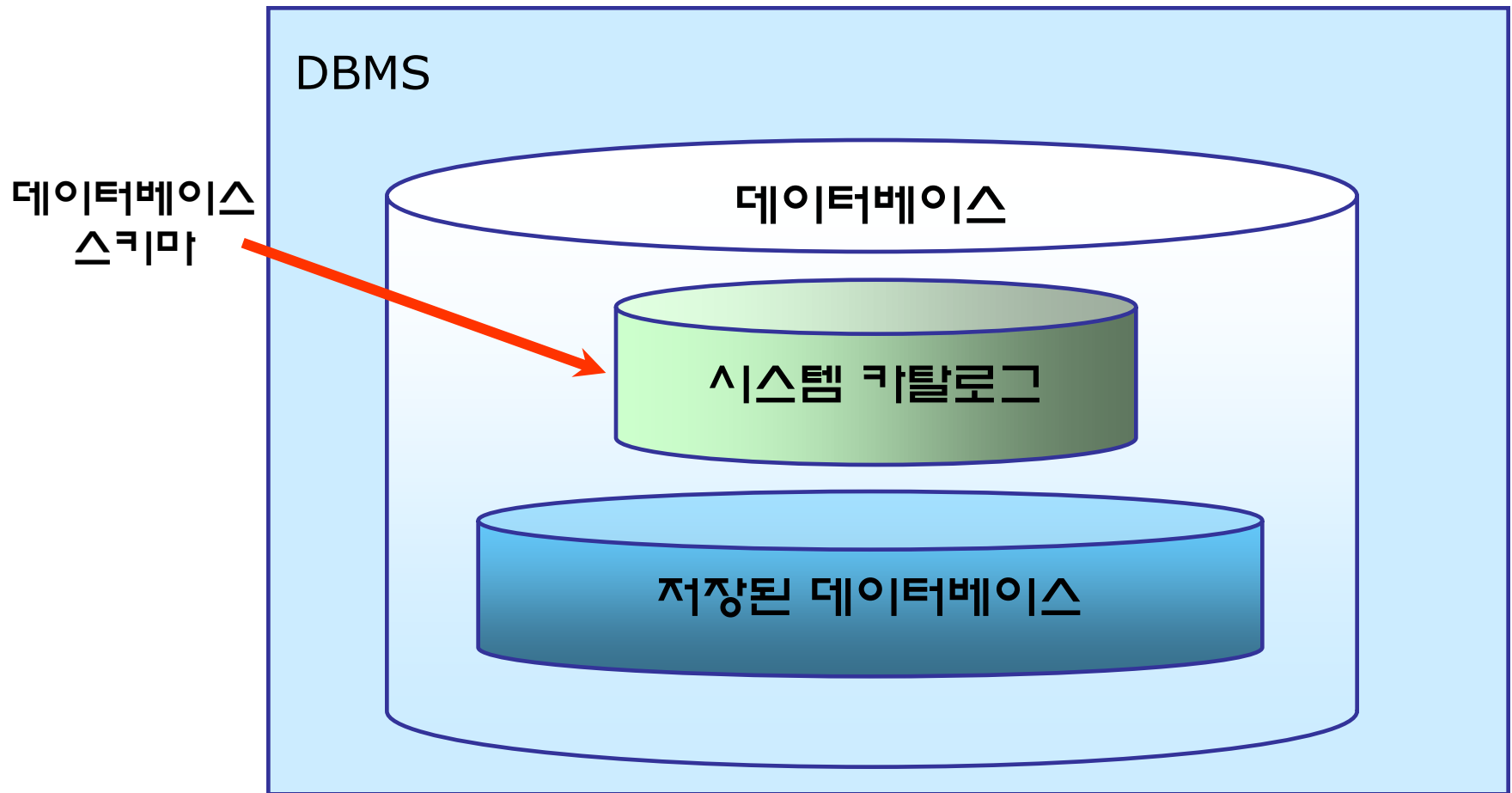


[그림] 데이터베이스 시스템의 구성요소

2. 데이터베이스 관리 시스템

- 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
 - 스키마 정보가 바뀌면 시스템 카탈로그에 반영

2. 데이터베이스 관리 시스템



[그림] 시스템 카탈로그와 저장된 데이터베이스

2. 데이터베이스 관리 시스템

- 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

2. 데이터베이스 관리 시스템



[그림] 컴퓨터 시스템에서 DBMS의 위치

2. 데이터베이스 관리 시스템

- 사용자
 - 데이터베이스 사용자는 여러 부류로 나눌 수 있음
 - DBA, 응용프로그래머, 최종사용자, DB설계자, 오퍼레이터
- 하드웨어
 - 데이터베이스는 디스크와 같은 보조 기억 장치에 저장되며, DBMS에서 원하는 정보를 찾기 위해서는 디스크의 블록들을 주 기억 장치로 읽어 들여야 하며, 계산이나 비교 연산들을 수행하기 위해 중앙 처리 장치가 사용됨
 - DBMS 자체도 주기억 장치에 적재되어 실행되어야 함

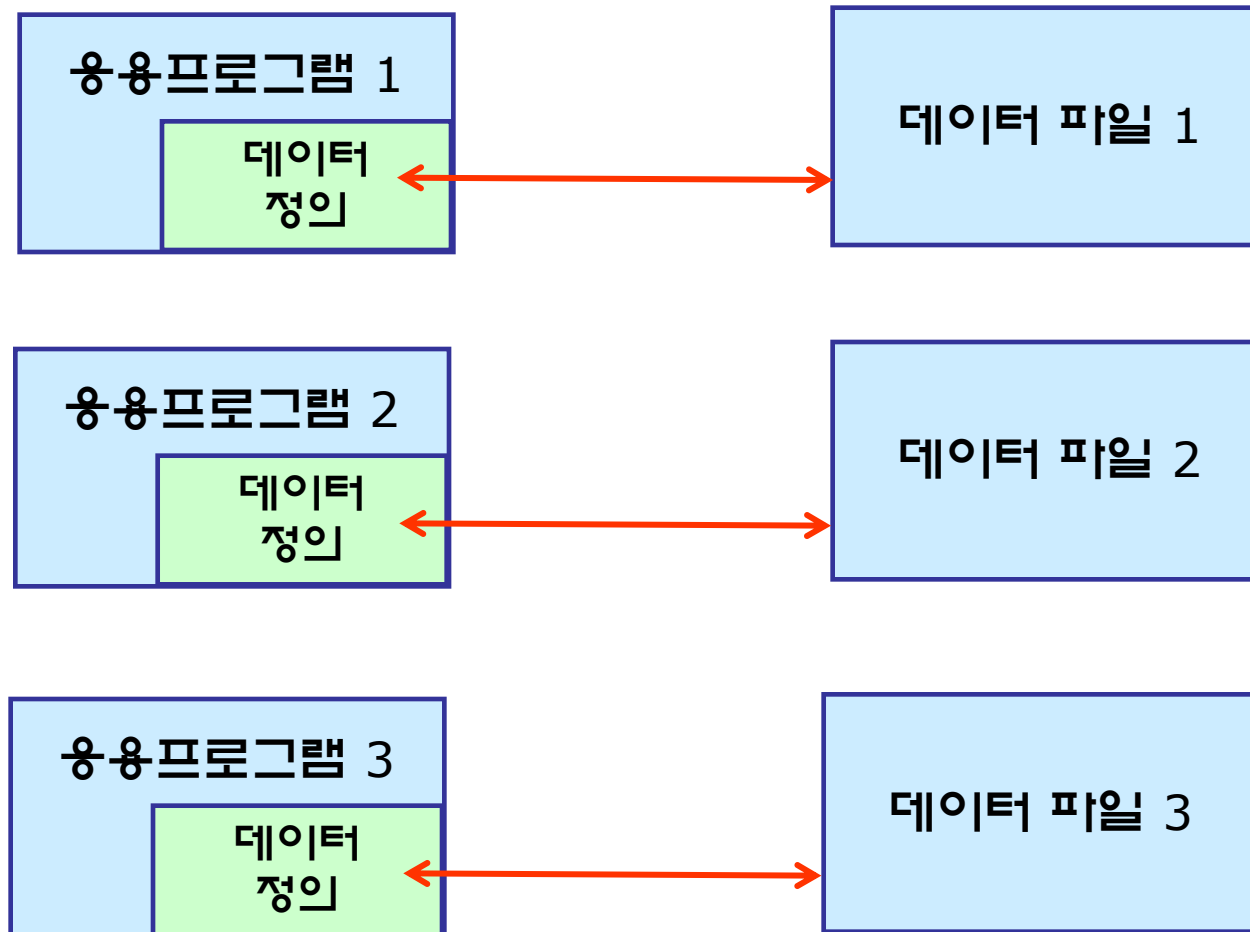
2. 데이터베이스 관리 시스템

- 데이터베이스 시스템의 요구사항
 - 데이터 독립성
 - 융통성
 - 효율적인 데이터 접근
 - 데이터에 대한 동시 접근
 - 백업과 회복
 - 중복을 줄이거나 제어하며 일관성 유지
 - 데이터 무결성
 - 데이터 보안
 - 쉬운 질의어
 - 다양한 사용자 인터페이스의 제공

2. 데이터베이스 관리 시스템

- 파일 시스템을 사용한 기존의 데이터 관리
 - 파일 시스템은 DBMS가 등장하지 않았을 때인 1960년대부터 사용되어 왔음
 - 파일의 기본적인 구성요소는 순차적인 레코드들
 - 한 레코드는 연관된 필드들의 모임
 - 파일을 접근하는 방식이 응용 프로그램 내에 상세하게 표현되므로 데이터에 대한 응용 프로그램의 의존도가 높음

2. 데이터베이스 관리 시스템



[그림] 파일 시스템에서 응용프로그램과 파일간의 대응 관계

2. 데이터베이스 관리 시스템

프로그램과 파일 간의 대응 관계

이전 그림에서 두 개의 비주얼베이직 프로그램에는 Employee 파일의 필드들이 열거되어있다. 만일 Employee 파일에 사원의 휴대폰 번호를 추가로 나타내려면 Employee 파일의 레코드를 하나씩 읽어서, 휴대폰 번호 필드를 추가한 레코드를 새로운 Employee 파일에 기록하는 프로그램을 작성해야 한다. 그 다음에 기존의 Employee 파일을 사용하던 모든 응용 프로그램들을 찾아서 휴대폰 번호 필드를 추가해야 한다.

비주얼 베이직 프로그램

File Division
01 Employee
 02 ID
 02 Name
 02 Address
 02 Cell Phone
01 Department
 02 ID
 02 ...

다른 프로그램들

File Division
01 Employee
...

[이전그림참조]

응용 프로그램과
데이터 파일의 대응
예

Employee 파일

101	정영호	충남	홍성군	홍성읍	오관리
102	조미림	충남	홍성군	홍성읍	학계리
...					

2. 데이터베이스 관리 시스템

- 파일 시스템의 단점

- 데이터가 많은 파일에 중복해서 저장됨
- 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
- 데이터 모델링 개념이 부족
- 무결성을 유지하기 어려움
- 파일을 검색하거나 갱신하는 절차가 상대적으로 복잡하기 때문에 프로그래머의 생산성이 낮음
- 데이터의 공유와 융통성이 부족

2. 데이터베이스 관리 시스템

파일 시스템의 단점 예

이전 그림에서 보는 것처럼 기업의 인사 관리 응용 프로그램에 사용되는 EMPLOYEE 파일과 사원 교육 관리 응용 프로그램에서 사용되는 ENROLLMENT 파일에 DEPARTMENT가 중복되어 나타날 수 있다. 어떤 사원의 DEPARTMENT 필드 값이 바뀔 때 두 파일에서 모두 수정하지 않으면 동일한 사원의 소속 부서가 파일마다 다르게 되어 데이터의 불일치가 발생한다.

EMPLOYEE 파일 (인사 관리 프로그램용)

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

ENROLLMENT 파일 (교육 관리 프로그램용)

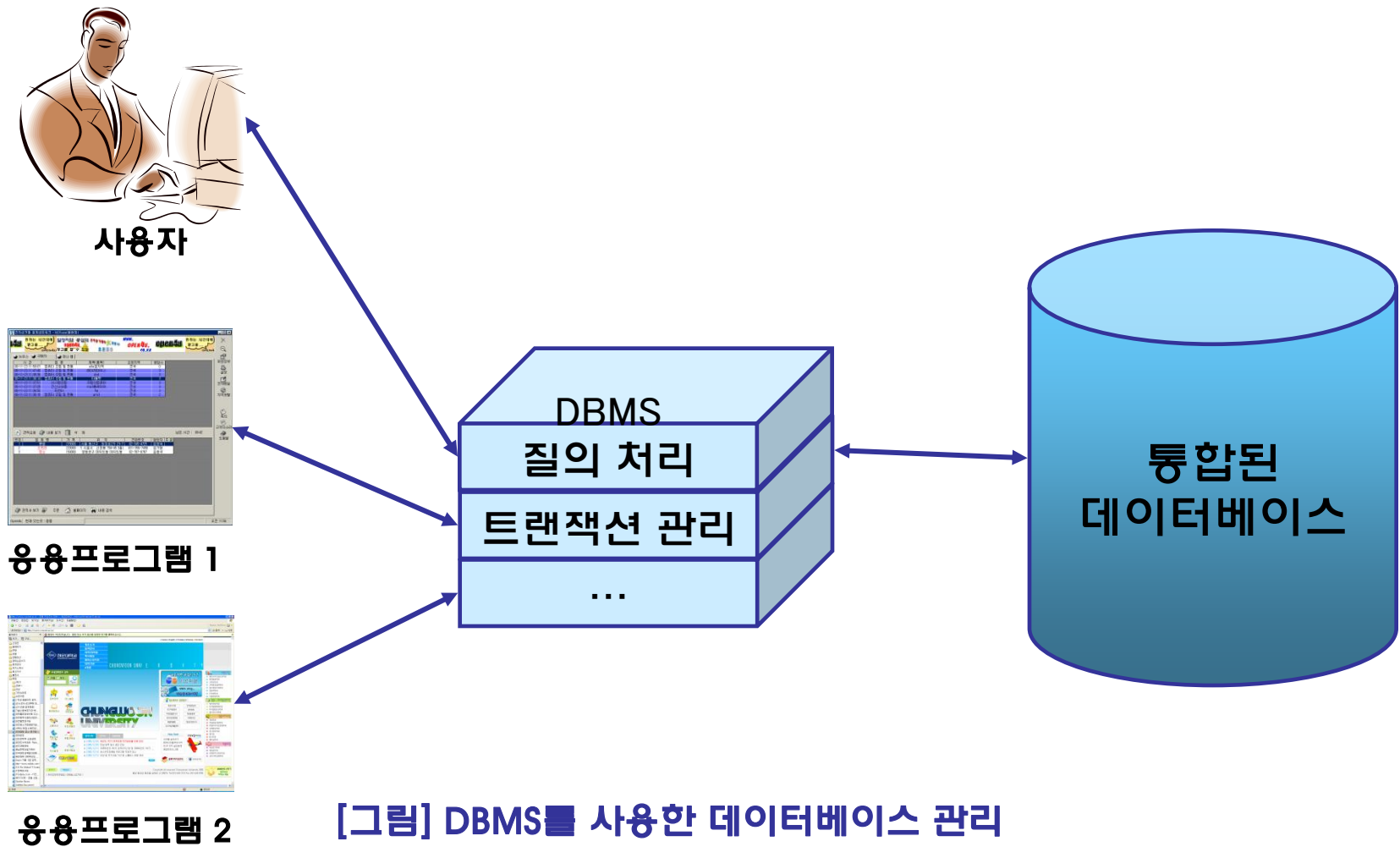
NAME	JUMIN-NO	DEPARTMENT	...	ADDRESS
------	----------	------------	-----	---------

[그림] 두 파일에서 DEPARTMENT가 중복됨

2. 데이터베이스 관리 시스템

- 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
 - 데이터베이스는 표준화된 형식으로 저장되며 통합된 데이터베이스에 대한 접근이 모두 DBMS를 통하여 이루어짐
 - 프로그램에 영향을 주지 않으면서 데이터베이스 구조를 변경할 수 있음
 - 프로그램-데이터 독립성(program-data independence)

2. 데이터베이스 관리 시스템



2. 데이터베이스 관리 시스템

- DBMS의 장점
 - 중복성과 불일치가 감소됨
 - 사용자에게 보다 나은 서비스가 제공됨
 - 시스템의 융통성이 향상됨
 - 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

2. 데이터베이스 관리 시스템

[표] 파일 시스템 방식과 DBMS 방식의 비교

파일 시스템 방식	DBMS 방식
데이터에 대한 물리적 접근만 조정한다	데이터에 대한 물리적 접근과 논리적인 접근을 모두 조정한다
동일한 파일을 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	접근권한이 있는 모든 프로그램이 데이터를 공유한다
데이터에 대한 접근은 미리 작성된 프로그램을 통해서만 가능하다	질의어를 사용하여 데이터에 대한 융통성 있는 접근이 가능하다
각 응용 프로그램마다 파일이 따로 있으므로 데이터가 통합되어 있지 않다	데이터가 중복을 배제하면서 통합되어 있다

2. 데이터베이스 관리 시스템

- DBMS 선정 시 고려 사항

- 기술적 요인

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

- 경제적 요인

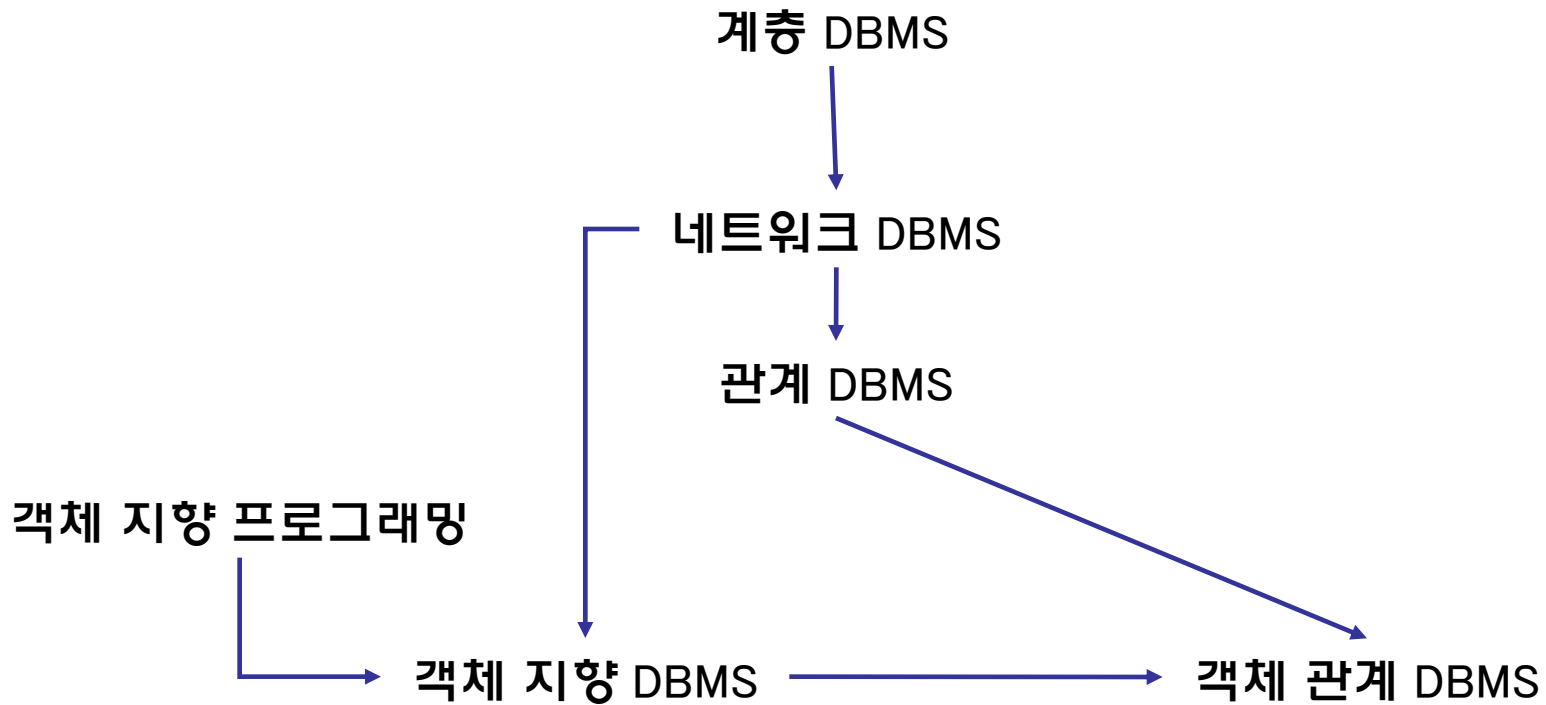
- 소프트웨어와 하드웨어 구입 비용, 유지 보수 비용, 직원들의 교육 지원 등

2. 데이터베이스 관리 시스템

- DBMS의 단점

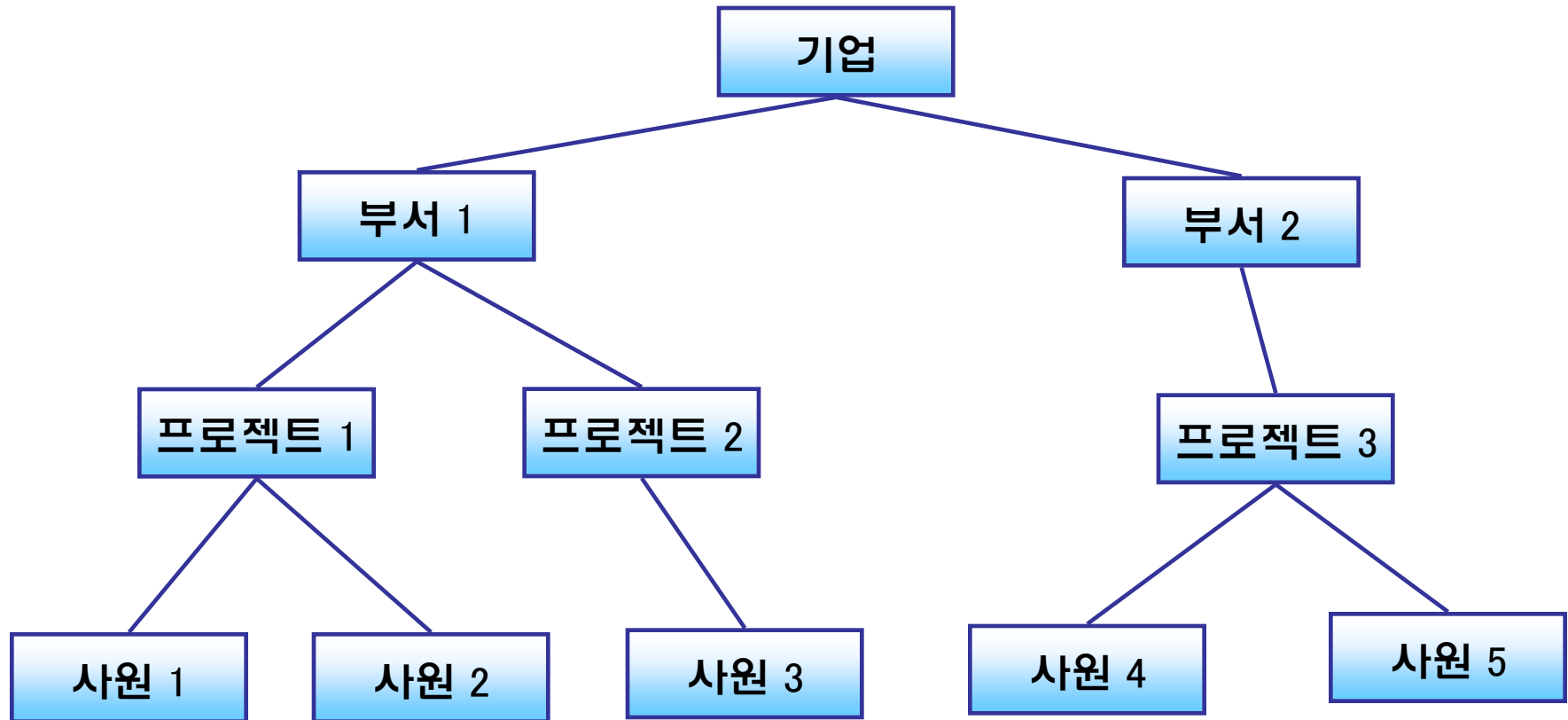
- 추가적인 하드웨어 구입 비용이 들고, DBMS 자체의 구입 비용도 상당히 비쌈
- 직원들의 교육 비용도 많이 소요됨
- 응답 시간이 많이 걸릴 수 있음
- 파일 방식에 비해서 고장의 영향을 더 크게 받을 수 있음
- 비밀과 프라이버시 노출 등의 단점이 존재할 수 있음
- 초기의 투자 비용이 너무 클 때, 오버헤드가 너무 클 때, 응용이 단순하고 잘 정의되었으며 변경되지 않을 것으로 예상될 때, 엄격한 실시간 처리 요구사항이 있을 때, 데이터에 대한 다수 사용자의 접근이 필요하지 않을 때는 DBMS를 사용하지 않는 것이 바람직할 수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

2. 데이터베이스 관리 시스템



[그림] DBMS의 발전과정

2. 데이터베이스 관리 시스템



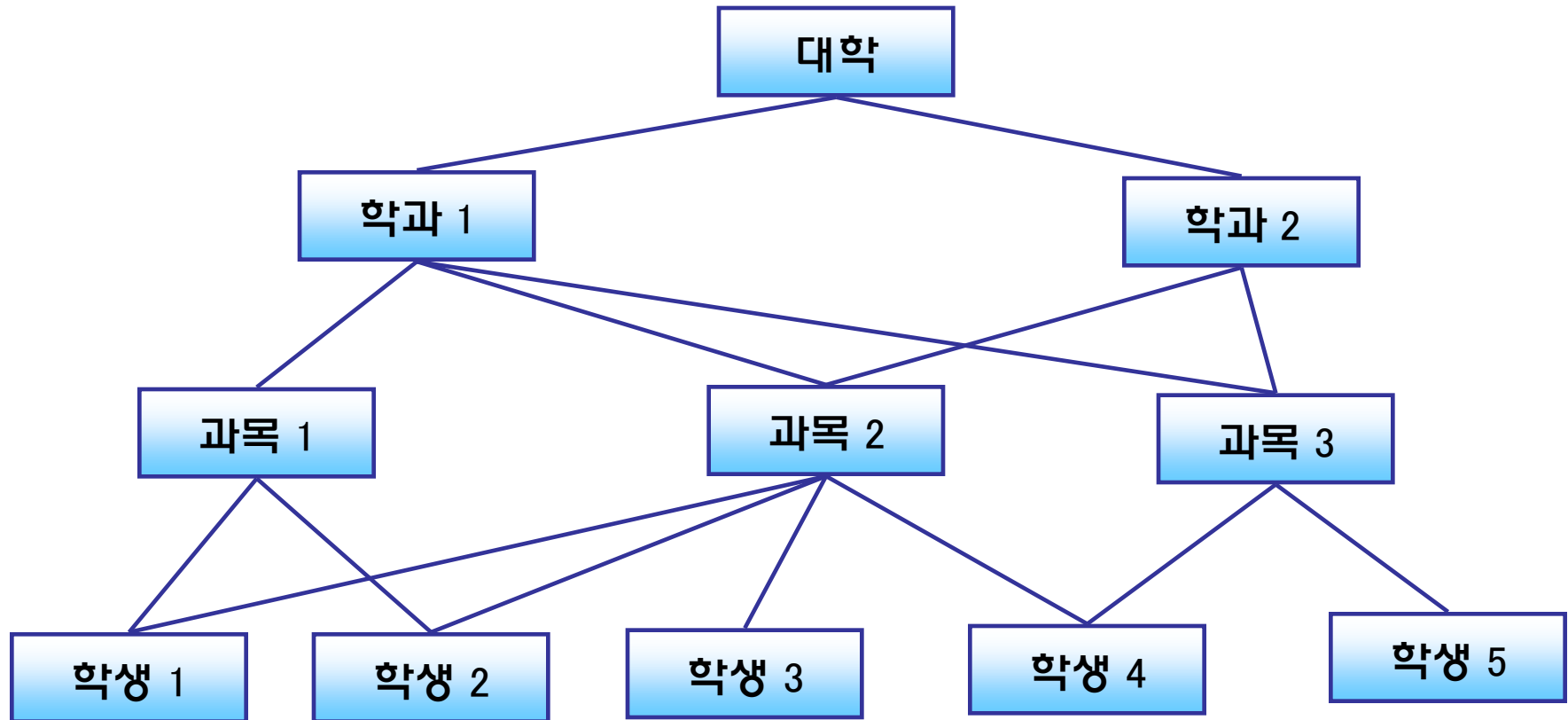
[그림] 계층 데이터베이스의 예

2. 데이터베이스 관리 시스템

- 네트워크 DBMS

- 1960년대 초에 Charles Bachman이 하니웰(Honeywell)사에서 최초의 네트워크 DBMS인 IDS를 개발
- 레코드들이 노드로, 레코드들 사이의 관계가 간선으로 표현되는 그래프를 기반으로 하는 네트워크 데이터 모델을 사용
- 네트워크 DBMS에서도 레코드들이 링크로 연결되어 있으므로 레코드 구조를 변경하기 어려움

2. 데이터베이스 관리 시스템



[그림] 네트워크 데이터베이스의 예

2. 데이터베이스 관리 시스템

- 관계 DBMS

- 1970년에 E.F. Codd가 IBM 연구소에서 관계 데이터 모델을 제안
- 미국 IBM 연구소에서 진행된 System R과 캘리포니아 버클리대에서 진행된 Ingres 프로젝트
- 장점
 - 모델이 간단하여 이해하기 쉬움
 - 사용자는 자신이 원하는 것(what)만 명시하고, 데이터가 어디에 있는지, 어떻게 접근해야 하는지는 DBMS가 결정
- 예: Oracle, MS SQL Server, Sybase, DB2, Informix 등

2. 데이터베이스 관리 시스템

- 객체 지향 DBMS

- 1980년대 후반 들어 새로운 데이터 모델인 객체 지향 데이터 모델이 등장
- 객체 지향 프로그래밍 패러다임을 기반으로 하는 데이터 모델
- 장점
 - 데이터와 프로그램을 그룹화하고, 복잡한 객체들을 이해하기 쉬우며, 유지와 변경이 용이함
- 예: ONTOS, OpenODB, GemStone, ObjectStore, Versant, O2 등

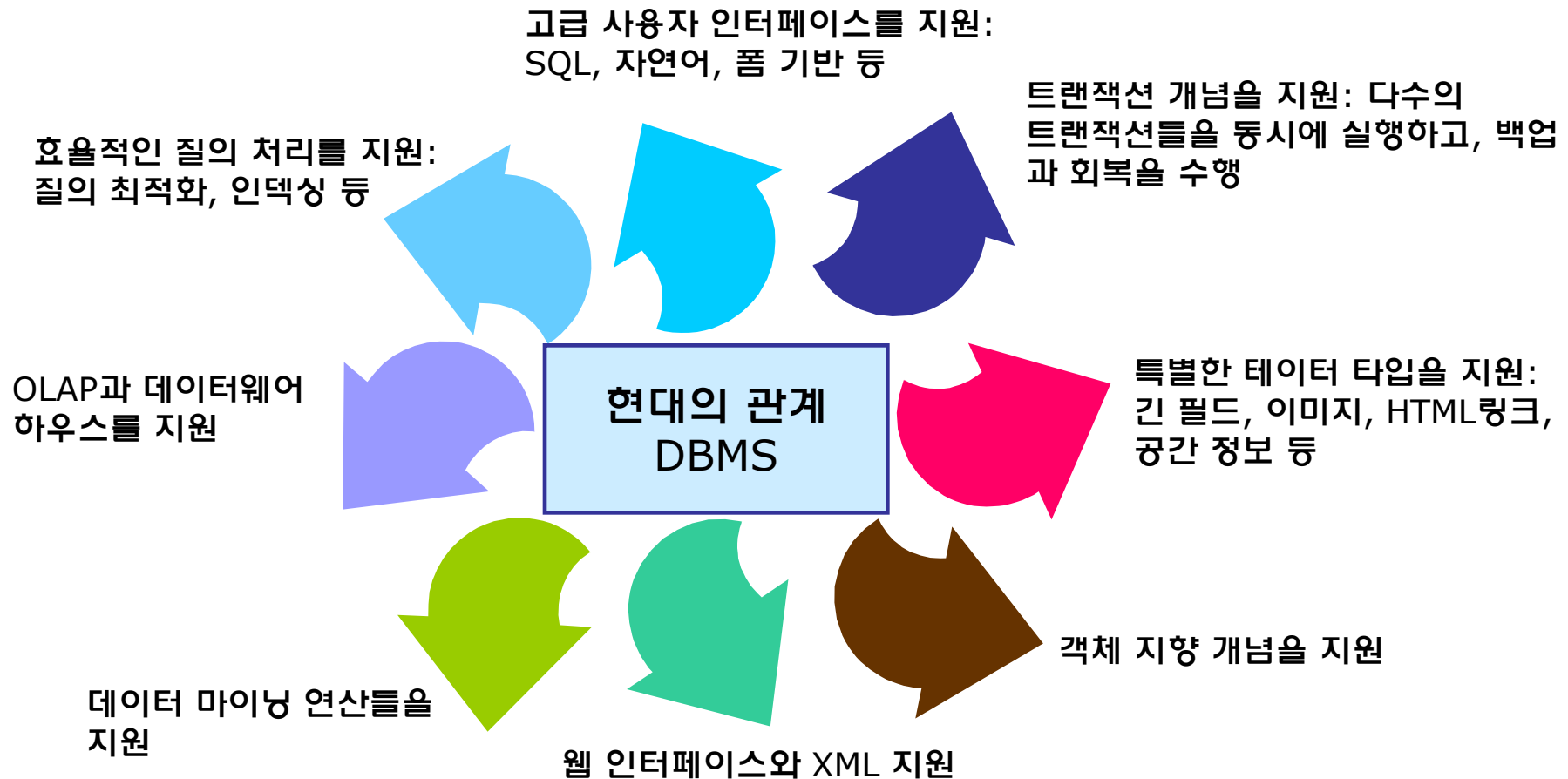
- 객체 관계 DBMS

- 1990년대 후반에 관계 DBMS에 객체 지향 개념을 통합한 객체 관계 데이터 모델이 제안됨
- 예: Oracle 9i, Informix Universal Server 등

2. 데이터베이스 관리 시스템

- 새로운 데이터베이스 응용
 - CAD 데이터베이스, 소프트웨어 공학 데이터베이스(재사용이 가능한 소프트웨어들의 라이브러리), 게놈 데이터베이스, 데이터 웨어하우스, 데이터 마이닝, OLAP, 멀티미디어 데이터베이스, 웹 데이터베이스 등

2. 데이터베이스 관리 시스템



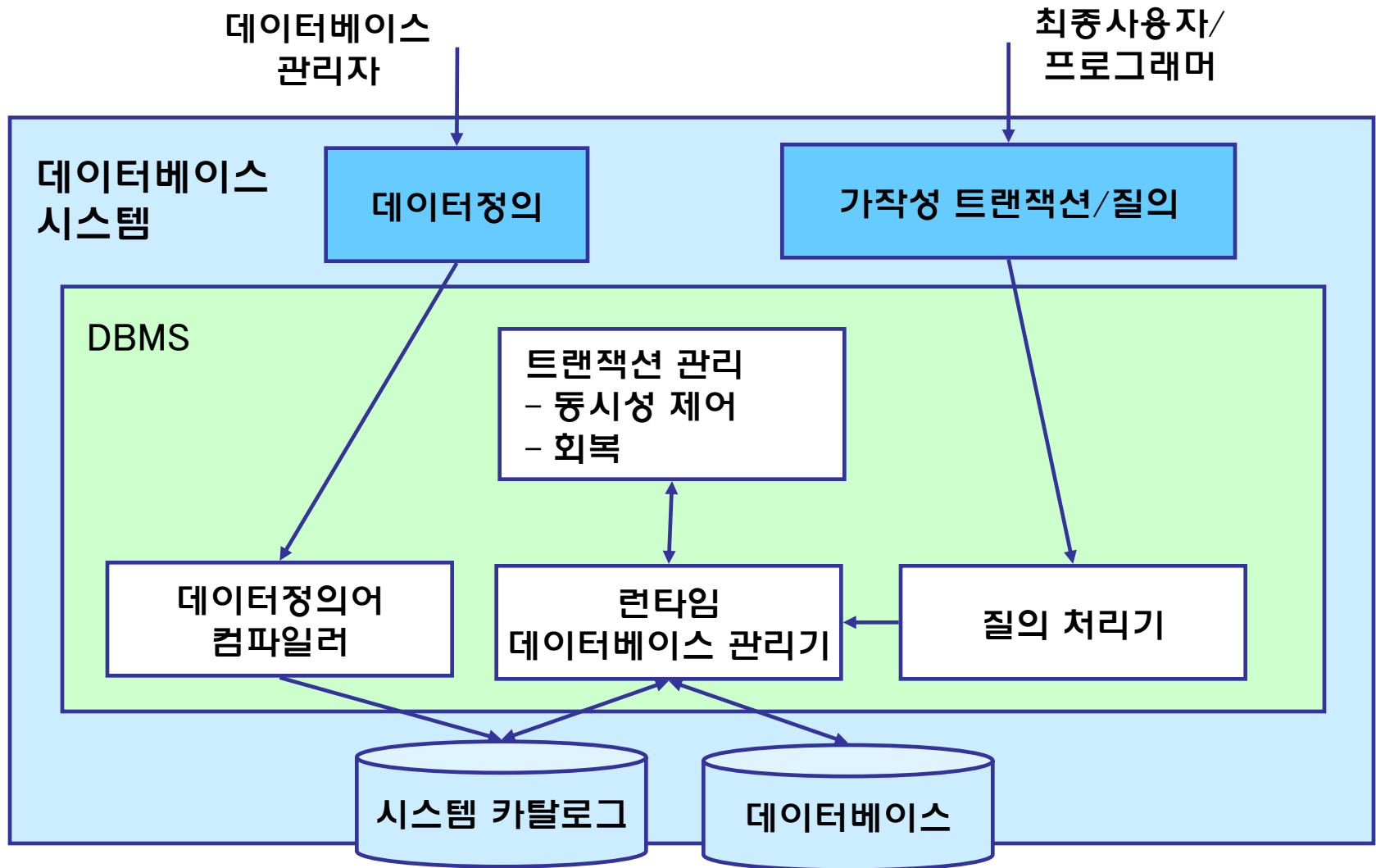
[그림] 현대의 관계 DBMS의 기능

2. 데이터베이스 관리 시스템

[표] DBMS들의 분류

기준	종류
데이터 모델에 따른 분류	<ul style="list-style-type: none">• 계층 DBMS• 네트워크 DBMS• 관계 DBMS• 객체 지향 DBMS• 객체 관계 DBMS
사용자의 수에 따른 분류	<ul style="list-style-type: none">• 단일 사용자 DBMS(주로 PC 용)• 다수 사용자 DBMS
사이트의 수에 따른 분류	<ul style="list-style-type: none">• 중앙 집중식 DBMS• 분산 DBMS
접근 방법에 따른 분류	<ul style="list-style-type: none">• 범용 DBMS• 특별한 DBMS(예: 공간 DBMS)

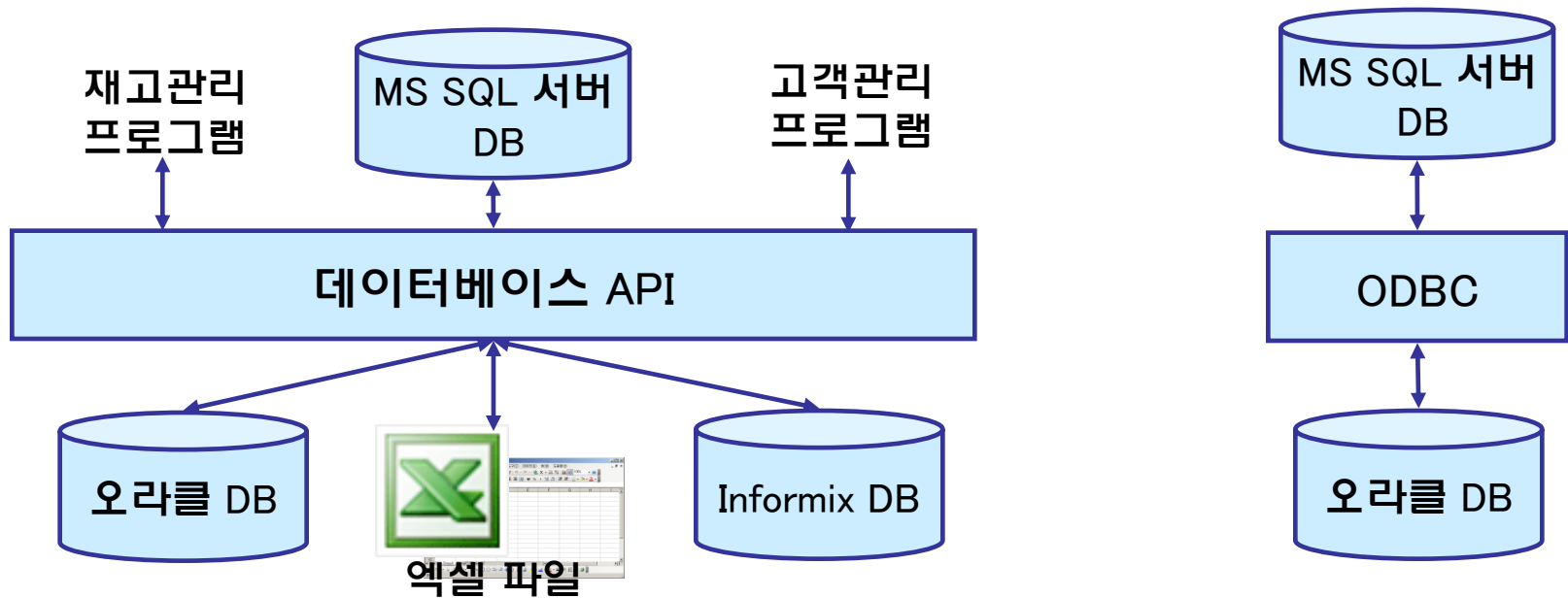
3. 데이터베이스 설계



[그림] 데이터베이스 시스템의 간단한 아키텍처

3. 데이터베이스 설계

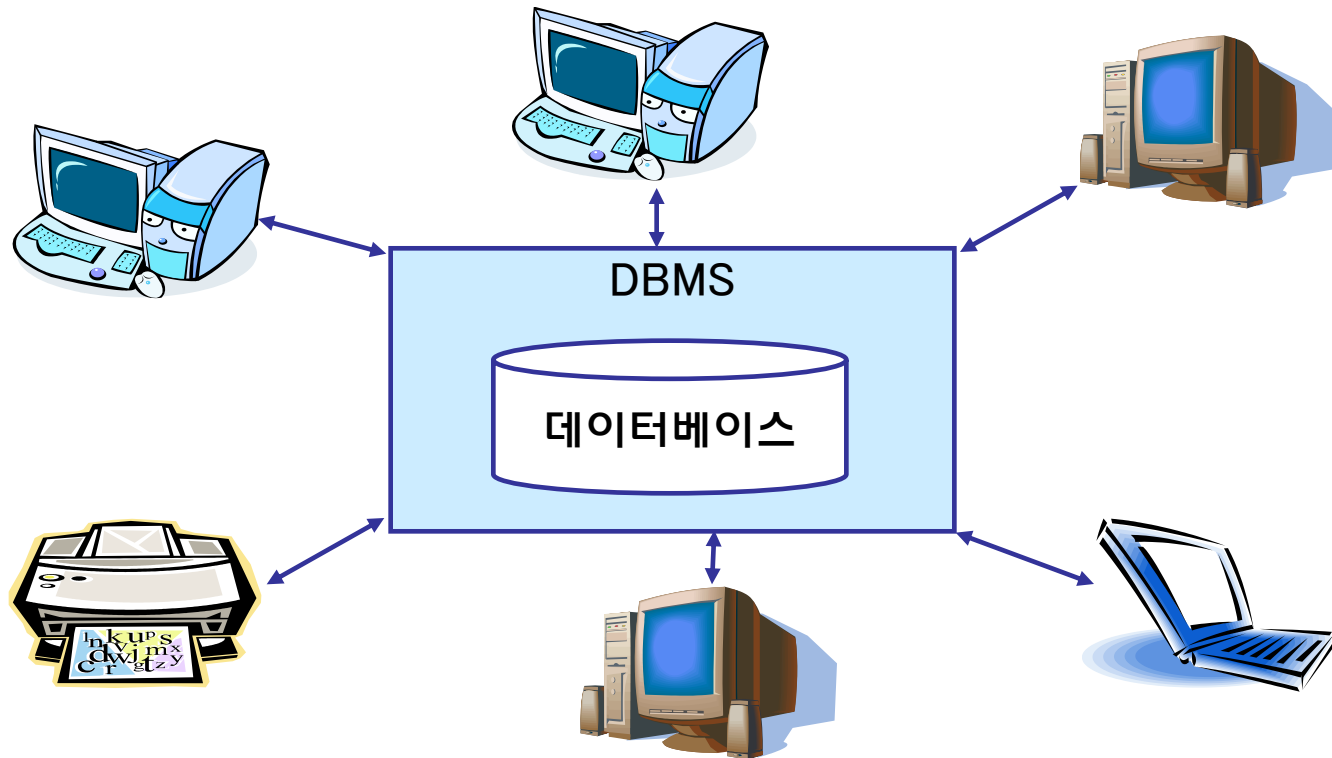
- 데이터베이스 API(Application Program Interface)
 - ODBC(Open Database Connectivity)는 마이크로소프트 사가 주도적으로 개발한 데이터베이스 API
 - ODBC를 지원하는 DBMS 간에는 서로 상대방의 데이터베이스를 접근할 수 있음



[그림] 데이터 베이스 API(ODBC)의 역할

3. 데이터베이스 설계

- 중앙 집중식 데이터베이스 시스템(centralized database system)
 - 데이터베이스 시스템이 하나의 컴퓨터 시스템에서 운영됨

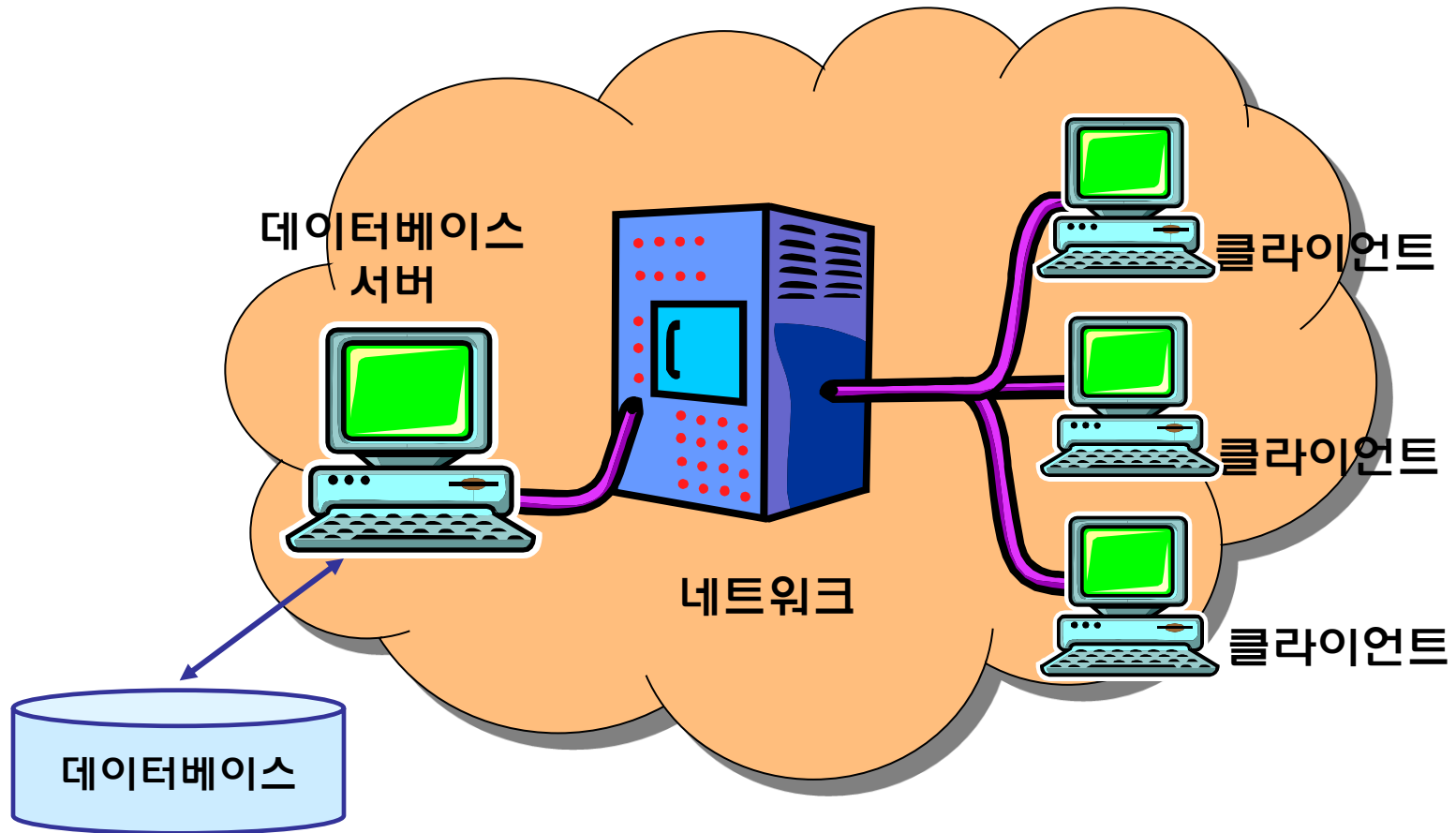


[그림] 중앙 집중식 데이터베이스 시스템

3. 데이터베이스 설계

- 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

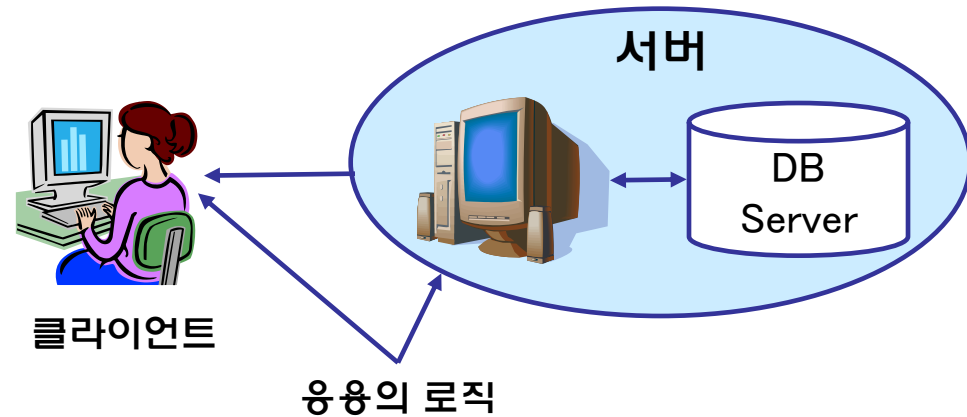
3. 데이터베이스 설계



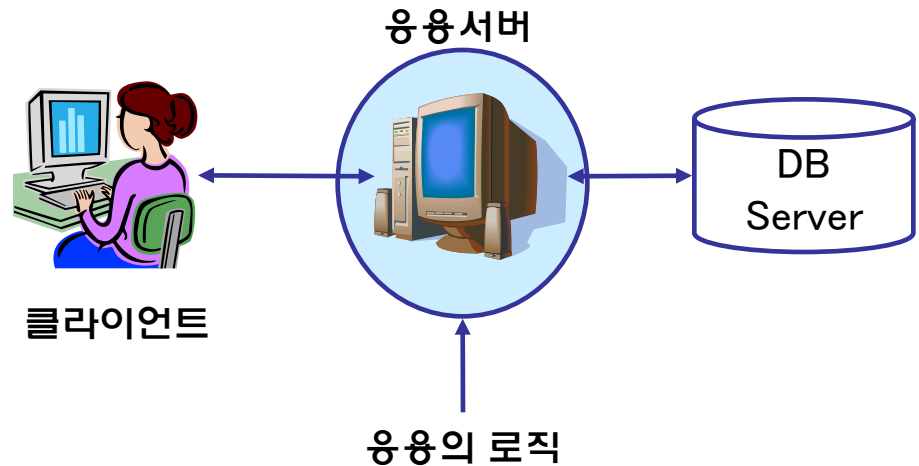
[그림] 클라이언트-서버 데이터베이스 시스템

3. 데이터베이스 설계

- 2계층 모델(2-tier model)
 - 클라이언트와 데이터베이스
 - 서버가 직접 연결됨



- 3계층 모델(3-tier model)
 - 클라이언트와 데이터베이스
 - 서버 사이에 응용 서버가 추가 됨



[그림] 2계층 모델과 3계층 모델

3. 데이터베이스 설계

- 클라이언트-서버 데이터베이스 시스템의 장점
 - 데이터베이스를 보다 넓은 지역에서 접근할 수 있음
 - 하드웨어 비용이 절감됨
 - 다양한 컴퓨터 시스템을 사용할 수 있음
- 클라이언트-서버 데이터베이스 시스템의 단점
 - 보안이 다소 취약할 수 있음

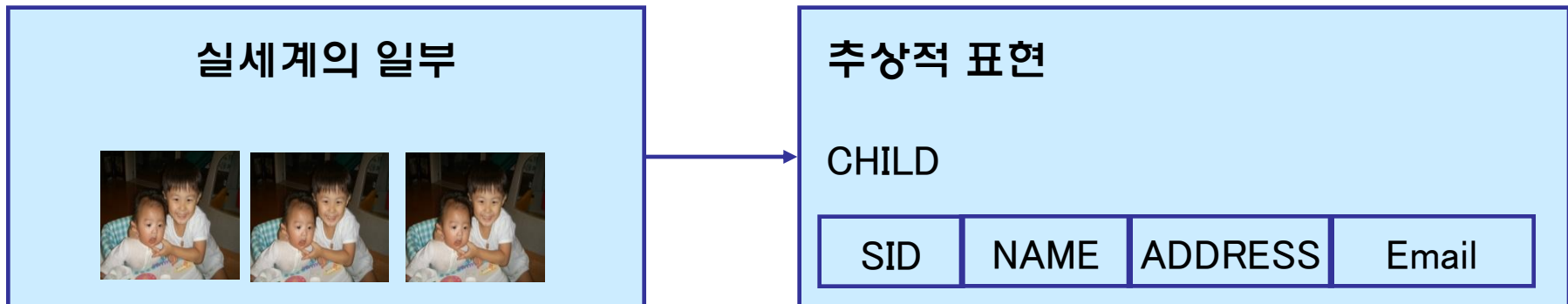
4. 데이터 모델

• 데이터 모델

- 데이터베이스의 구조를 기술하는데 사용되는 개념들의 집합인 구조(데이터 타입과 관계), 이 구조 위에서 동작하는 연산자들, 무결성 제약조건들로 이루어짐
- 사용자에게 내부 저장 방식의 세세한 사항은 숨기면서 데이터에 대한 직관적인 뷰를 제공하는 동시에 이들 간의 사상을 제공

데이터 모델 예

관계 데이터 모델에서는 그림과 같이 실세계 객체를 테이블 형태로 표현한다.



[그림] 관계 데이터 모델에서의 실세계 표현