

u 강의 교재

- 데이터베이스 시스템 개론, 김명호, 이윤준, 유재수 역
- 그린 출판사, 2004년 개정
- **A First Course in Database Systems, Jeffrey D. Ullman and Jennifer Widom, Prentice Hall, 2008 (2nd Ed. or 3rd Ed.)**

1장. 데이터베이스 시스템의 세계

- u 데이터베이스 시스템의 발전
- u **DBMS**의 구조
- u 데이터베이스 시스템의 미래



데이터베이스 시스템

ㄴ 데이터베이스

- 데이터베이스 관리 시스템 (**DBMS:Database Management System**)에 의해 관리되는 데이터의 모임

ㄴ **DBMS**의 기능

- **DB 생성 및 스키마 정의** : 데이터 정의 언어(**data definition language:DDL**)를 사용
- **DB에 대한 질문 및 수정** : 질의어(**Query Language**) 및 데이터 조작 언어(**data manipulation language:DML**) 사용
- 대용량 데이터 저장소 지원 : 장기간 저장 및 효율적 접근 제공
- 영구적 **DB** 저장 : 어떤 오류나 위험도 극복
- **DB에 대한 다수 사용자들의 동시 사용** : 일관성(**consistency**) 보장

데이터베이스 시스템

ㄴ 데이터베이스 관리 시스템(DBMS:DataBase Management System)

: 특정 목적을 위해서 존재하는 데이터베이스를 다수의 사용자가 동시에 안전하고 효율적으로 사용할 수 있도록 하는 S/W들의 집단

- 영구적인 저장(persistent storage)

- » 대용량 데이터의 안전하고 효율적인 영구적 저장 및 관리

- 프로그램 인터페이스

- » 사용자나 프로그램 개발자에게 데이터의 용이한 접근 및 수정 방법 제공

- 트랜잭션(transaction) 관리

- » 다수의 사용자나 프로세스가 동시에 데이터를 접근하도록 허용

- » 고립성(isolation), 원자성(atomicity), 지속성(durability), 일관성(consistency) 등의 성질을 만족

초기 데이터베이스 관리 시스템

ㄴ 1960년대 후반에 최초의 상용 DBMS가 등장

- 화일 시스템으로부터 발전
- 여러 종류의 데이터 모델
 - » 계층 모델, 네트워크 모델
- 고수준 질의 언어를 지원하지 않았음
 - » 데이터 검색을 위해서 매번 프로그램 작성

ㄴ DBMS의 중요한 응용 분야

- 항공 예약 시스템 : 항공편 예약, 항공편에 대한 정보
- 은행 시스템 : 고객 정보, 계좌, 대출, 입출금, 잔액 등의 정보
- 법인 레코드 : 제품 판매, 직원, 고용인에 대한 정보
- 웹 쇼핑 몰 : 제품 검색, 현금 및 신용카드 구매, 주문 확인 등에 대한 정보

관계 데이터베이스 시스템

ㄴ 관계 모델(relational model)

- 1970년 Ted Codd에 의해 제안
- 데이터를 테이블의 형태로 표현
- 질의를 고수준 언어로 표현

SQL

애트리뷰트 →	<i>accountNo</i>	<i>balance</i>	<i>type</i>
튜플 →	12345	1000.00	savings
	67890	2846.92	checking

Accounts 테이블 (또는 릴레이션)

```
SELECT balance
FROM Accounts
WHERE accountNo = 67890;

SELECT accountNo
FROM Accounts
WHERE type = 'savings' and
       balance < 0;
```

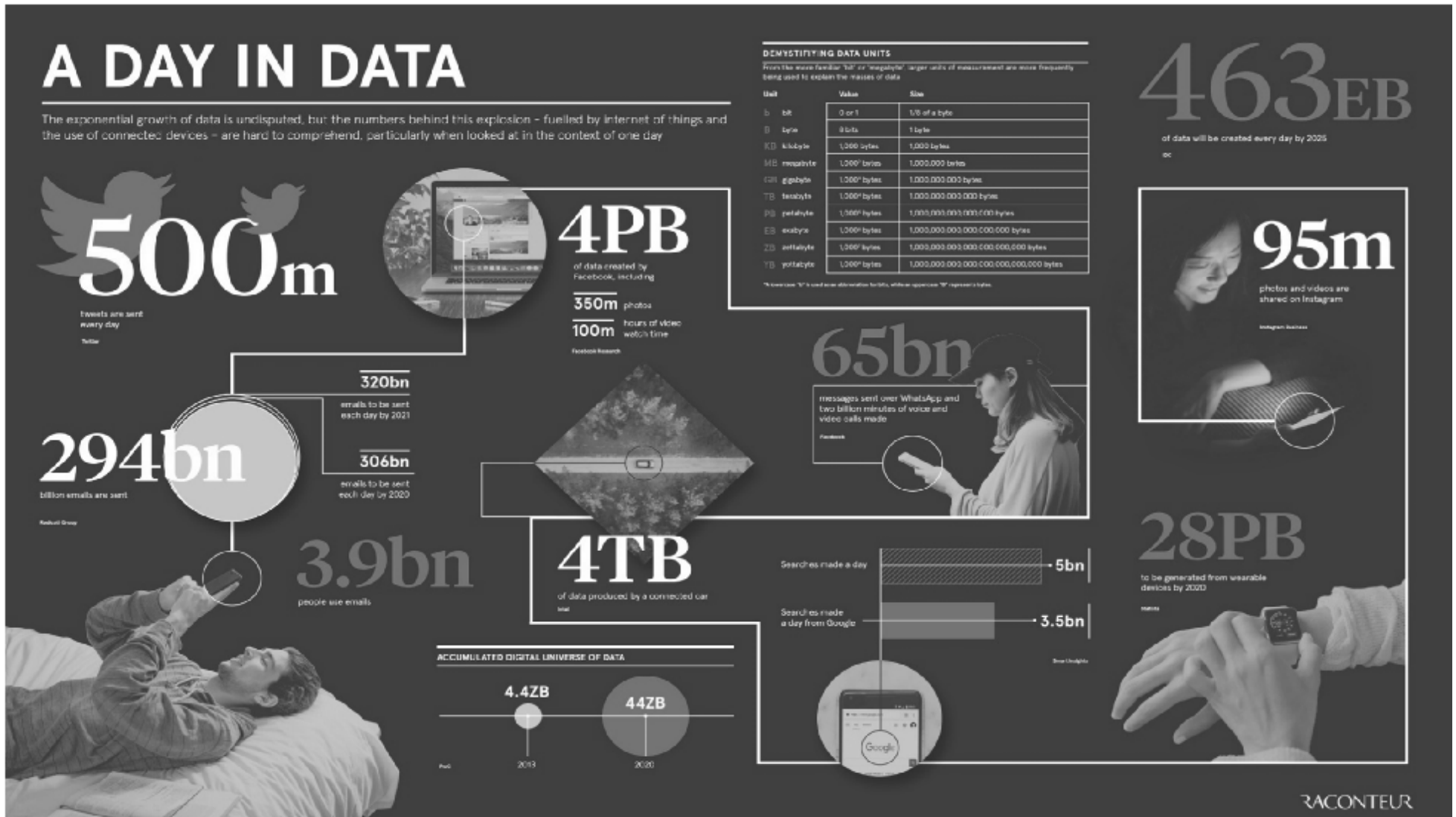
소형화 되어가는 시스템

- u 원래 **DBMS**는 대형 컴퓨터에서 가동되는 거대하고 값비싼 소프트웨어 시스템이었다.
- u 오늘날, 기가바이트단위의 데이터는 하나의 디스크에 저장될 수 있고, DBMS를 PC에서 구동하는 것도 가능하다.
 - **DBMS**가 소형 기계에서도 사용 가능하게 되었음
 - » Linux, Windows 등에서 Oracle, DB2, MS SQL Server 등의 **DBMS** 구동

대형화 되어가는 시스템

- u 이제 기가바이트단위도 대용량이 아니다.
 - 테라바이트 혹은 그 이상의 정보가 요구되어지기도 함
 - » Kilobyte(KB) : $2^{10}(10^3)$, Megabyte(MB) : $2^{20}(10^6)$
 - » Gigabyte(GB) : $2^{30}(10^9)$, Terabyte(TB) : $2^{40}(10^{12})$
 - » Petabyte(PB) : $2^{50}(10^{15})$, Exabyte(EB) : $2^{60}(10^{18})$
 - » Zettabyte(ZB) : $2^{70}(10^{21})$, Yottabyte(YB) : $2^{80}(10^{24})$
 - 이미지, 오디오, 비디오 같은 대용량의 데이터 타입
- u 분당 300시간 분량의 YouTube 영상이 upload 됨

하루에 생산되는 데이터의 양



정보 통합

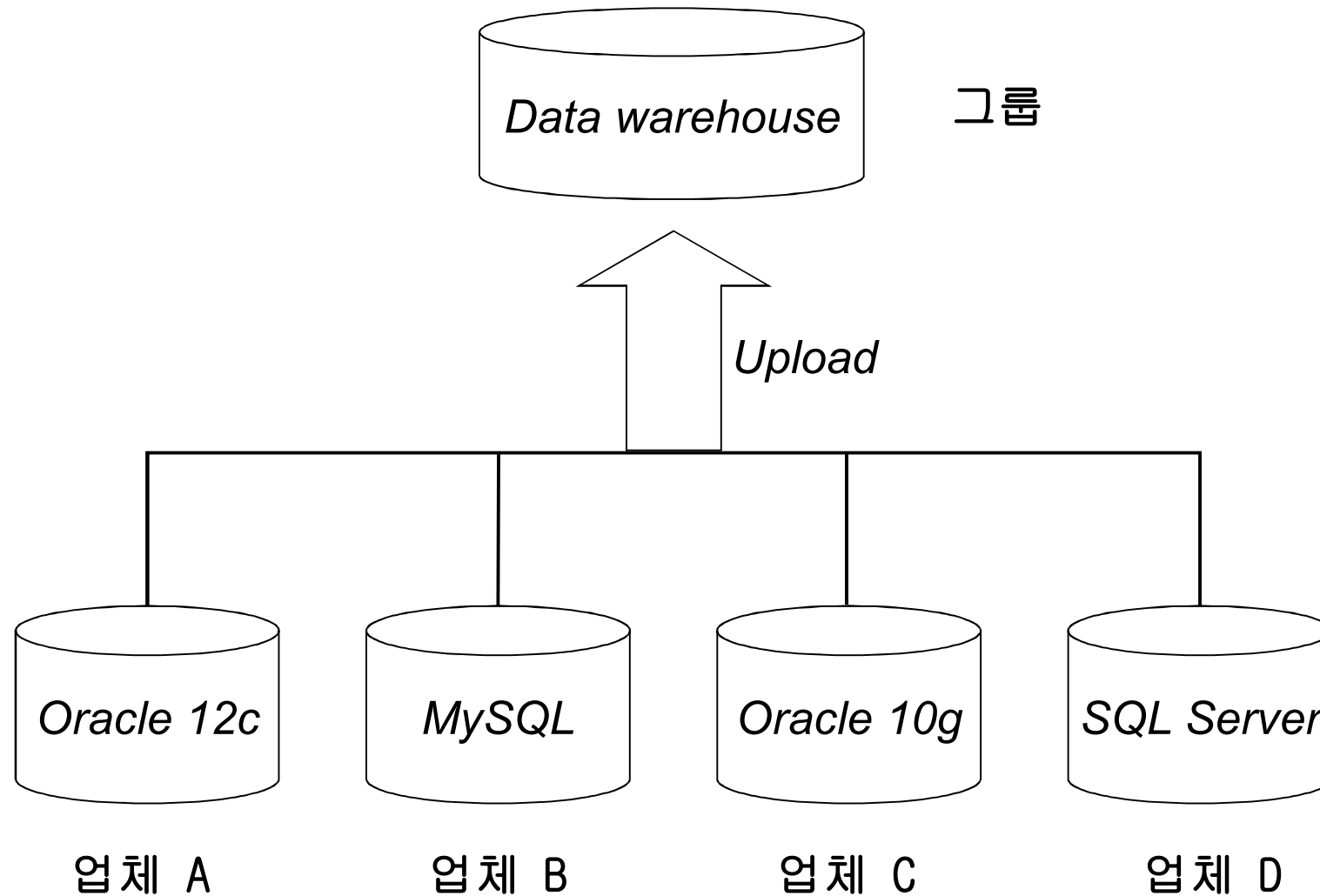
u 데이터 웨어하우스(warehouse)

- 한 기관내에 여러 데이터베이스들이 있을 수 있음
 - » 다른 **DBMS**들, 정보에 대한 다른 구조들
 - » 여러 데이터베이스들에 있는 정보는 적절하게 번역되어 중앙 데이터베이스에 복사되어 짐
- 기획과 분석을 위해 사용되어질 수 있음

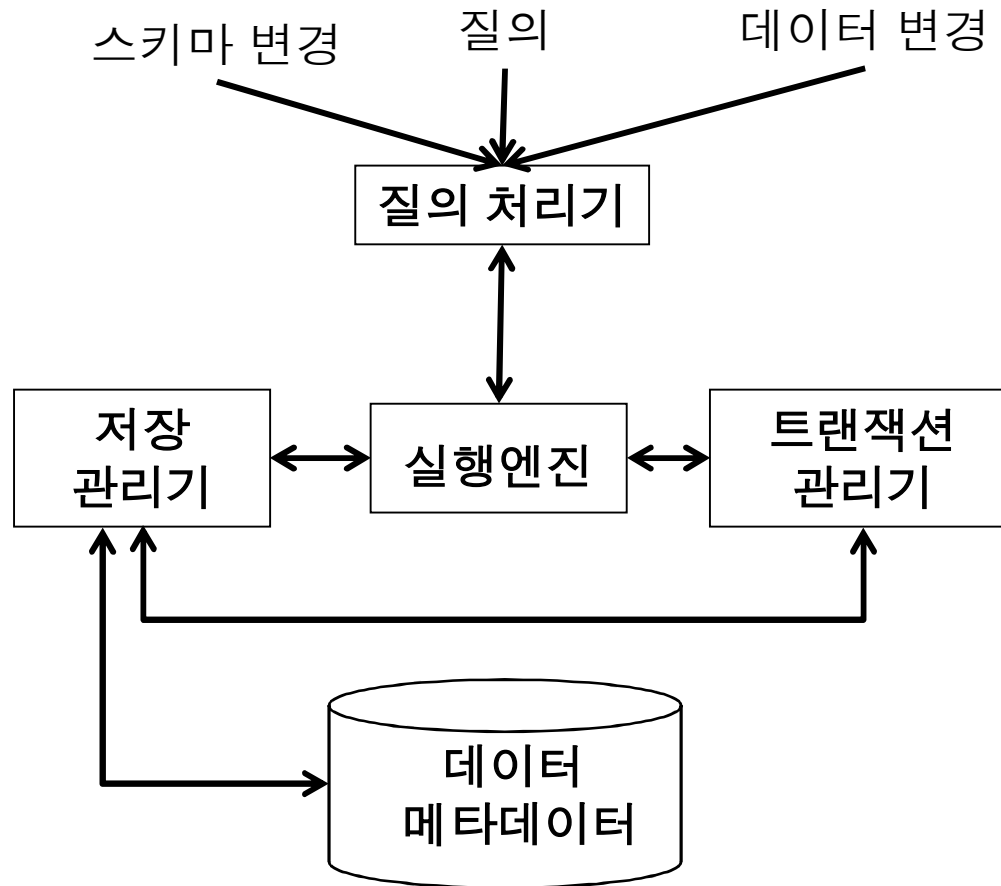
u 데이터 마이닝(mining)

- 대용량 데이터로부터 관심이 있거나 특이한 패턴을 발견
- 기획과 분석을 위해 사용

데이터 웨어하우스의 필요성



DBMS 구성요소의 개요



- ∅ **질의(query):** 데이터에 관한 질문
- ∅ **데이터 변경(modification):** 데이터를 변경하기 위한 연산
- ∅ **스키마(schema) 변경:** 메타데이터를 변경하기 위한 연산
- ∅ **메타데이터(metadata):** 데이터의 구조에 대한 정보

질의 처리기

u 질의 파서(parser)

- 문자열 형태의 질의를 트리 구조로 변환

u 질의 선처리기(preprocessor)

- 질의에 대한 의미 조사 및 초기 질의 계획 작성

u 질의 최적화(optimization)

- 효과적인 질의 계획(즉, 저장 시스템에 대한 요청들)을 선택
 - » Disk I/O의 횟수를 최소화시킴

[예제] Customers(고객이름, 주민등록번호, 주소)와 Account(계좌번호, 잔액, 계좌주의 주민등록번호) 등의 두 개의 테이블에 대해서 “Sally Jones가 소유한 모든 계좌의 잔액을 찾아라” 라는 질의 요청

- » Sequential v.s. Index search의 선택 사항에 따라 성능 좌우

참고: 색인

↳ 색인(index)

- 데이터의 일부로 데이터를 신속하게 찾을 수 있도록 도와 주는 데이터 구조
 - (예) 계좌 번호에 대한 잔액을 빠르게 탐색
- 저장된 데이터의 일부

↳ B-tree

- 이진 탐색 트리의 일반화
 - 많은 fanout, 즉, 많은 자식 노드들
 - » **B-tree**의 높이는 이진 탐색 트리의 높이보다 낮음
 - » 일반적으로 3 또는 4 레벨
 - 2차 저장매체에서 데이터에 대한 가장 일반적인 색인 구조
- F** 해쉬(hash) 테이블도 색인 구조로 사용되기도 함

실행 엔진(execution engine)

- u 질의 처리기에서 생성한 질의 계획의 각 단계별 작업들을 실제 실행
- u **DBMS**의 다른 구성 요소들과 연동
 - 저장 관리기, 트랜잭션 관리기 등

저장 관리기

u 파일 관리기

- 디스크에서 파일의 위치를 관리
- 디스크는 4 킬로바이트에서 16 킬로바이트 정도의 블록들로 나뉘어짐

u 버퍼(buffer) 관리기

- 디스크로부터 데이터를 저장하기 위해 주기억 장치에 있는 버퍼를 관리
- 파일 관리기를 통해 디스크로부터 데이터 블록을 가져옴
- 블록을 저장할 주 기억 장치의 페이지를 선택

트랜잭션 관리기

ㄴ 트랜잭션(transaction)

- ACID 특성을 만족하는 연산들의 그룹

ㄴ ACID 성질

- 원자성(**Atomicity**) : 모두 실행되거나 전혀 실행되지 않아야 함
- 일관성(**Consistency**) : 트랜잭션의 실행이 완료된 후에는 데이터베이스가 모든 일관성 조건을 만족해야 함
- 고립성(**Isolation**) : 트랜잭션이 병행수행되는 다른 트랜잭션으로부터 영향을 받아서는 안됨
- 지속성(**Durability**) : 완료된 트랜잭션의 효과가 손실되어서는 안됨

트랜잭션 관리기 (계속)

u ACID 특성을 위한 기법

- 로킹(locking) : 고립성과 일관성의 보장
 - » 한 트랜잭션이 어떤 항목에 로크를 설정하면, 다른 트랜잭션은 이 항목에 접근할 수 없음
- 로깅(logging) : 지속성과 원자성의 보장
 - » 데이터베이스의 변화와 각 트랜잭션의 상태를 지속적으로 기록
 - » 비휘발성 저장 매체에 기록
- 트랜잭션 완료(commit)
 - » 트랜잭션이 완료되면 데이터베이스의 변화는 영구적으로 데이터베이스에 반영됨
 - » 로그 우선 기록(write ahead logging): 변화가 DB에 반영되기 전에 로깅함

데이터베이스 시스템 공부의 개요

u 데이터베이스 설계

- 어떻게 하면 유용한 데이터베이스를 만들 수 있는가? 어떤 종류의 정보들이 데이터베이스에 저장되어야 하는가? 정보는 어떻게 구조화되는가?
- 데이터 모델링, **ER** 모델, 정규화, 객체-관계형 모델

u 데이터베이스 프로그래밍

- 데이터베이스에 대한 질의와 기타 다른 연산들을 어떻게 표현하는가?
트랜잭션이나 트리거와 같은 **DBMS**의 다양한 기능들을 어떻게 사용하는가?
- 관계대수, **SQL**, 트랜잭션 프로그래밍, 객체 지향 **DB** 프로그래밍

u 데이터베이스 시스템 구현

- (질의 처리, 트랜잭션 처리, 효율적인 접근을 위한 저장매체의 구성 등을 포함하여) **DBMS**를 어떻게 만들 것인가?
- 저장 관리, 질의 처리, 트랜잭션 관리