

데이터베이스 기초

Chapter 2

데이터베이스 시스템

오 세 종

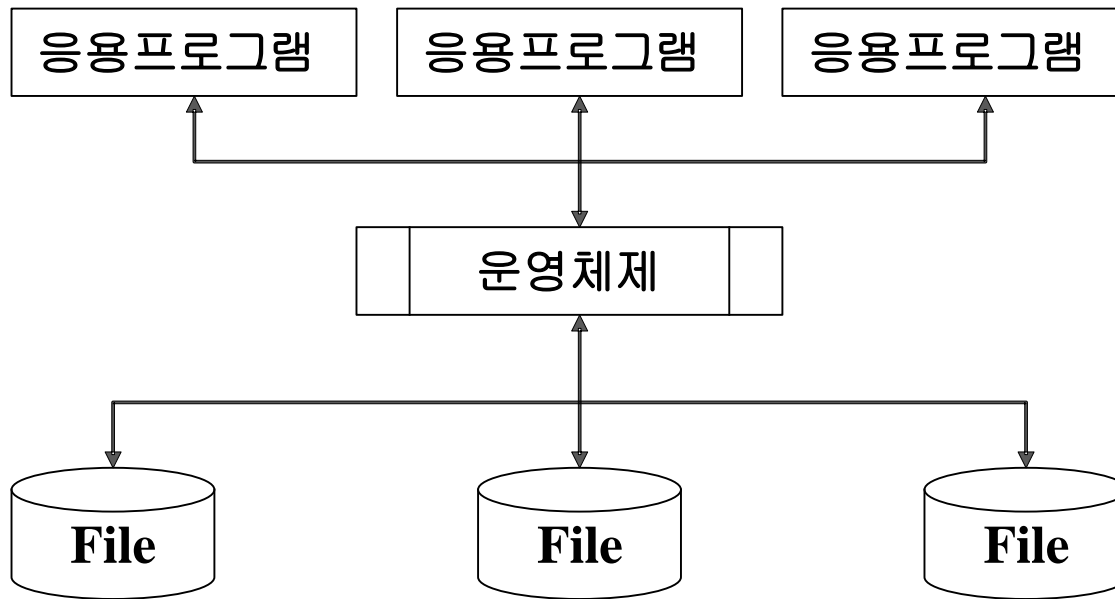
 DANKOOK UNIVERSITY

2.1 데이터베이스의 역사

- 파일 시스템의 위기
 - 컴퓨터발전의 역사 = 데이터처리의 발전사
 - EDPS (Electronic Data Processing System)
 - 컴퓨터와 데이터처리의 밀접한 관계를 보여줌
 - 제1세대컴퓨터시스템
 - 소프트웨어나 저장장치등의 개발이 부족
 - 주로 기술분야의 계산, 자료 분류 등에 사용
 - 제2세대컴퓨터시스템
 - 운영체제가 도입되고 FORTRAN, COBOL 등의 고급 언어 개발
 - 파일시스템(file system)의 도입
 - 자료를 분석하고 처리하는 일에 본격적으로 사용되기 시작

2.1 데이터베이스의 역사

- 파일 시스템의 위기



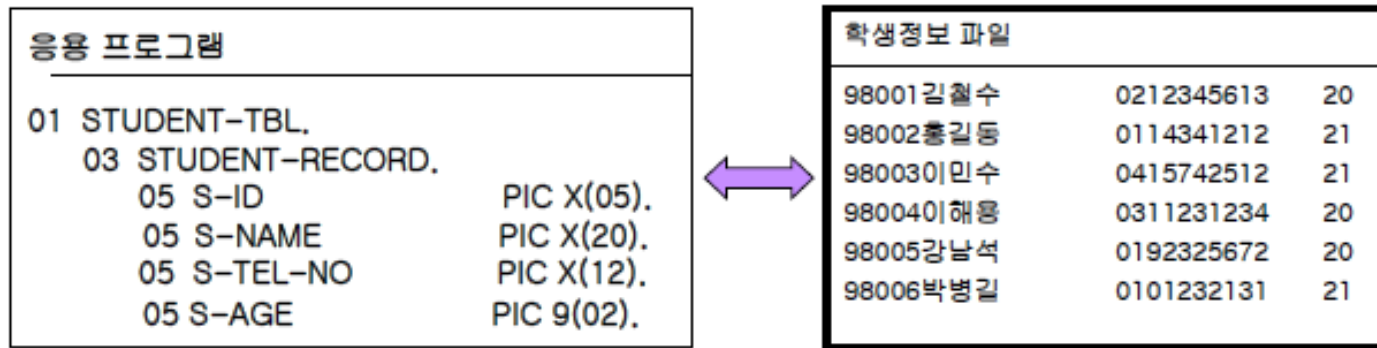
<그림 2.1> 파일 시스템에 기초한 자료처리 모델

2.1 데이터베이스의 역사

- 파일 시스템의 위기
 - 파일 시스템의 문제점
 - 데이터 종속성 (data dependency)
 - 데이터를 사용하는 프로그램의 구조가 데이터 구조 (파일 구조)의 영향을 받는다는 것을 의미
 - 데이터 종속성은 프로그램의 개발과 유지 보수를 어렵게 한다

2.1 데이터베이스의 역사

- 파일 시스템의 위기



<그림 2.2 2.2> 학생정보를 읽어 처리하는 COBOL 프로그램

- 학생의 이름을 저장하는 필드의 길이를 현재 20자리에서 30자리로 늘려야 하는 경우
- 학생정보 파일에서 필드의 순서를 바꾸어 저장하고자 하는 경우

2.1 데이터베이스의 역사

- 파일 시스템의 위기

- 데이터 무결성(data integrity)의 침해

- 데이터 무결성이란 저장된 데이터의 내용이 본래 의도했던 데이터의 형식, 범위를 준수해야 한다는 성질
 - <그림 2.2>의 학생정보 파일에서
 - 나이(AGE) 필드는 숫자 형식이어야 하고 음수가 아닌 양수이어야 한다
 - 나이의 범위는 20~60 사이
 - 과거의 정보시스템에서는 데이터 무결성을 지켜야 할 책임이 프로그래머에게 있음
 - 정보화 사회에서는 인간의 삶이 많은 부분 컴퓨터 시스템에 저장된 데이터에 의존하고 있기 때문에 데이터 무결성의 침해는 매우 심각한 문제



<http://www.infogix.com/blog/data-integrity-the-key-to-recovery-for-healthcare-payer-organizations>

2.1 데이터베이스의 역사

- 파일 시스템의 위기

- 데이터 중복성(data redundancy)

- 같은 내용의 데이터가 여러 곳에 중복하여 저장되는 것을 의미
 - 과거의 정보시스템에서는 개별 부서나 응용 프로그램에서 필요로 하는 데이터 파일을 각각 만들어 사용하는 일이 많았음
 - 저장 공간의 낭비 문제 발생
 - 데이터의 불일치, 보안의 어려움과 같은 문제들이 발생



<http://www.infogix.com/blog/data-integrity-the-key-to-recovery-for-healthcare-payer-organizations>

2.1 데이터베이스의 역사

- 파일 시스템의 위기

- 데이터 불일치(data inconsistency)

- 중복 저장된 데이터들이 서로 일치하지 않는 것을 의미
 - 예1) 우리가 이사를 하게 되면 변경된 주소를 학교나 직장, 은행, 가입된 웹사이트 등에 통보를 해야함
 - 주소 정보가 여러 기관에 중복 저장되어 있다는데서 기인
 - 예2) 어떤 학생이 교무과에 휴학신청을 하여 휴학을 했는데, 그 사실을 모르는 재무과에서는 등록금 고지서를 그 학생에게 발송
 - 교무과의 학생 정보와 재무과의 학생 정보가 각각 관리되면서 불일치



<http://www.infogix.com/blog/data-integrity-the-key-to-recovery-for-healthcare-payer-organizations>

2.1 데이터베이스의 역사

- 파일 시스템의 위기
 - 데이터 표준화(data standardization)의 어려움
 - 일정 규모 이상의 정보 시스템을 개발하기 위해서는 많은 수의 개발자들이 협력 작업 필요
 - 개발자A는 응용 프로그램에서 학생이름을 'S-NAME' 으로, 길이는 20자리로 사용
 - 개발자B는학생이름을'SNME' 으로, 길이는15 자리로사용
 - 표준화가 되어 있지않으면 제 3자가 프로그램을 이해하기도 어렵고 두 응용 프로그램간의 호환성에도 문제
 - 학생 이름을 지칭하고 표현하는 표준화된 규칙이 있다 하더라도 응용 프로그래머가 이를 지키지 않을 수 있음

2.1 데이터베이스의 역사

- 파일 시스템의 위기

- 데이터 보안성(data security)의 결여

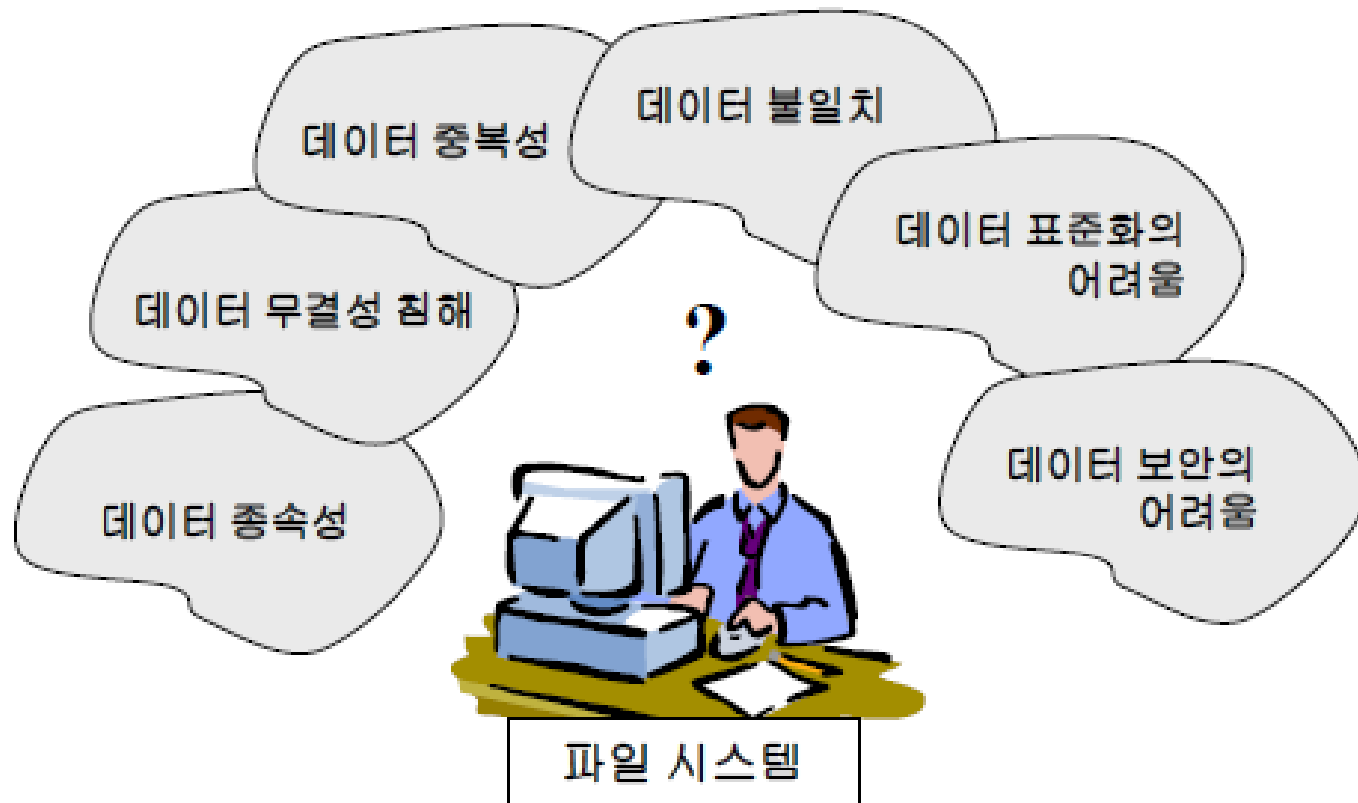
- 데이터가 저장되어 있는 파일은 그 내용이 Text 형식이나 잘 알려진 형식으로 저장되기 때문에 응용프로그램 없이도 쉽게 파일을 열어 내용을 볼 수가 있음
 - 파일의 공유를 위해 접근이 쉬운 위치에 파일을 저장했기 때문에 보안을 유지하기가 어려움
 - 현대의 정보시스템에는 기업의 영업 비밀이나 고객의 사생활 정보와 같은 보안을 필요로 하는 데이터가 많이 저장
 - 보안성의 결여는 심각한 문제임



<http://informationoutsourcing.com/capabilities/data-security>

2.1 데이터베이스의 역사

- 파일 시스템의 위기

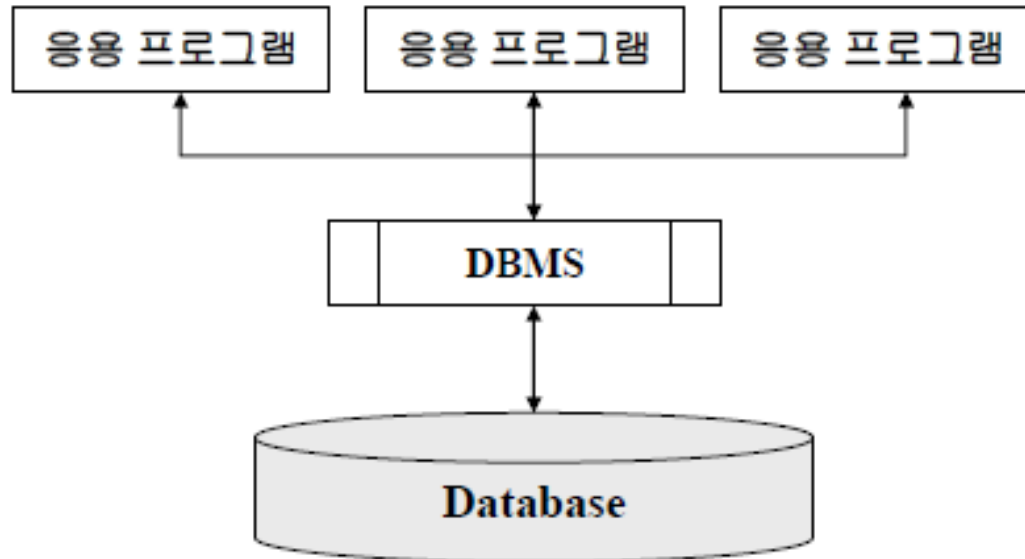


2.1 데이터베이스의 역사

- 데이터베이스의 등장
 - 파일 시스템의 단점을 극복하면서도 다수의 사용자들이 정보를 공유할 수 있어야 한다는 요구에 따라 제안됨
 - 데이터베이스의 철학
 - 파일 형태로 여기저기에 흩어져 있는 데이터, 정보들을 하나로 모아 관리 모아놓은 데이터들을 관리하고
 - 사용자(응용 프로그램)와 데이터 사이에 인터페이스 역할을 할 수 있는 S/W를 제공
 - 초기에는 계층형(hierarchical) 데이터베이스, 네트워크형(network) 데이터베이스가 사용되었으나 현재는 관계형(relational) 데이터베이스가 주류를 이루고 있음

2.1 데이터베이스의 역사

- 데이터베이스의 등장



<그림 2.4> 데이터베이스 시스템의 개요

- 모아놓은 데이터의 집합 : 데이터베이스(database)
- 데이터를 관리하는 S/W : 데이터베이스 관리 시스템 (DBMS; Database Management System)



2.1 데이터베이스의 역사

- 데이터베이스의 등장
 - 데이터베이스의 특징
 - 데이터 독립성(independency) 지원
 - 사용자 혹은 응용 프로그램이 직접 데이터베이스에 접근할 수 없고 반드시 DBMS를 통해서만 접근 가능
 - DBMS는 데이터베이스 내에 있는 데이터의 물리적, 논리적 변화가 응용 시스템에 영향을 미치지 않도록 함으로써 데이터 독립성을 보장
 - 데이터 무결성 유지
 - DBMS가 무결성을 위반하는 데이터가 들어올 경우 처리를 거절함으로써 데이터의 무결성을 지원

2.1 데이터베이스의 역사

- 데이터베이스의 등장
 - 데이터베이스의 특징
 - 데이터 중복성 및 불일치 최소화
 - 데이터베이스 내의 데이터는 한 개인의 관점이나 특정 부서의 관점에서 관리되는 것이 아니라 데이터베이스를 공유하는 조직 전체의 관점에서 관리
 - 동일 데이터가 여러 부서에서 사용하는 경우 이를 하나로 관리함으로써 중복성을 방지
 - 데이터 표준화의 용이성
 - 데이터베이스 관리자(DBA)가 설계과정을 주도함으로써 부서간 이해를 조정하고 관리될 데이터를 표준화시킴
 - 응용 프로그램에서 데이터에 접근하기 위해서는 DBMS가 가지고 있는 구조 정보에 따라야 하기 때문에 자연스럽게 표준화됨

2.1 데이터베이스의 역사

- 데이터베이스의 등장

- 데이터베이스의 특징

- 높은 데이터 보안성

- DBMS는 사용자의 권한에 따라 데이터베이스 내에 있는 데이터에 대한 접근을 제한할 수 있음
 - 저장된 데이터베이스는 일반적으로 DBMS를 통하지 않고는 외부에서 내용을 알아내기 매우 어려움

- 데이터 공유(data sharing)의 용이성

- 데이터베이스 시스템의 기본 철학이 데이터를 통합 관리하고 이를 여러 부서, 사용자들이 공유하도록 하는 것
 - DBMS는 여러 사용자의 요구를 동시에 처리할 수 있는 능력을 가지고 있음
 - DBMS는 데이터를 쉽게 이용할 수 있는 수단을 제공함

2.1 데이터베이스의 역사

- 데이터베이스의 등장



2.1 데이터베이스의 역사

- 관계형 데이터베이스(relational database) 모델
 - 현재 가장 많이 사용되는 데이터베이스 모델
 - 데이터가 테이블 형태로 표현되며, 사용자가 데이터를 쉽게 다룰 수 있도록 해주는 질의어(SQL)를 제공
 - 테이블 형태로 표현된 데이터는 단순해서 누구나 쉽게 이해할 수 있음
 - SQL은 자연어에 가까운 문법을 가지고 있어서 배우기 쉽고, 데이터를 어떻게(how) 가져올 것인가 대신에 어떤(what) 데이터를 원하는지만 기술해주면 되기 때문에 사용자나 개발자의 입장에서는 데이터를 다루는 작업이 매우 단순해짐
 - SQL 명령어나 문법은 표준화 되어 있기 때문에 대부분의 명령어는 모든 관계형 데이터베이스 제품에서 공통적으로 사용 가능

2.1 데이터베이스의 역사

- 관계형 데이터베이스 모델

EMPLOYER

empno	ename	dept	tel	salary
100	김기훈	영업	1241	200
101	홍성범	기획	5621	200
102	이만수	영업	5251	250
103	강나미	생산	1231	300

“영업부에 속한 모든 사원의 이름
과 전화 번호를 보이시오”

⇒

```
SELECT ename, tel  
FROM   employer  
WHERE  dept = '영업'
```

2.1 데이터베이스의 역사

- 관계형 DBMS 제품들

ORACLE®

 Microsoft®
SQL Server®

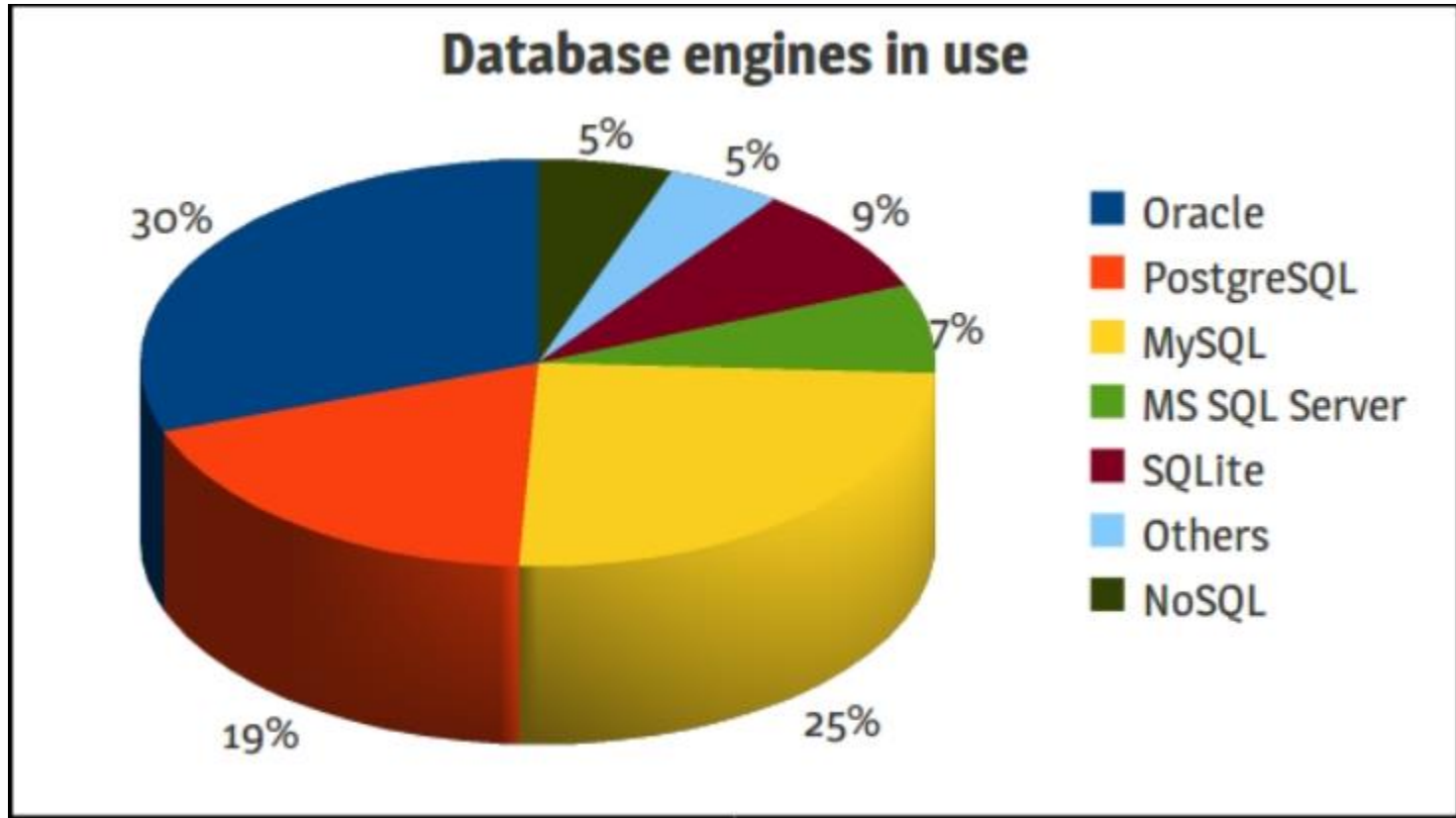
 SYBASE

 MySQL®

 PostgreSQL

2.1 데이터베이스의 역사

- 관계형 DBMS 제품들



<http://www.vertabelo.com/blog/vertabelo-news/jdd-2013-what-we-found-out-about-databases>

2.1 데이터베이스의 역사

- 비 관계형 데이터베이스의 등장
 - 관계형 데이터베이스는 테이블 형태의 정형화된 데이터의 저장 및 관리에 용이
 - SNS, 문자 메시지, 이미지 데이터 등 비정형 데이터의 증대로 기존 관계형 데이터베이스로는 처리가 어려운 데이터들이 증가
 - noSQL 등장 : "Not Only SQL"



2.2 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.

2.2 데이터베이스 시스템

- 데이터베이스의 예

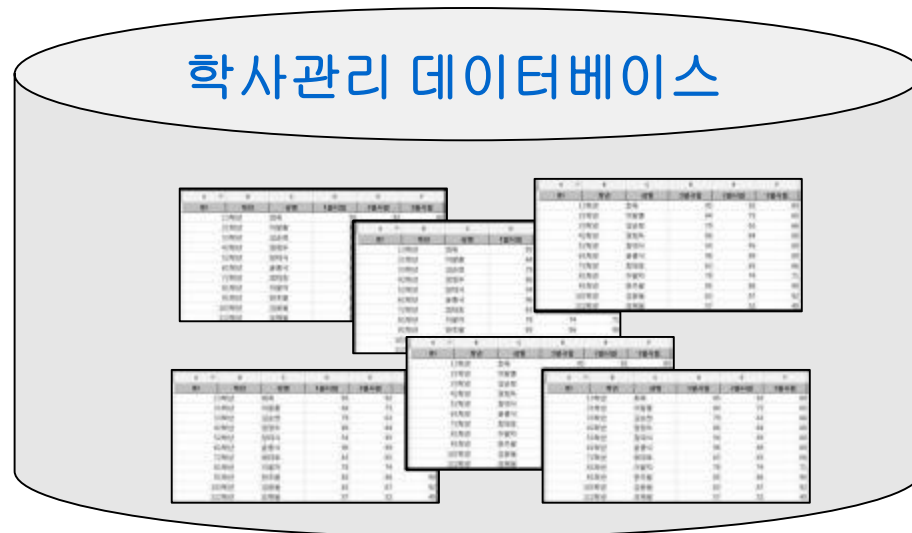
대학에서는 데이터베이스에 학생들에 관하여 신상 정보, 수강 과목, 성적 등을 기록하고, 각 학과에 개설되어 있는 과목들에 관한 정보를 유지하고, 교수에 관해서 신상 정보, 담당 과목, 급여 정보를 유지한다.

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

2.2 데이터베이스 시스템

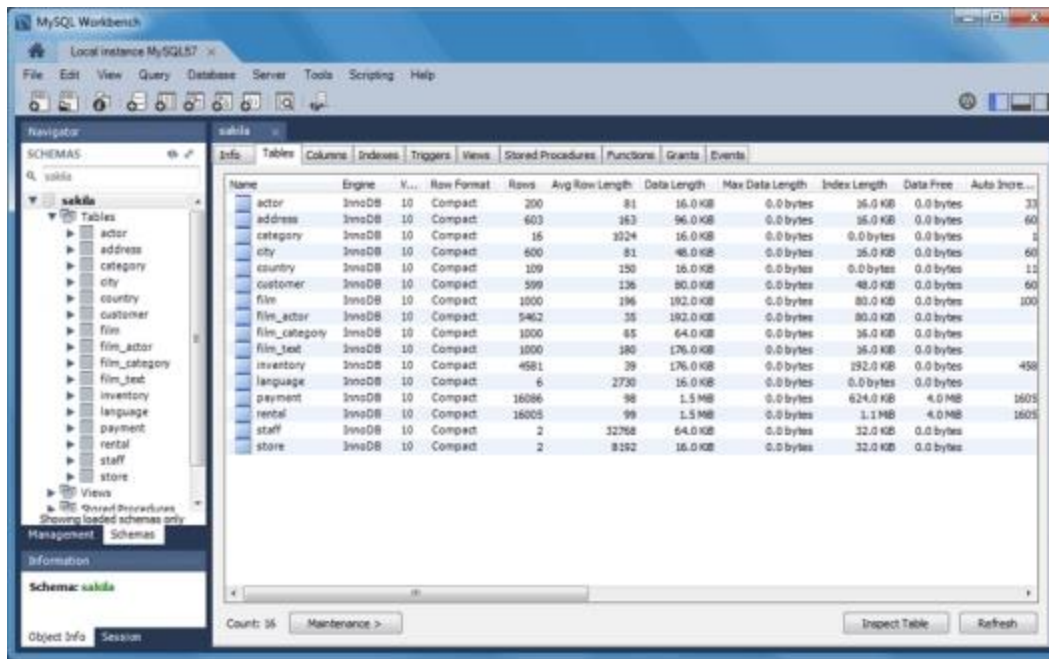
- 데이터베이스의 특징

- 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



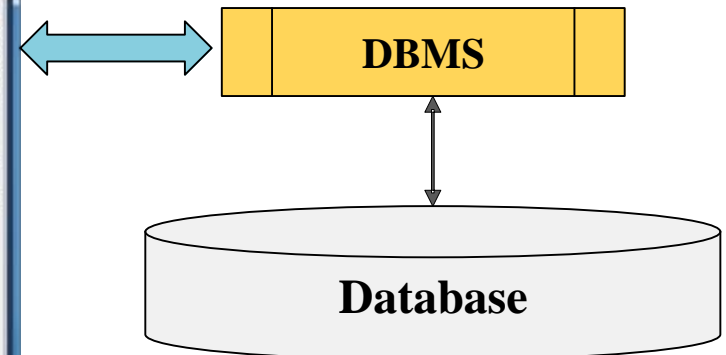
2.2 데이터베이스 시스템

- 데이터베이스 관리 시스템(DBMS: Database Management System)
 - 데이터베이스를 정의하고, 질의어를 지원하고, 리포트를 생성하는 등의 작업을 수행하는 소프트웨어



The screenshot shows the MySQL Workbench interface with the 'sakila' database selected. The 'Tables' tab is active, displaying a list of tables and their properties. The tables listed are: actor, address, category, city, country, customer, film, film_actor, film_category, film_text, inventory, language, payment, rental, staff, and store. The table properties include Engine, Rows, Avg Row Length, Data Length, Max Data Length, Index Length, Data Free, and Auto Increment.

Name	Engine	Rows	Avg Row Length	Data Length	Max Data Length	Index Length	Data Free	Auto Incr...
actor	InnoDB	200	81	16.0 KB	0.0 bytes	16.0 KB	0.0 bytes	33
address	InnoDB	603	163	96.0 KB	0.0 bytes	16.0 KB	0.0 bytes	60
category	InnoDB	16	1024	16.0 KB	0.0 bytes	0.0 bytes	0.0 bytes	1
city	InnoDB	600	81	48.0 KB	0.0 bytes	16.0 KB	0.0 bytes	60
country	InnoDB	109	150	16.0 KB	0.0 bytes	0.0 bytes	0.0 bytes	11
customer	InnoDB	599	136	80.0 KB	0.0 bytes	48.0 KB	0.0 bytes	60
film	InnoDB	1000	196	192.0 KB	0.0 bytes	80.0 KB	0.0 bytes	100
film_actor	InnoDB	5462	35	192.0 KB	0.0 bytes	80.0 KB	0.0 bytes	100
film_category	InnoDB	1000	65	64.0 KB	0.0 bytes	16.0 KB	0.0 bytes	100
film_text	InnoDB	1000	180	176.0 KB	0.0 bytes	16.0 KB	0.0 bytes	100
inventory	InnoDB	4581	39	176.0 KB	0.0 bytes	192.0 KB	0.0 bytes	458
language	InnoDB	6	2730	16.0 KB	0.0 bytes	0.0 bytes	0.0 bytes	100
payment	InnoDB	16086	98	1.5 MB	0.0 bytes	624.0 KB	4.0 MB	1608
rental	InnoDB	16005	99	1.5 MB	0.0 bytes	1.1 MB	4.0 MB	1600
staff	InnoDB	2	32768	64.0 KB	0.0 bytes	32.0 KB	0.0 bytes	100
store	InnoDB	2	8192	16.0 KB	0.0 bytes	32.0 KB	0.0 bytes	100



2.2 데이터베이스 시스템

- 데이터베이스 관리 시스템 (계속)
 - 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

2.2 데이터베이스 시스템

- 데이터베이스 스키마(schema)
 - 전체적인 **데이터베이스 구조**를 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

2.2 데이터베이스 시스템

- 데이터베이스 스키마

DEPARTMENT(DEPTNO, DEPTNAME, FLOOR)
EMPLOYEE(EMPNO, EMPNAME, TITLE, DNO, SALARY)

- 데이터베이스 상태

DEPARTMENT

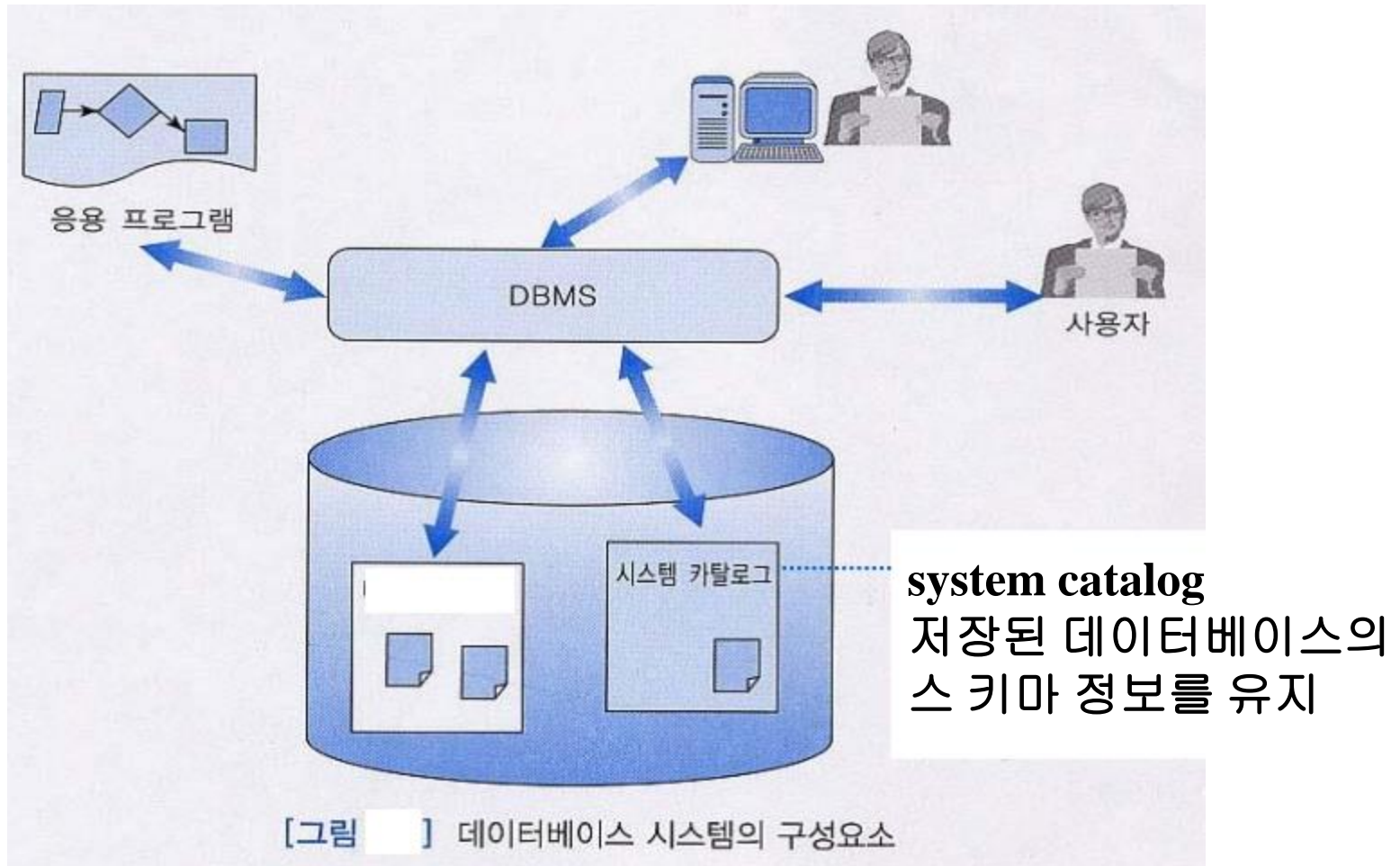
DEPTNO	DEPTNAME	FLOOR
1	영업	8
2	기획	10
3	개발	9

EMPLOYEE

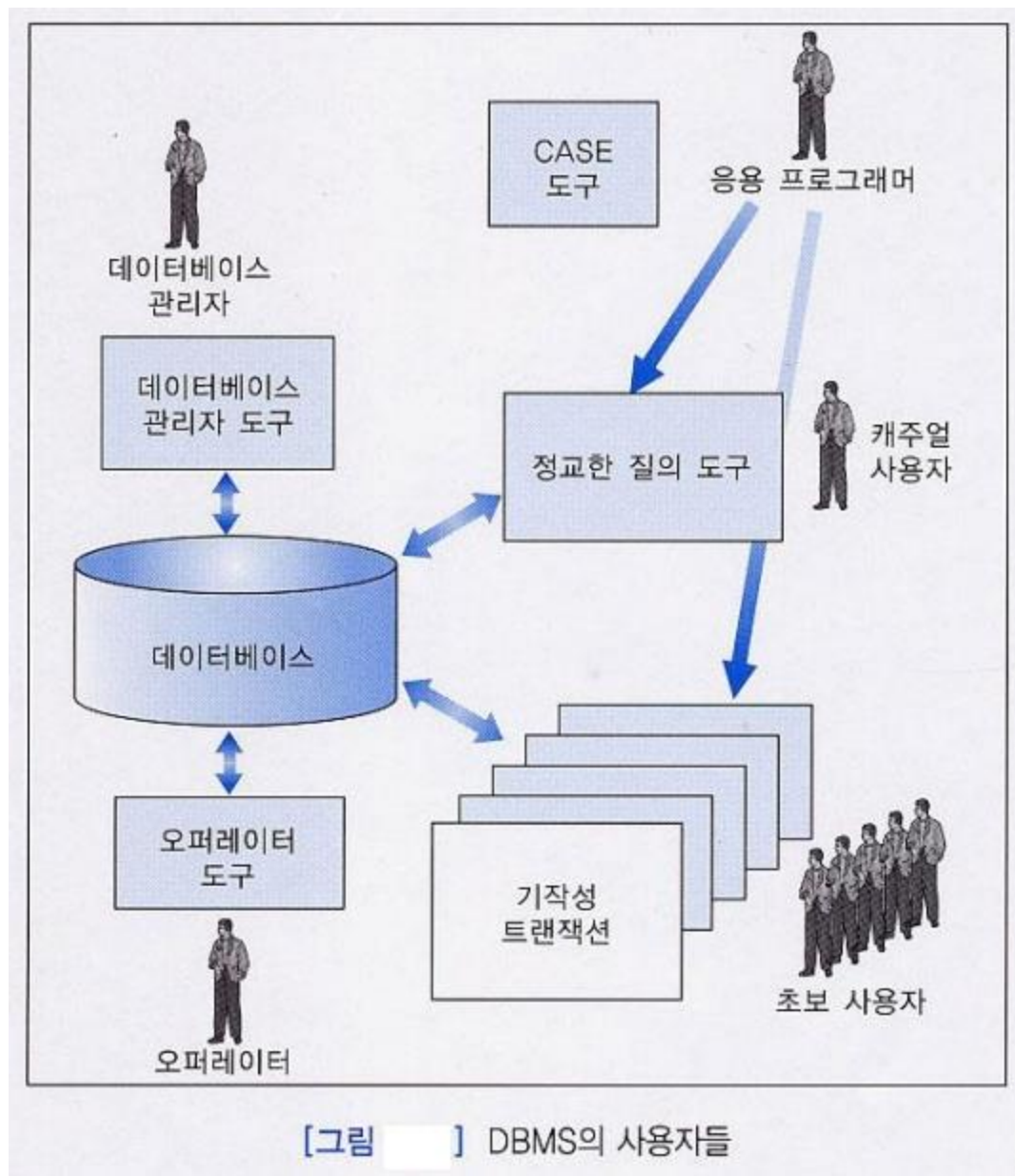
EMPNO	EMPNAME	TITLE	DNO	SALARY
2106	김창섭	대리	2	2000000
3426	박영권	과장	3	2500000
3011	이수민	부장	1	3000000
1003	조민희	대리	1	2000000
3427	최종철	사원	3	1500000

2.2 데이터베이스 시스템

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

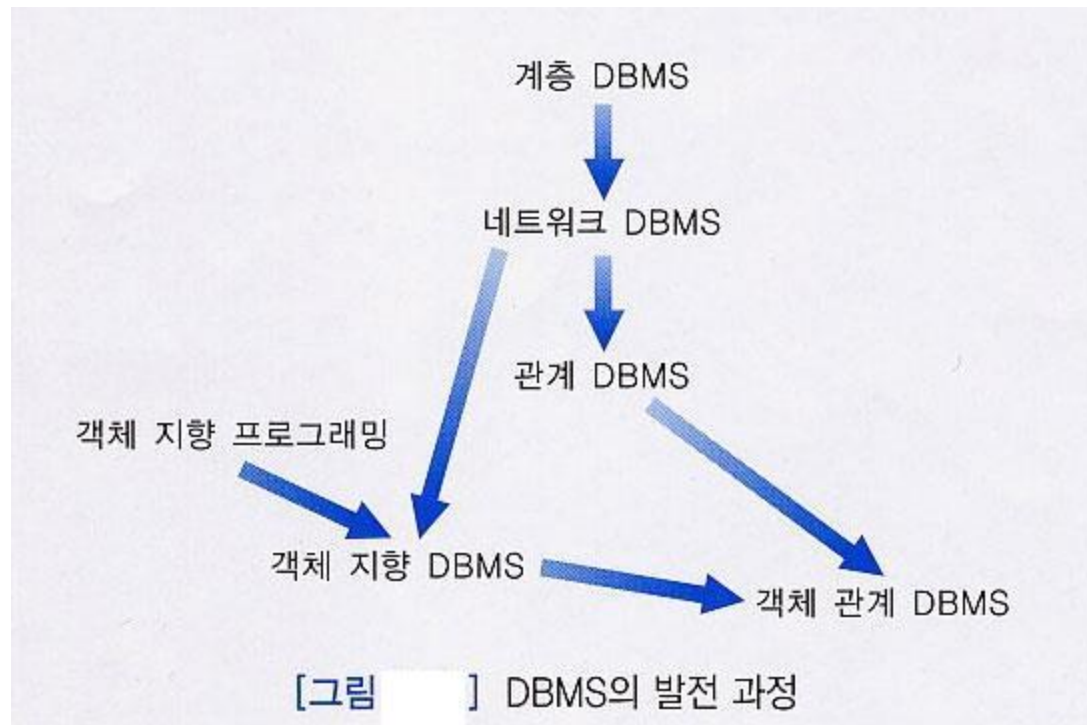


2.2 데이터베이스 시스템



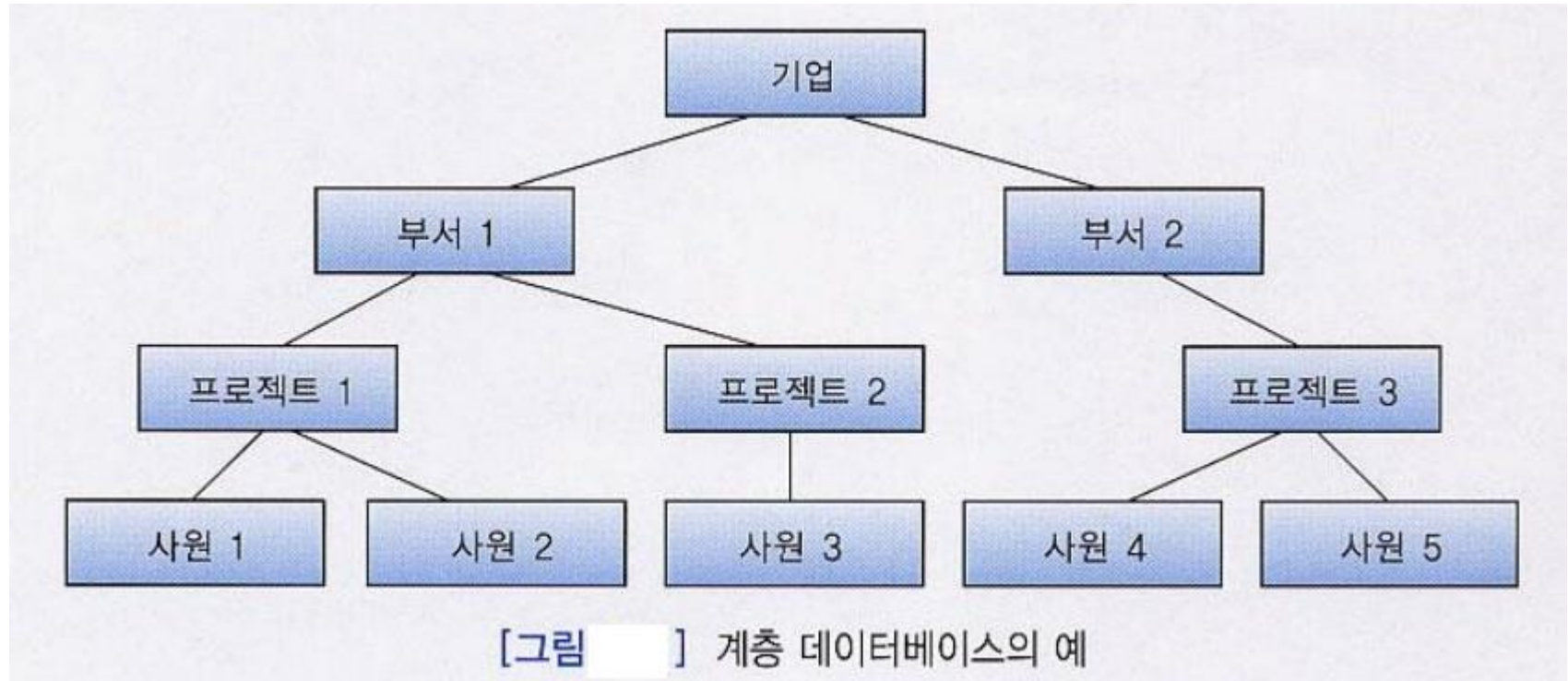
2.3 관계형 데이터베이스

- 데이터 모델(data model)
 - 데이터베이스의 구조를 기술하는데 사용되는 개념들의 집합인 구조(데이터 타입과 관계), 이 구조 위에서 동작하는 연산자들, 무결성 제약 조건들로 이루어짐
 - 사용자들에게 데이터가 어떤 모양으로 저장되어 있다고 보이게 할 것인가?



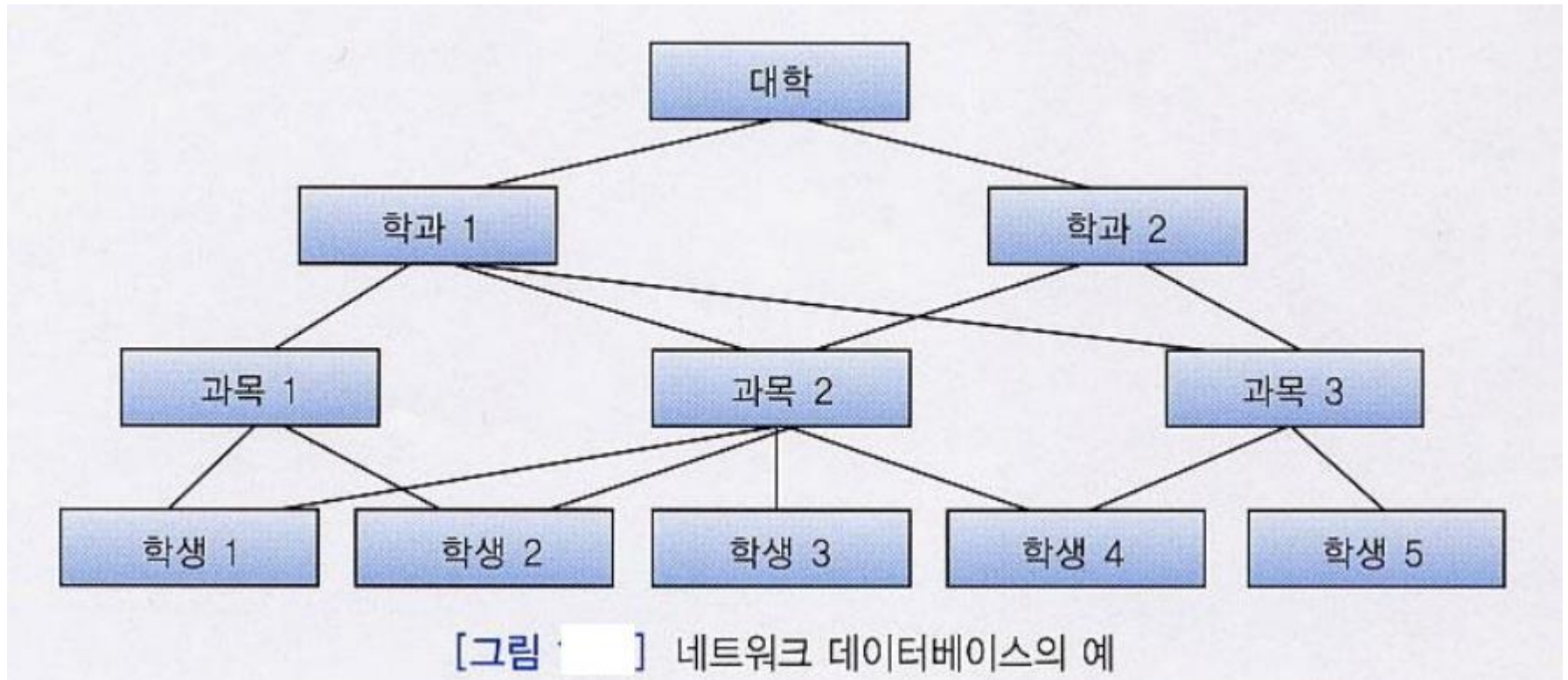
2.3 관계형 데이터베이스

- 계층형 (hierarchical) 데이터 모델



2.3 관계형 데이터베이스

- 네트워크(network) 데이터 모델



2.3 관계형 데이터베이스

- 관계형(relational) 데이터 모델
 - 사용자에게 데이터가 테이블 형태로 관리된다고 보이게 함

STUDENT

NAME	ST_NO	ADDRESS	DEPT	GRADE
송치윤	52015	사당동	컴퓨터	3.3
김구완	53116	홍제동	정보통신	3.1
최재석	56034	양재동	정보관리	3.5
송혜영	52042	신정동	컴퓨터	2.9
조미림	53108	역삼동	정보통신	3.4

[그림] STUDENT 테이블

2.3 관계형 데이터베이스

- 관계형(relational) 데이터 모델
 - 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 → SQL 언어

School Table			
ID	Name		
S001	University of Technology		
S002	University of Applied Science		

Student Table			
School ID	ID	Name	DOB
S001	UT-1000	Tommy	05/06/1995
S001	UT-1000	Better	16/04/1995
S002	UAS-1000	Linda	02/09/1995
S002	UAS-1000	Jonathan	22/06/1995

2.4 DBMS 언어

- 데이터베이스에 접근하기 위해서는 DBMS 와 커뮤니케이션이 필요하다. 사용자(사람, 프로그램)와 DBMS가 통신하기 위한 언어를 SQL (Structured Query Language) 라고 한다
- SQL은 대표적인 비절차적(non-procedural language) 언어
- SQL 명령어
 - 데이터 정의어 : CREATE, DROP
 - 데이터 조작어 : INSERT, UPDATE, DELETE
 - 데이터 질의어 : SELECT

2.4 DBMS 언어

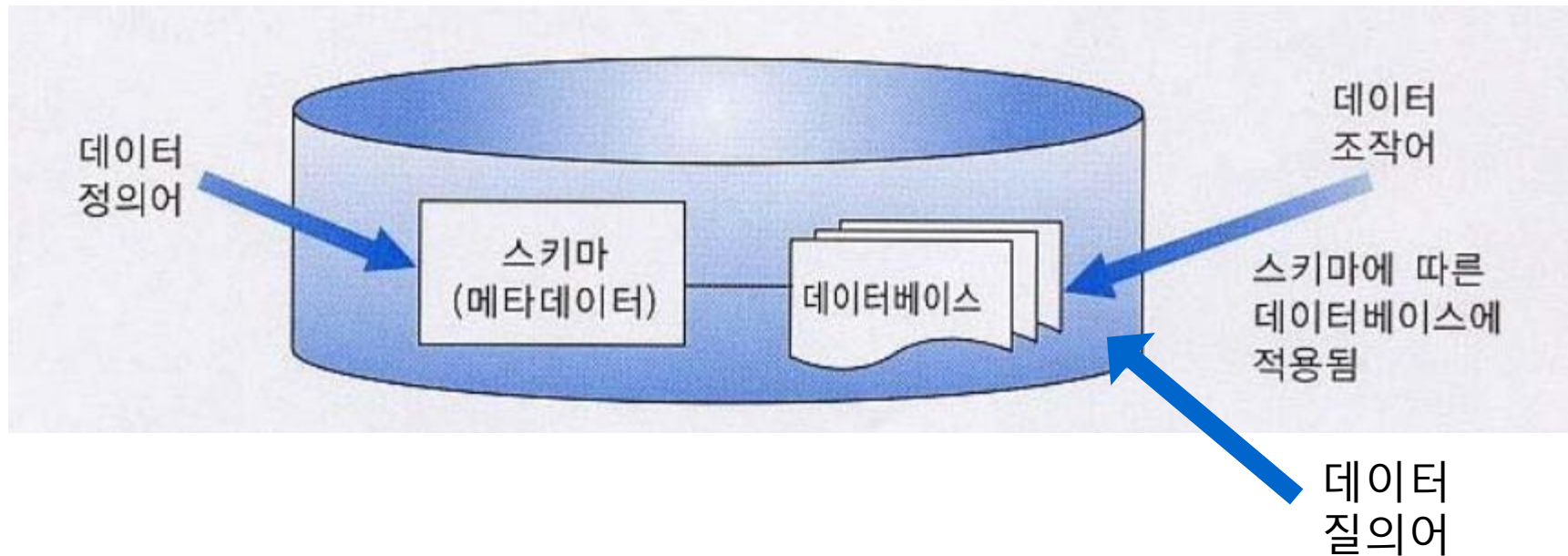
- 데이터 정의어(DDL: Data Definition Language)
 - 데이터베이스 **스키마를 정의**하는데 사용하는 명령어
 - 데이터 정의어로 명시된 문장이 입력되면 DBMS는 사용자가 정의한 스키마에 대한 명세를 **시스템 카탈로그** 또는 **데이터 사전**에 저장
 - 데이터 정의어의 기본적인 기능
 - 데이터 모델에서 지원하는 데이터 구조를 생성
 - (예, SQL에서 CREATE TABLE)
 - 데이터 구조의 변경
 - (예: SQL에서 ALTER TABLE)
 - 데이터 구조의 삭제
 - (예: SQL에서 DROP TABLE)
 - 데이터 접근을 위해 특정 애트리뷰트 위에 인덱스를 정의 (예: SQL에서 CREATE INDEX)

2.4 DBMS 언어

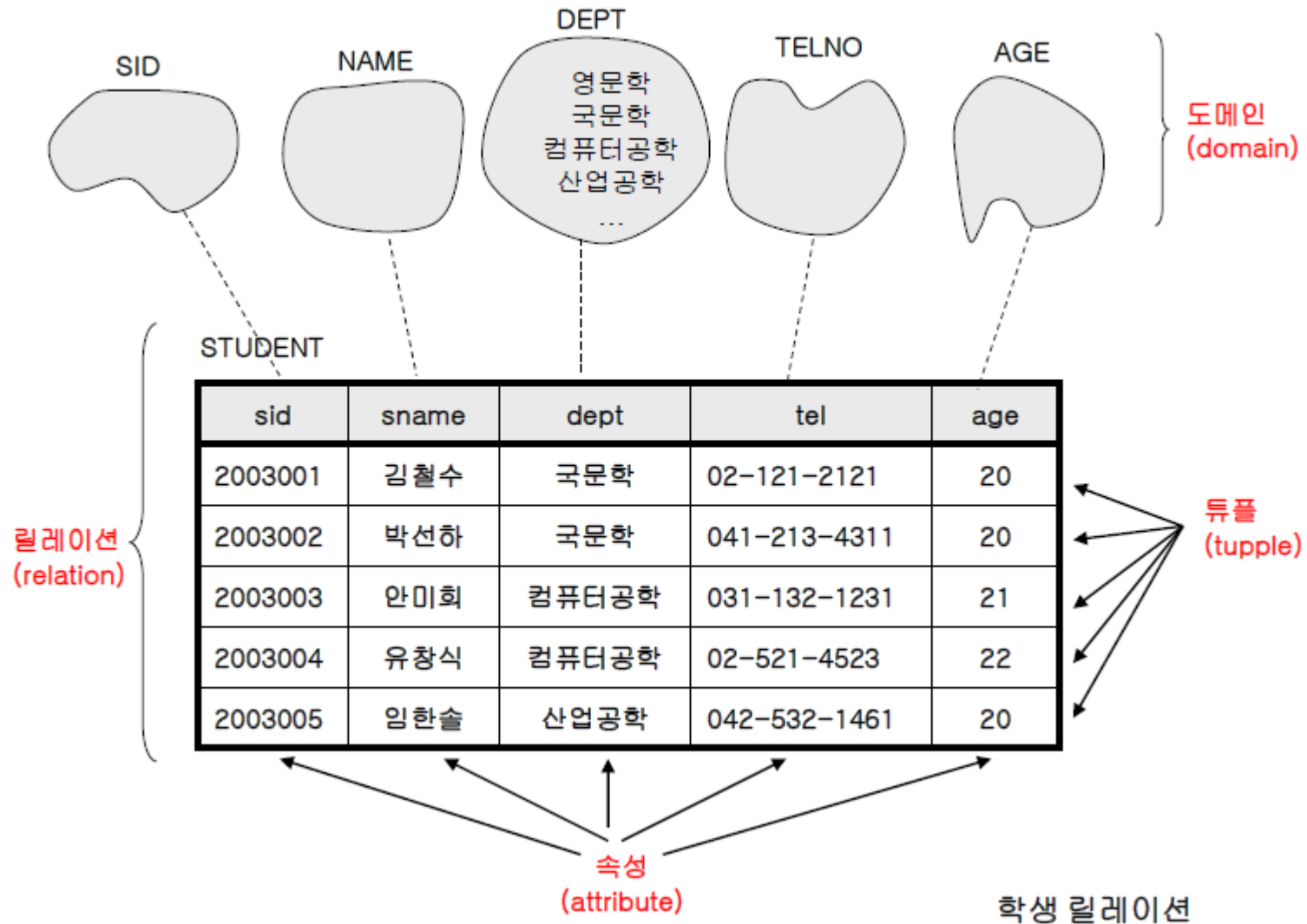
- 데이터 조작어(DML: Data Manipulation Language)
 - 데이터베이스 내의 원하는 데이터를 수정하고, 삽입하고, 삭제하는 명령어
 - INSERT, UPDATE, DELETE
 - 대부분의 데이터 조작어는 SUM, COUNT, AVG와 같은 내장 함수들을 지원함
 - 데이터 조작어/질의어는 단말기에서 대화식으로 입력되어 수행되거나 C, 코 볼 등의 고급 프로그래밍 언어로 작성된 프로그램에 내포되어 사용될 수 있음(embedded SQL)

2.4 DBMS 언어

- 데이터 질의어(QL: Query Language)
 - 데이터베이스 내의 원하는 데이터를 조회하는 명령어
 - SELECT
 - 가장 많이 사용하는 명령어



2.5 관계형 데이터베이스 용어



2.5 관계형 데이터베이스 용어

- E.F.Codd가 정의한 것으로 오늘날 일반적으로 사용하는 용어와는 차이가 있음
- 릴레이션(relation)
 - 테이블이라는 용어로 더 많이 사용.
 - 관계형 데이터베이스에서 정보를 구분하여 저장하는 기본 단위
STUDENT 릴레이션 : 학생에 관한 정보를 저장
 - SUBJECT라는 릴레이션 : 과목에 대한 정보를 저장
 - 동일한 데이터베이스 내에서는 같은 이름을 갖는 릴레이션이 존재할 수 없다

2.5 관계형 데이터베이스 용어

- 속성(attribute)

- 릴레이션에서 관리하는 구체적인 정보 항목에 해당하는 것이 속성 현실세계의 개체(예: 학생, 교수, 과목,...)들은 많은 속성들을 갖는데 그 중에서 관리해야 할 필요가 있는 속성들만을 선택하여 릴레이션에 포함시킴
- 속성 역시 고유한 이름을 갖으며 동일 릴레이션 내에서는 같은 이름의 속성이 존재할 수 없음

- 튜플(tuple)

- 릴레이션이 현실세계의 어떤 개체를 표현한다면 튜플은 그 개체에 속한 구성원들 개개의 정보를 표현
- 예) '학생'은 개체를 나타내는 이름이고 '김철수', '박선하', '안미희',... 등은 '학생' 개체의 구성원
- 한 릴레이션에 포함된 튜플의 개수는 시간에 따라 변할 수 있으며, 한 릴레이션은 적게는 수십 개 많게는 수십만 개의 튜플을 포함할 수 있음

2.5 관계형 데이터베이스 용어

- 도메인(domain)
 - 도메인이란 릴레이션에 포함된 각각의 속성들이 갖을 수 있는 값들의 집합
 - 도메인이라는 개념이 필요한 이유는 릴레이션에 저장되는 데이터 값들이 본래 의도했던 값들만 저장되고 관리되도록 하는데 있음 예)'성별'이라는 속성이 있다면 이 속성이 가질 수 있는 값은 {남,여}.
 - 데이터베이스 설계자는 성별의 도메인으로 'SEX'를 정의하고 그 값으로 {남,여}를 지정한 뒤, '성별'이라는 속성은 'SEX' 도메인에 있는 값만을 갖을 수 있다고 지정해 놓으면 사용자들이 실수로 남,여 이외의 값을 입력하는 것을 DBMS가 막을 수 있음
- 현실적으로 도메인을 구현하는 것은 어렵기 때문에 대부분의 DBMS 제품에서는 사용자 정의 데이터타입으로 사용

2.5 관계형 데이터베이스 용어

- 현재는 여러 용어가 혼용되고 있는 상황임

E.F.Codd 의 용어	File 시스템의 용어	자주 사용되는 용어
릴레이션(relation)	파일(file)	테이블(table)
속성(attribute)	필드(field)	열(column), 컬럼
튜플(tuple)	레코드(record)	행(row)

본 강의에서는 앞으로 테이블, 컬럼, 튜플이라는 용어를 사용하기로 함

기억해야할 용어들

- 데이터베이스(database)
- 관계형 데이터베이스(relational database)
- DBMS(database management system)
- 스키마(schema)
- SQL(structured query language)

- 테이블(table)
- 컬럼(column)
- 튜플(tuple)



수고했습니다.
이제 어려운 고비는 모두 넘었습니다