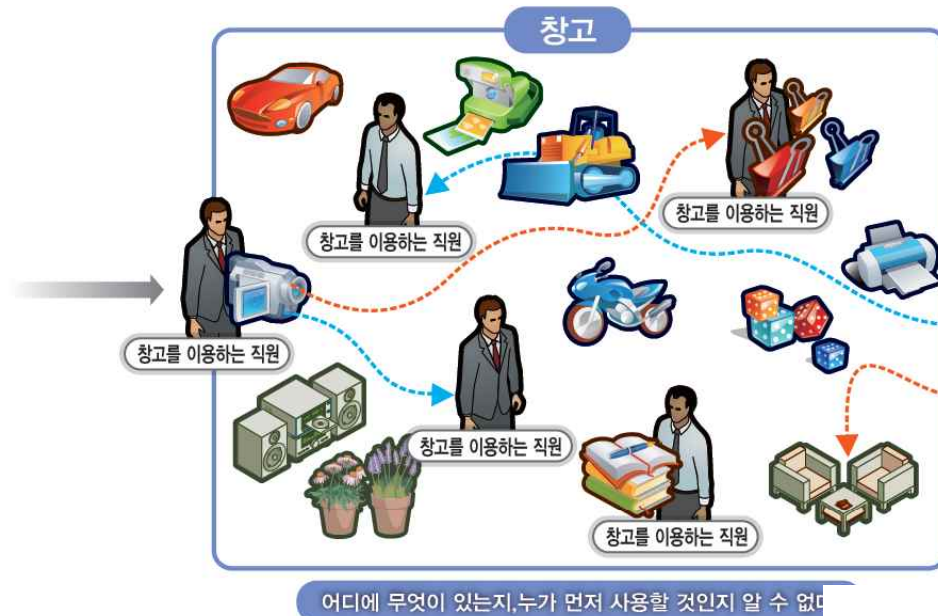


Database Technology

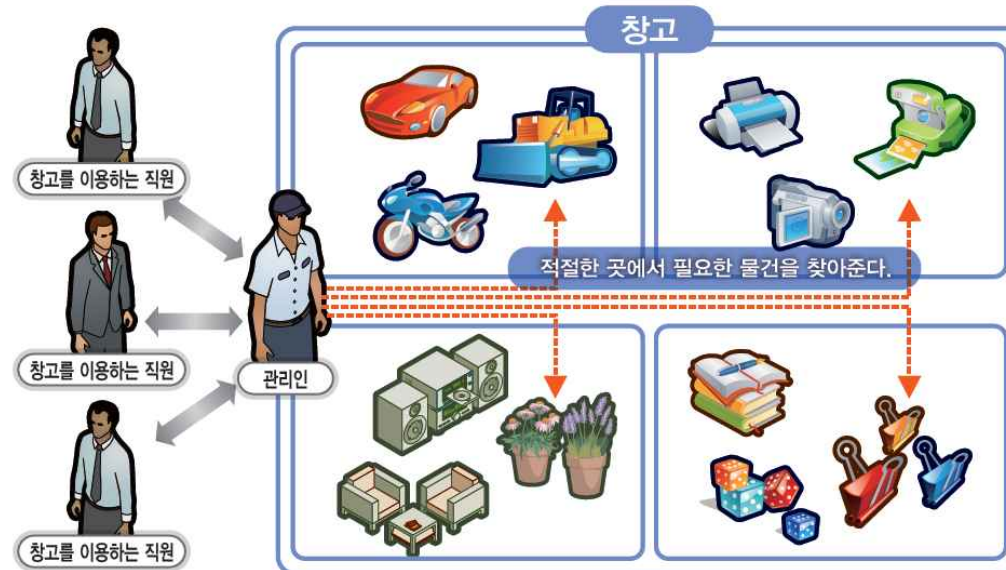
김 형주 교수
Internet Database Lab
서울대학교 컴퓨터공학부



Data Management

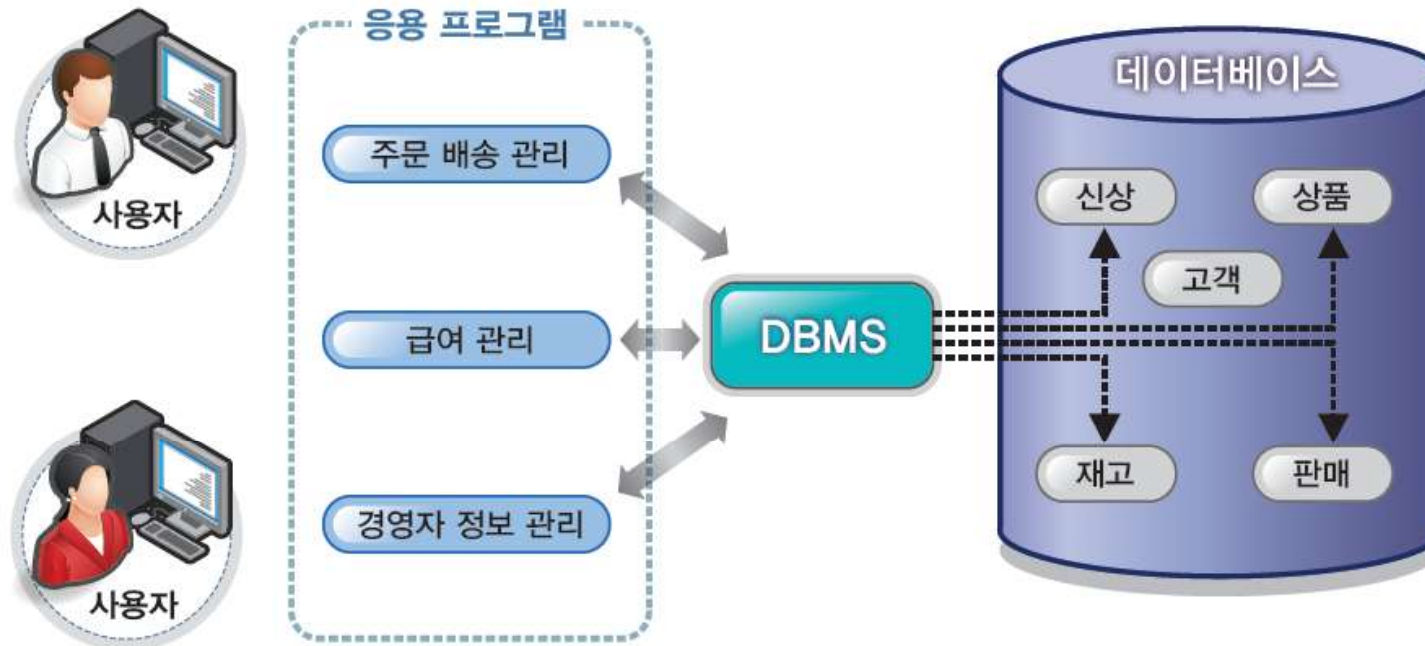


<혼란스러운 창고>



<정리된 창고>

Data Management – Dedicated System

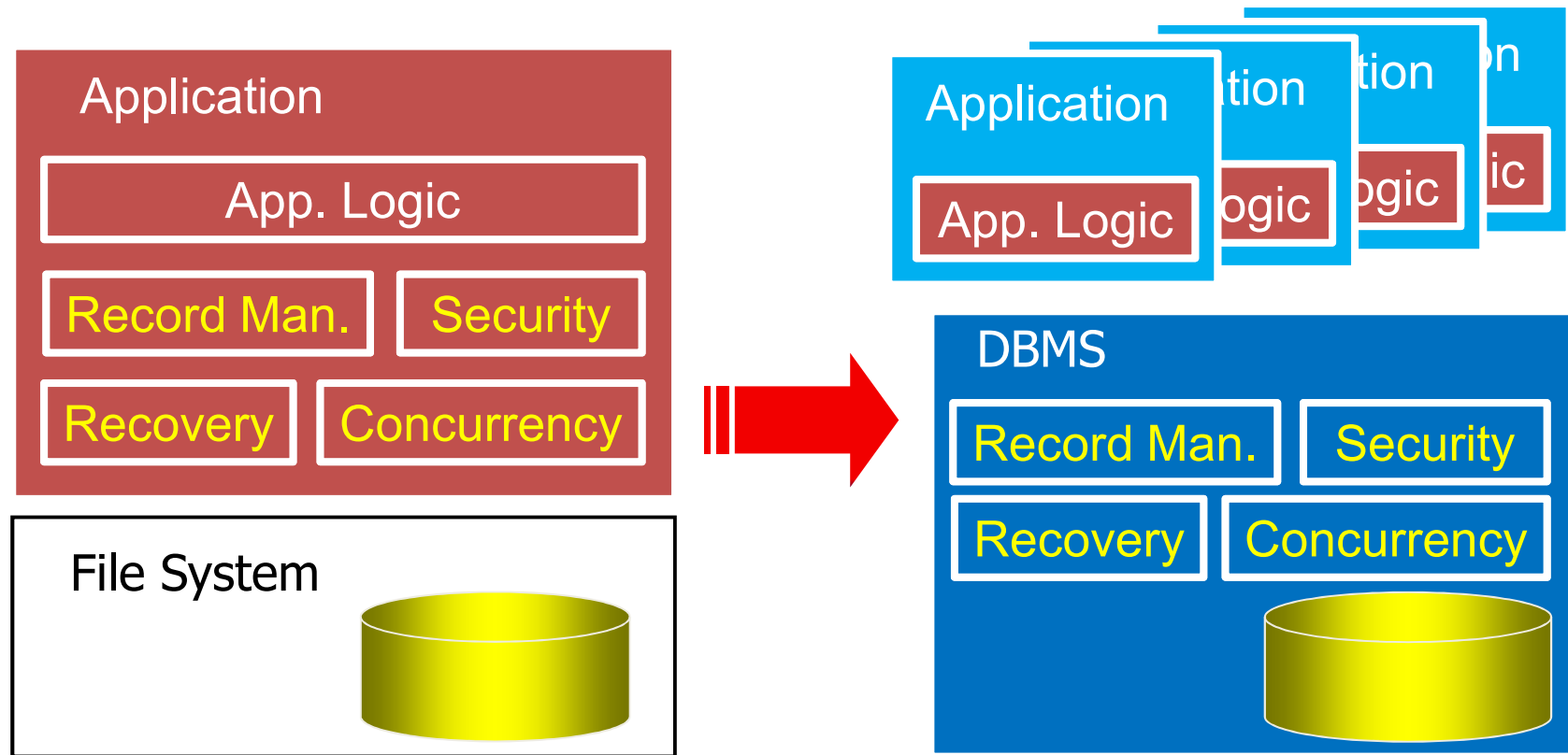


물건	데이터
창고	데이터베이스(디스크)
창고관리인	Dedicated System - DBMS
직원	응용 프로그램 또는 사용자

File Systems

- File System
 - Core part of OS
 - Stores programs, data, documents, or anything
 - (in disk)
- Drawbacks:
 - Redundancy and Inconsistency
 - Multiple file formats, duplication of information in different files
 - Atomicity of updates
 - Failures may leave database in an inconsistent state with partial updates carried out
 - Concurrent access by multiple users
 - Concurrent accessed needed for performance

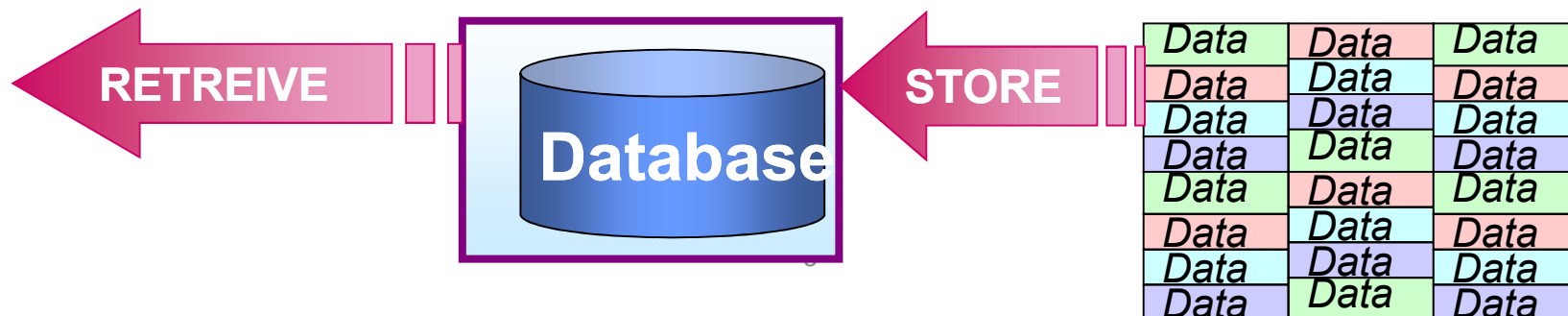
Database Management System (DBMS)



Data Base Management System (DBMS)

- Basic functionalities

- Store and Retrieve massive data effectively
- Provide “ad-hoc” queries
- Provide concurrent accesses to data (transaction)
- Keep the integrity of data (recovery) despite of failures
- Provide standard platform for application SWs
- Enforce security constraints on data

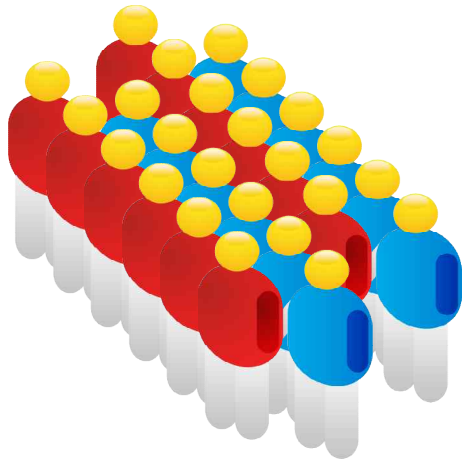


Simple Data Base

- Mobile phone accounting data



120KB record per call



Phone No	Name	station	Start-time	End-time
----------	------	---------	------------	----------

.....
-------	-------	-------	-------	-----

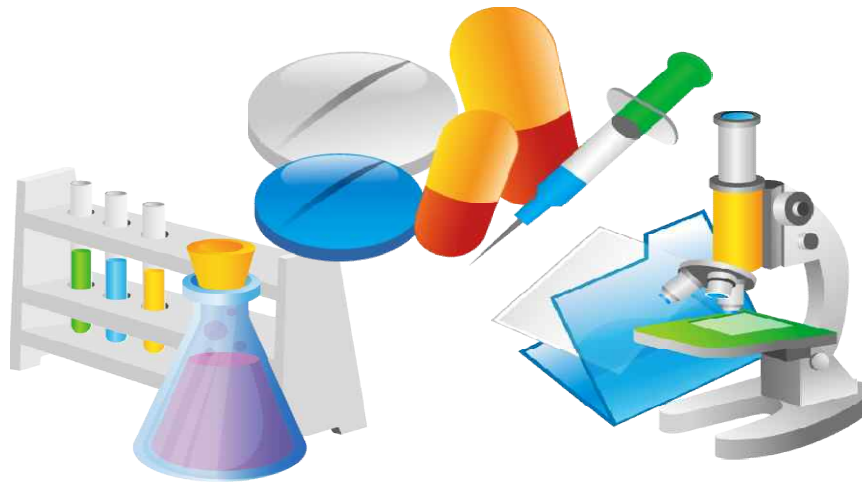


.....
-------	-------	-------	-------	-----

40M Persons * 120 Byte * 50 calls/day * 365 days
= 80000 G Byte / 1 year = 80 Tera Byte / 1 year

Fastly Growing Big Data Base

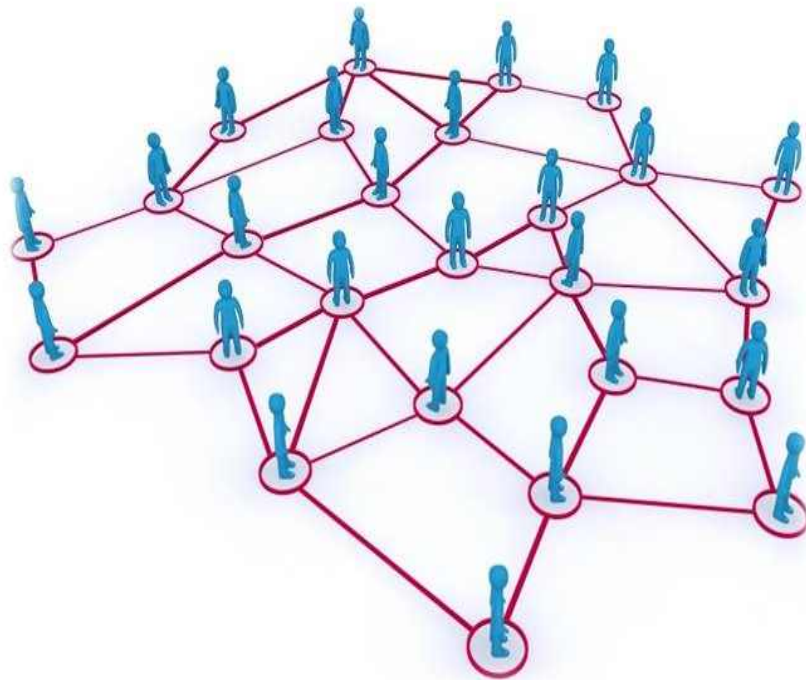
- NCBI (National Center for Biotechnology Information)



GenBank

- management of information of 165,000 species
- add 3 million new DNA sequences monthly

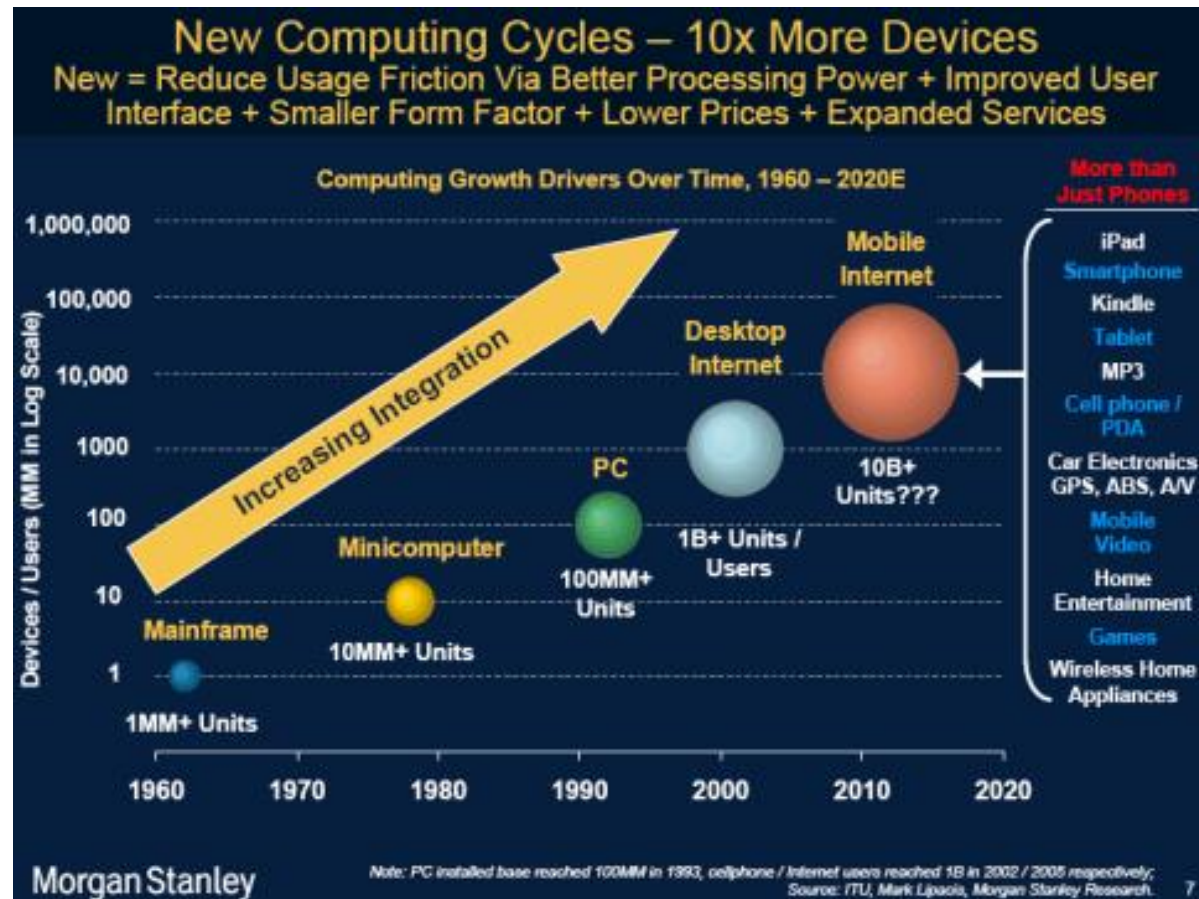
Enormous Data in Social Network



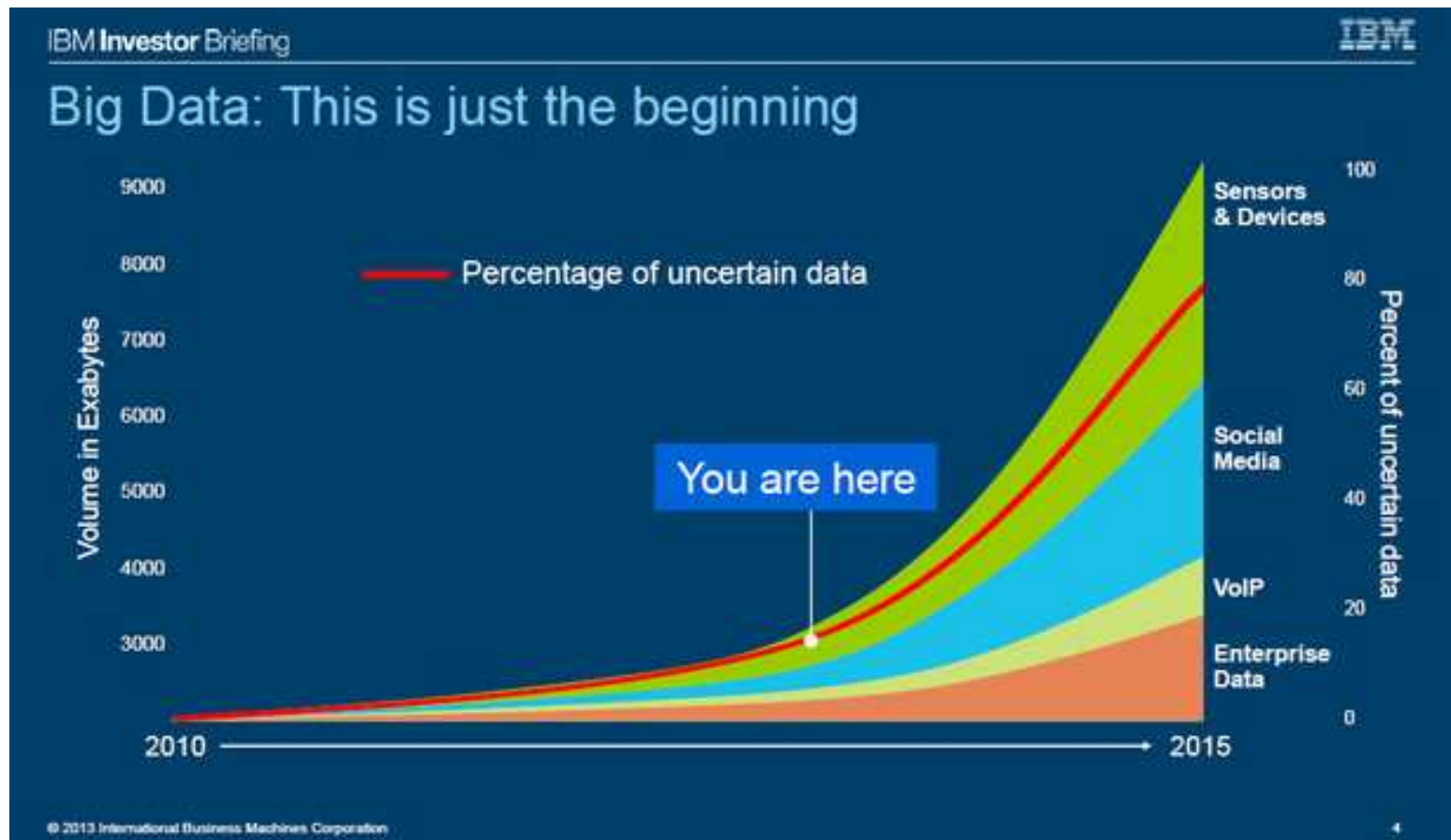
Web(Internet)을 소통과 협력의 도구로 활용

~~15,000,000,000~~ Smart Devices (5000억개)

- Programmable (프로그램 작동이 가능하고)
- Internet connected (인터넷 접속된)

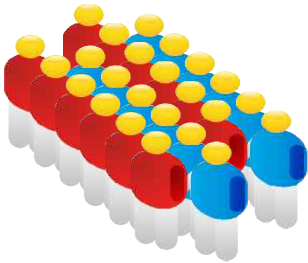


Data Explosion!



Role of DBMS [1]: Simple record search

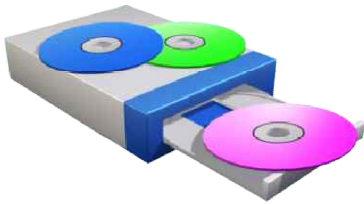
- 주민번호 “840101-1212141” 인 학생의 수능 수학성적을 찾아라?



740,000명 * 5 records = 3,700,000 records

If 12ms is required for fetching a record & checking using a file system

3,700,000 * 12ms = 44.4K secs = *over 12 hours*



If we use DBMS, it will be less than **0.1sec!**

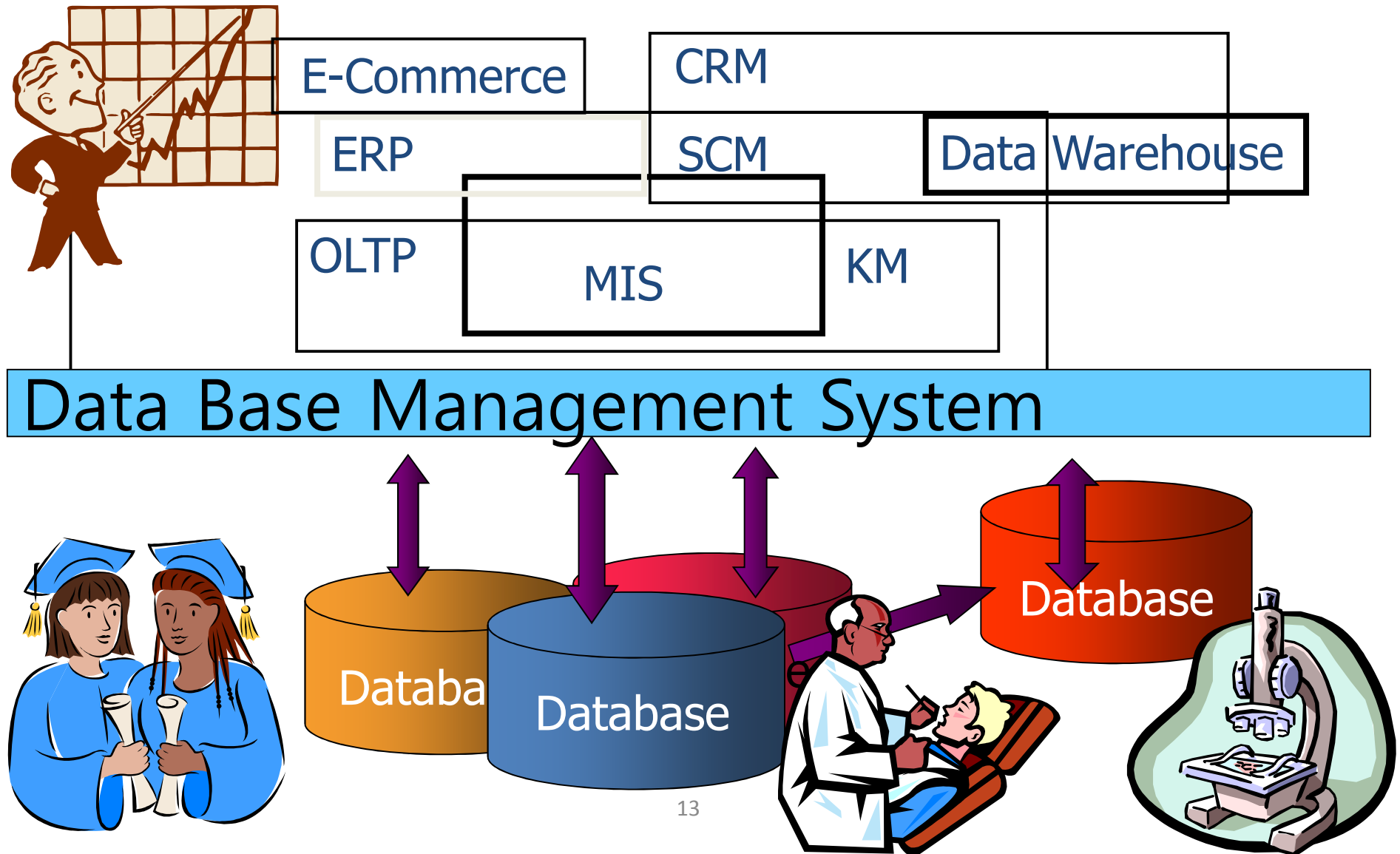
Statistical processing
for population census



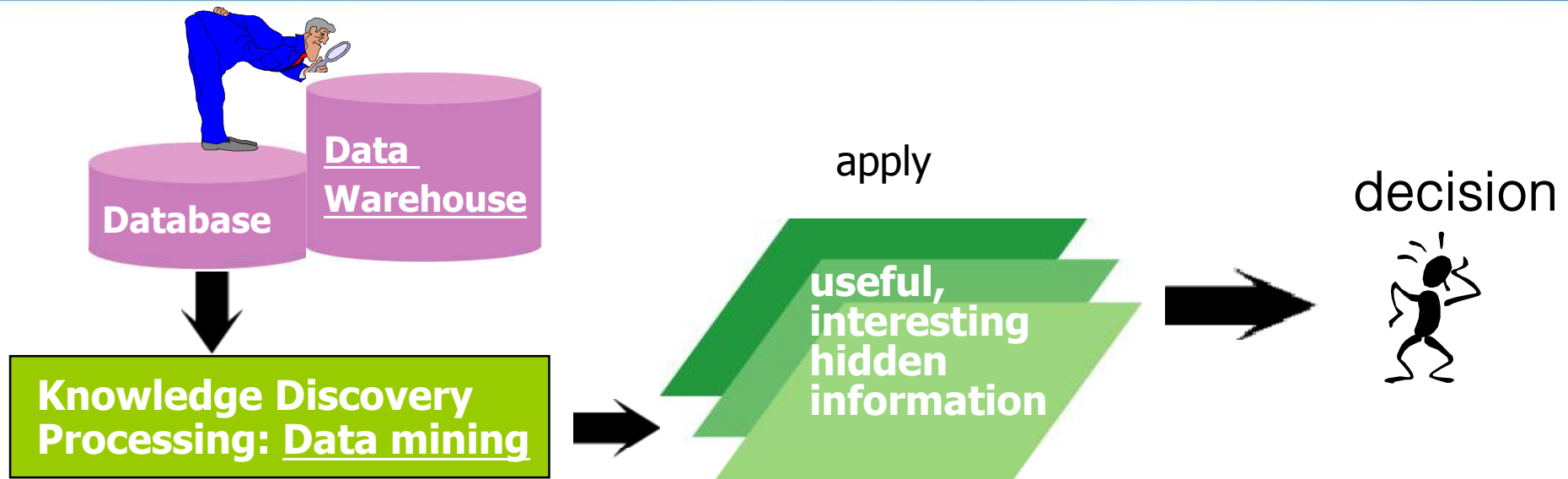
Search for the purchase
pattern on customer
groups

Search for the correlation
between gene and disease

Role of DBMS[2] : Supporting Enterprise Applications



Role of DBMS [3] : Even Knowledge Discovery!



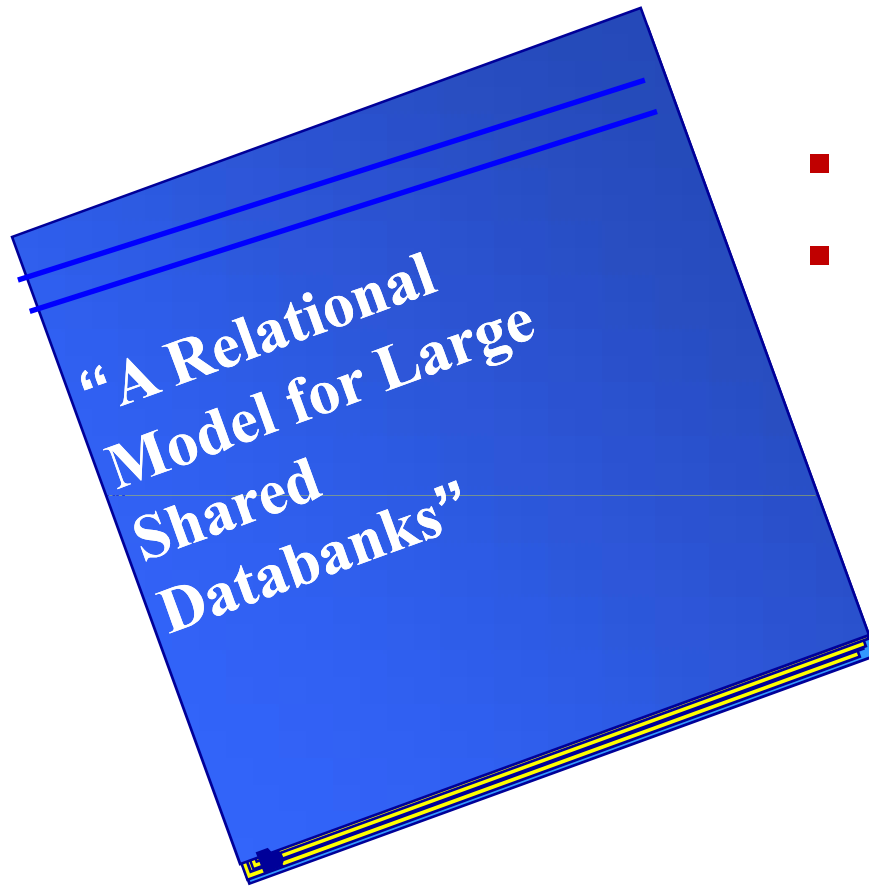
■ Data Analysis/Mining

- * 빵과 과자를 사는 사람의 80%는 우유를 같이 산다
- * 분유와 기저귀를 사는 사람의 74%는 맥주를 같이 산다

■ Decision Making

- * 상품 진열대에 [빵, 과자, 우유], [분유, 기저귀, 맥주]를 같이 진열
- * 우유 소비를 조절하기 위해 빵, 과자 가격을 조정

In The Beginning...



- Everything in Table
- Set-oriented Query Language

E.F. Codd

- 1970 CACM Paper
- Turing Award

1970 →

A Sample Relational Database

<i>customer-id</i>	<i>customer-name</i>	<i>customer-street</i>	<i>customer-city</i>
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

<i>account-number</i>	<i>balance</i>
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

<i>customer-id</i>	<i>account-number</i>
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

SQL: supporting ad-hoc queries

- SQL: widely used commercial query language
 - E.g. Find the name of the customer with customer-id 192-83-7465

```
select    customer.customer-name
from      customer
where     customer.customer-id = '192-83-7465'
```

- E.g. Find the balances of all accounts held by the customer with customer-id 192-83-7465

```
select    account.balance
from      depositor, account
where     depositor.customer-id = '192-83-7465' and
           depositor.account-number = account.account-number
```

Experimental RDBMS Prototypes



1970



1979

Commercial RDBMS Products

ORACLE

INGRES
(commercial)



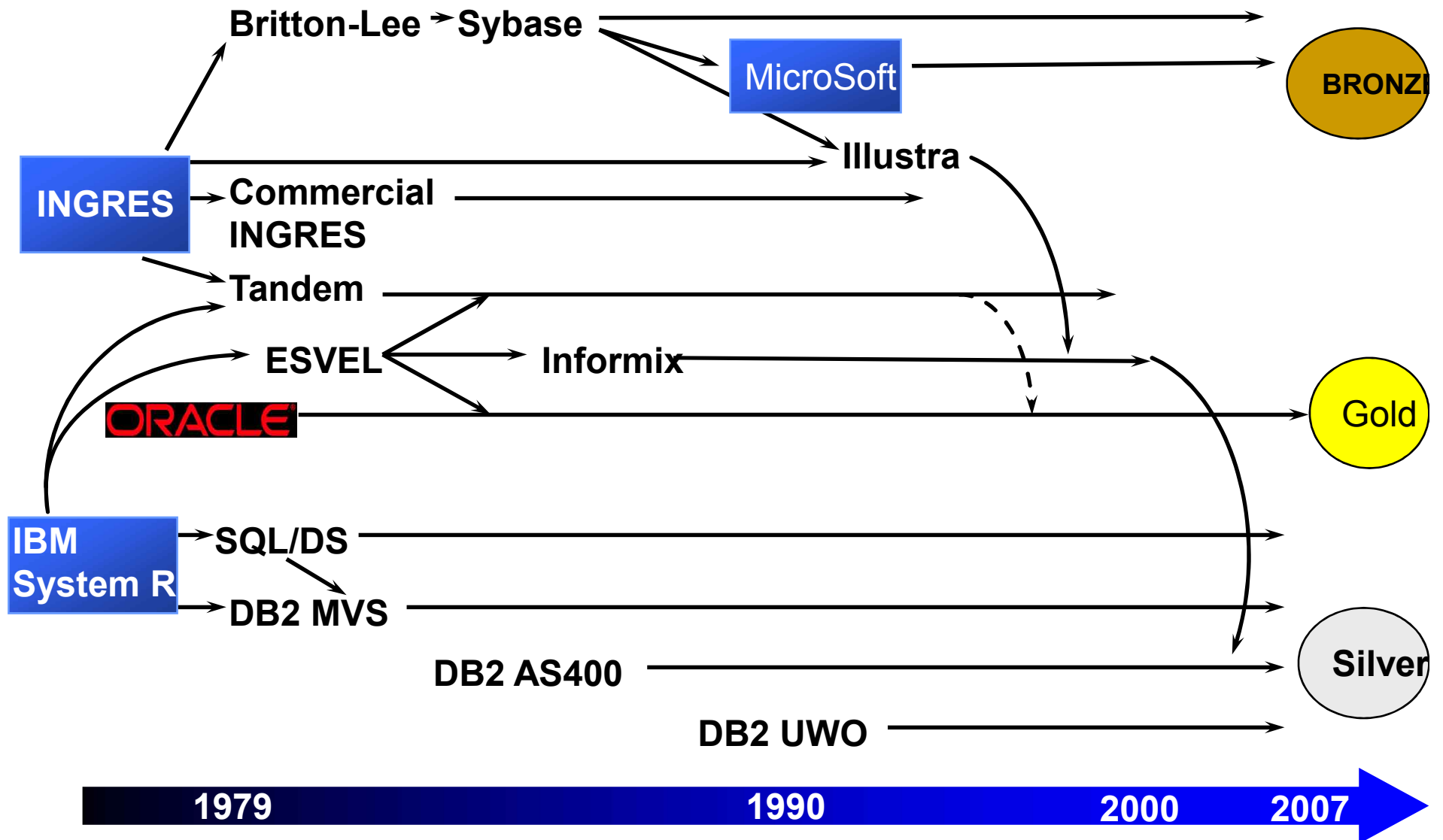
SQL/DS

1979

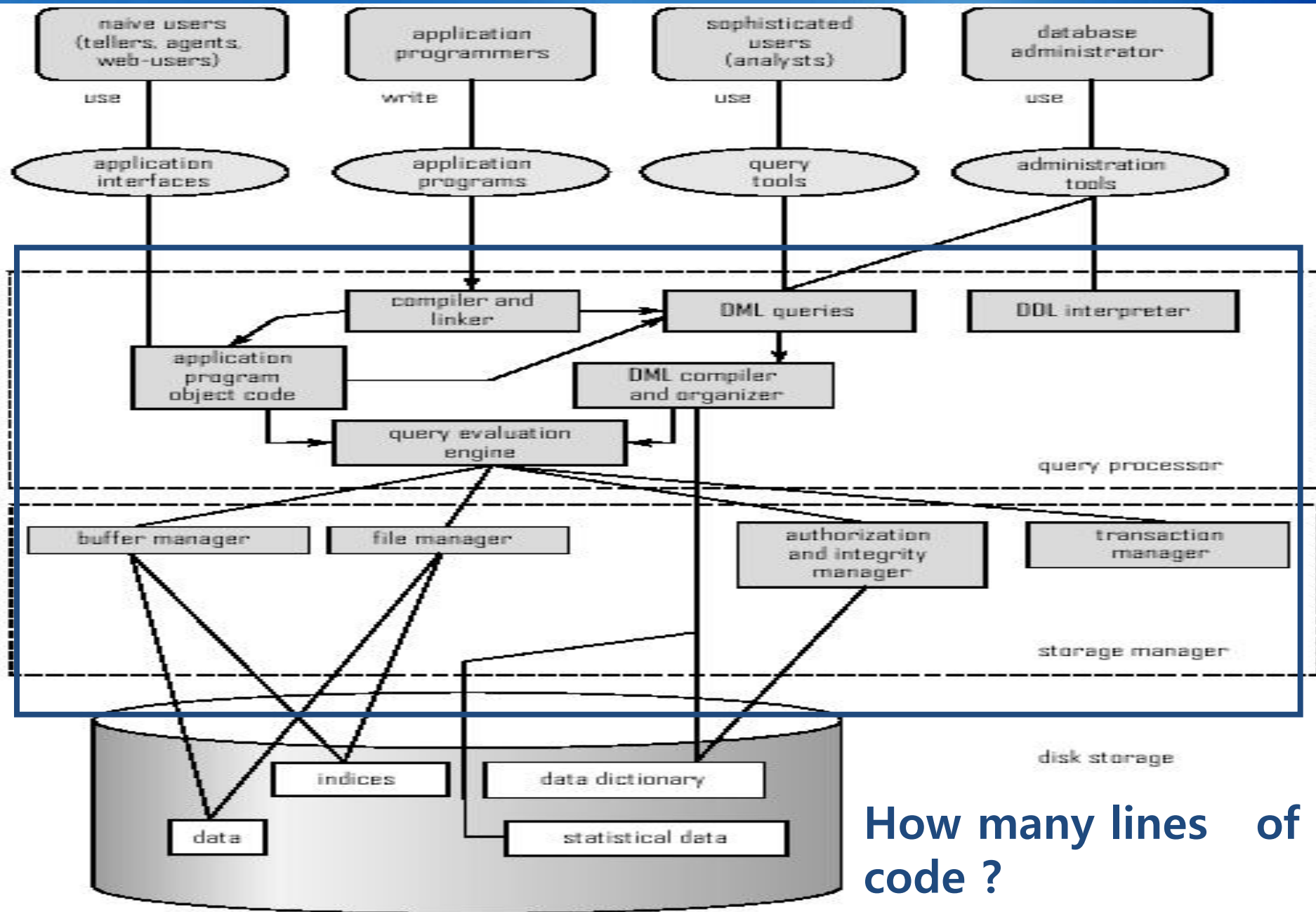


1982

Genealogy of Commercial DBMS Products



범용 “Disk-기반” DBMS Architecture



How many lines of code ?

Database Companies in the World



DBMS Market Share: World Wide (2011)

리서치 전문회사 [가트너](#)

2011년 매출기준 TOP 5 상용 관계형 데이터베이스 업체

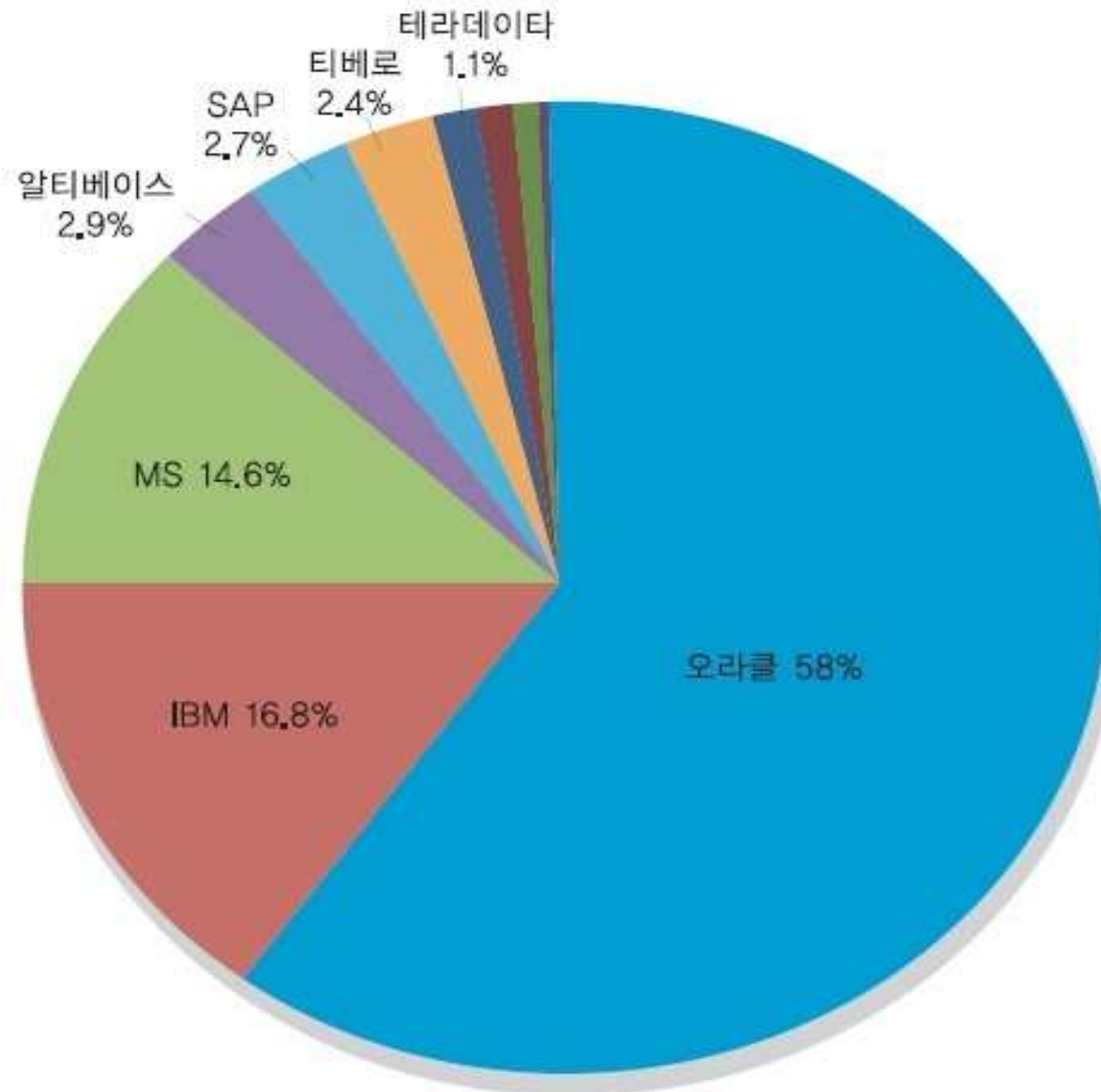
- Oracle (48.8%)
- IBM (20.2%)
- MicroSoft (17.0%)
- SAP (4.6%)
- TeraData (3.7%)

Source: Gartner Dataquest

국내 DBMS 시장규모 (5천억원 내외)



국내 DBMS Market Share (2013)



Lawrence Joseph Ellison (Oracle Founder)

- Univ of Chicago, Physics Major
- 1976년까지 캘리포니아의 중소기업에서 SW Programmer
- 1977년 1200달러로 Oracle 창업
(Oracle: 신탁, 신의 뜻)
- 보유주식: 430억달러 (약 50조원)
- 세계 부자순위 5위
- Oracle 현황
 - 오라클 2014년도 매출 380억달러 (약 42조원)
 - DBMS, ERP, Data Warehouse, etc
 - 고객: 미국CIA포함 전세계 27만개 기업
 - 직원: 전세계 145개국에서 13만2천명
 - 기업용 SW업체 중 1위, 종합 SW 업체 순위는 2위



ORACLE®

Oracle 성장 과정

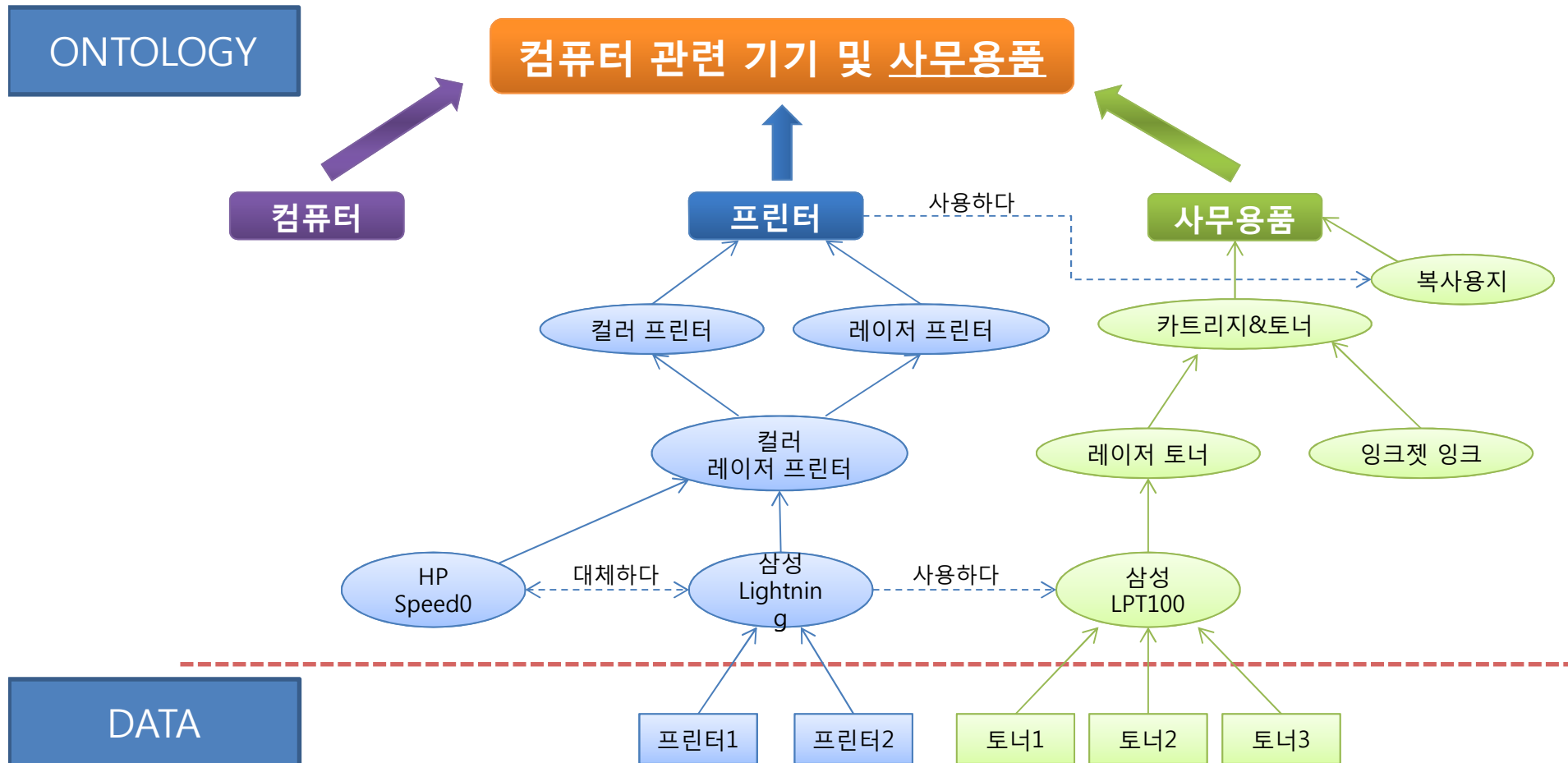
- 1970년대말: CIA의 수집된 정보를 체계적 관리·분석해주는 프로젝트 성공
- 1980년대: 관계형 DBMS의 시장점유에 선도적 역할
- 1990년대: Internet & E-Business 열풍과 맞물려 또 한번 폭발적 성공
- 2000년 이후: 기업용 SW의 모든 것을 공급하는 One-Stop 서비스 업체
- 2000년 후반기: 200억달러를 들여 무려 27개의 기업 M&A
- 2009년 세계 4대 HW업체인 미국의 [선마이크로시스템스](#)를 \$74억에 인수

최근 Database 기술의 주된 발전방향

- Intelligent Retrieval (지능형 검색)
- Large scale Processing (대규모검색, 빅데이터처리)

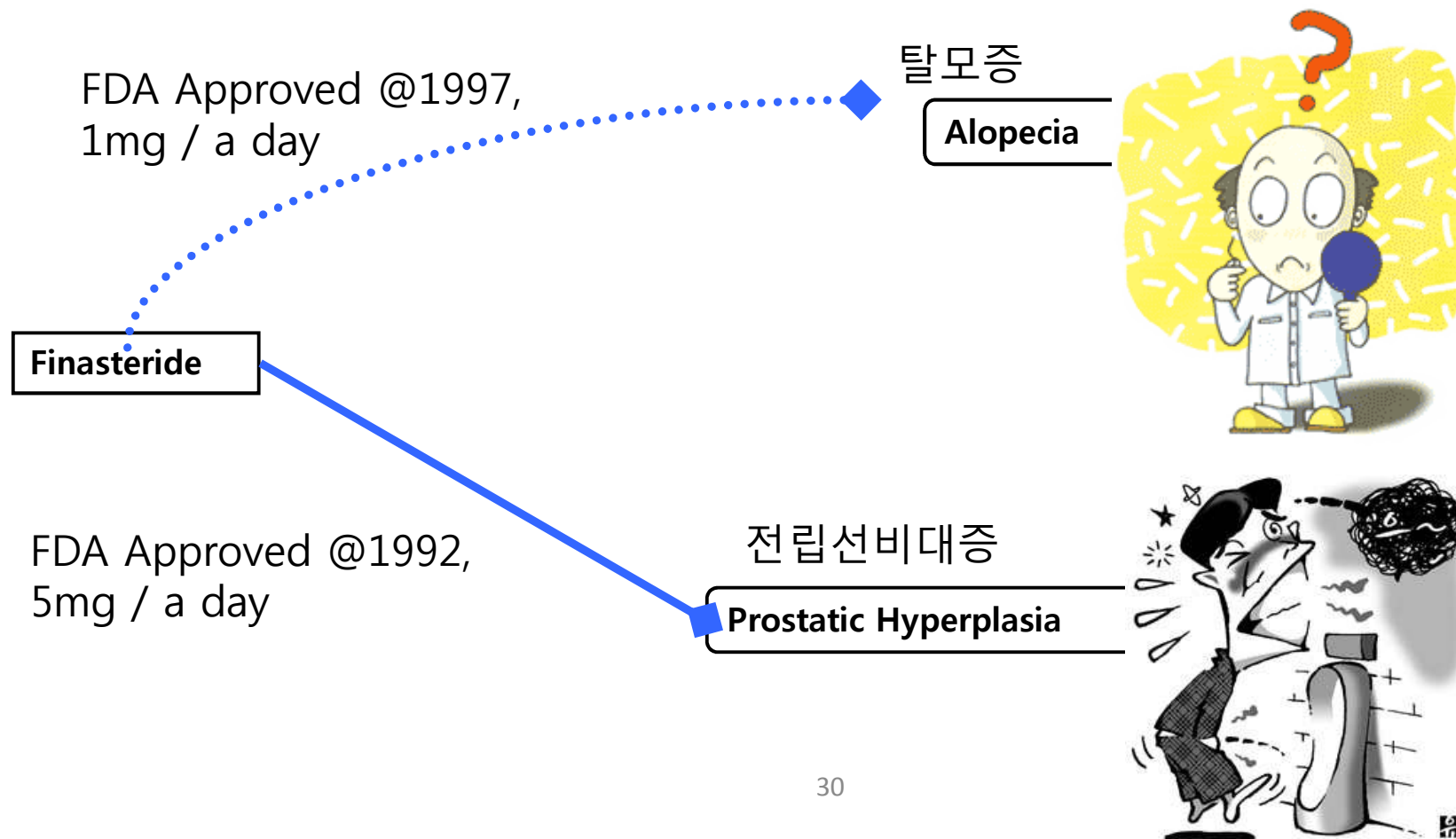
Ontology: 지식과 개념의 표현과 처리기술

컴퓨터 관련 기기 및 사무용품 온톨로지

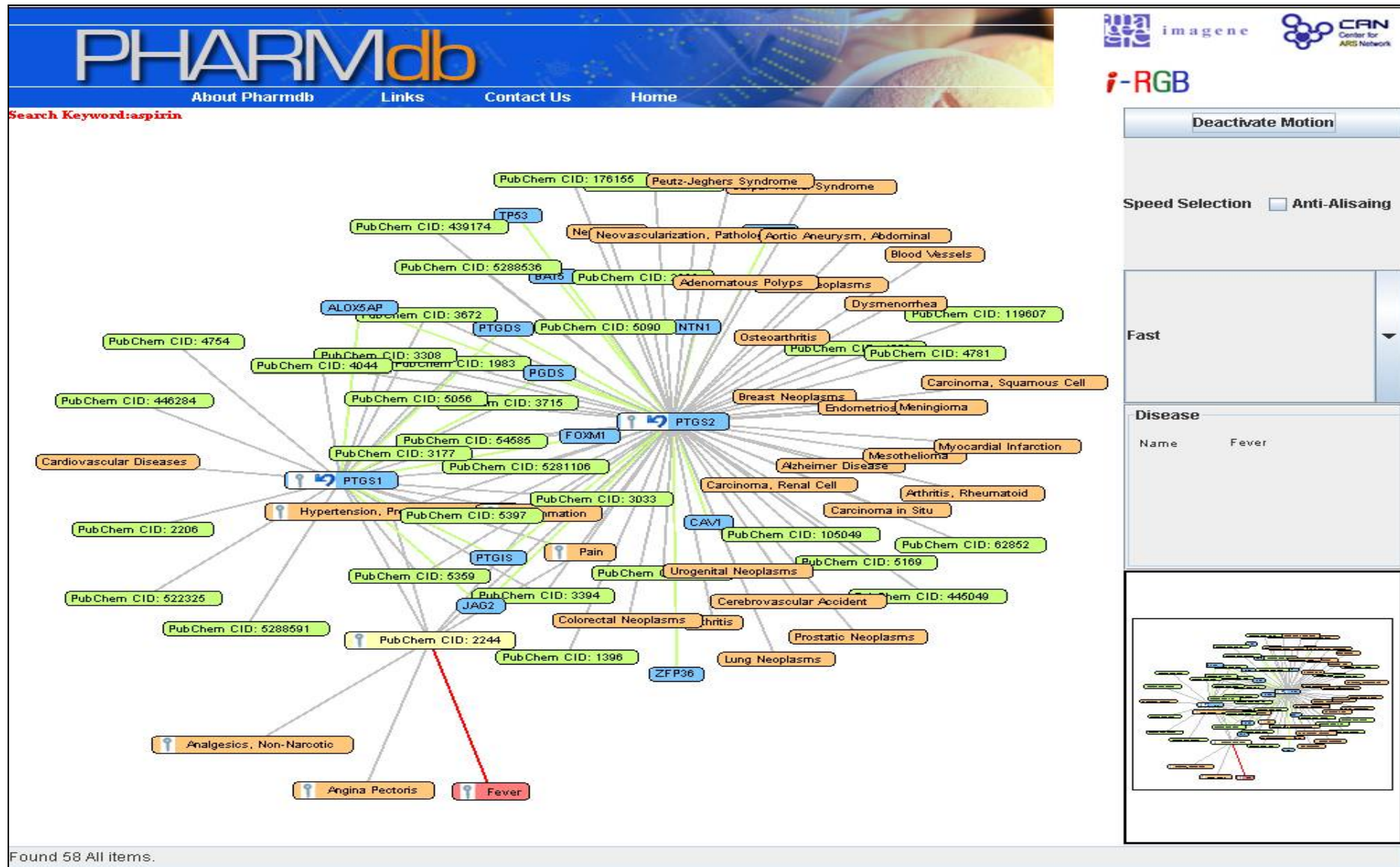


Ontology Application: 신약개발분야

Ontology Applicable Example: Drug repositioning of Finsteride

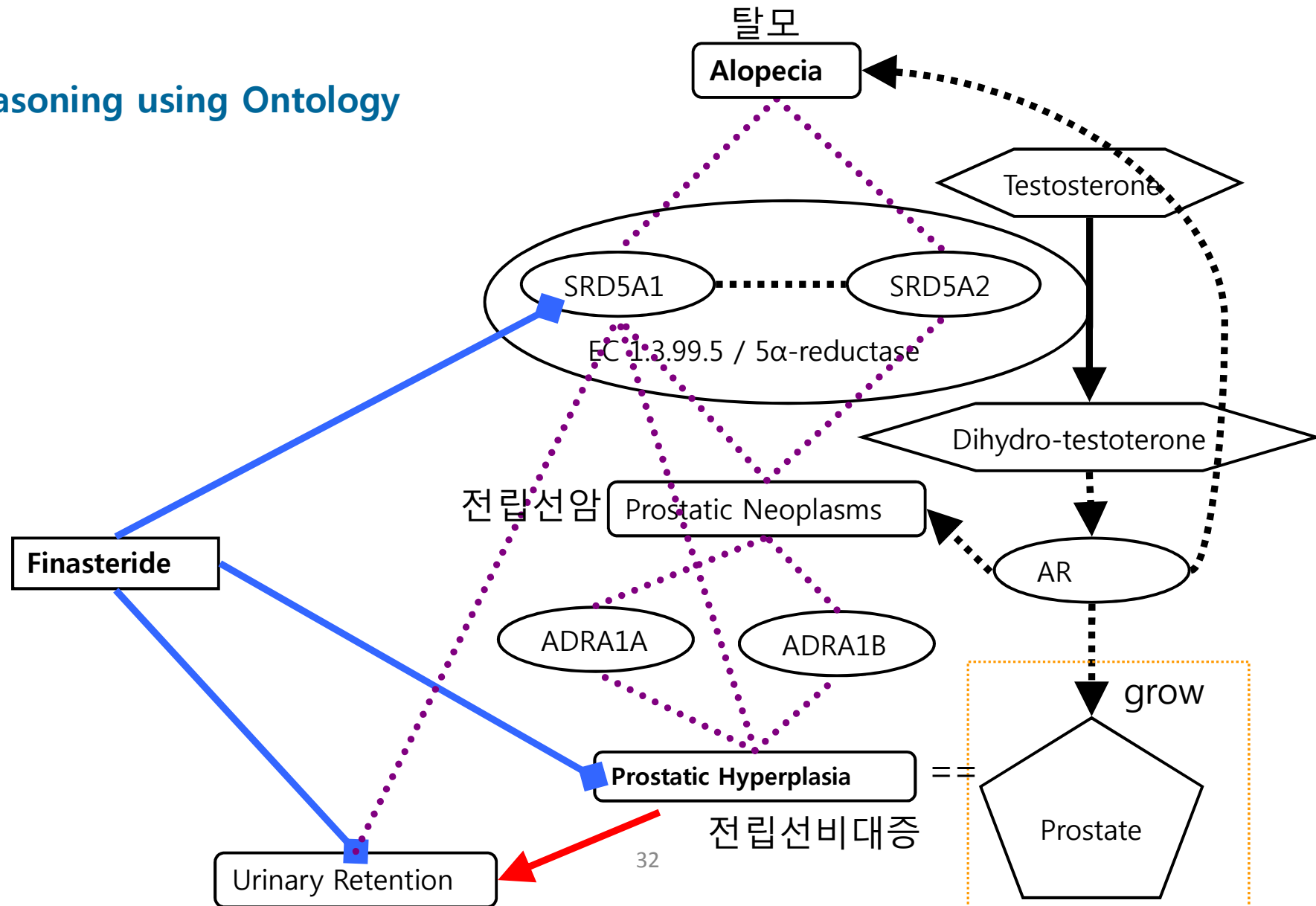


Ontology Application: 신약개발분야



Ontology Application: 신약개발분야

Reasoning using Ontology



Large Scale Web Search

보조자료



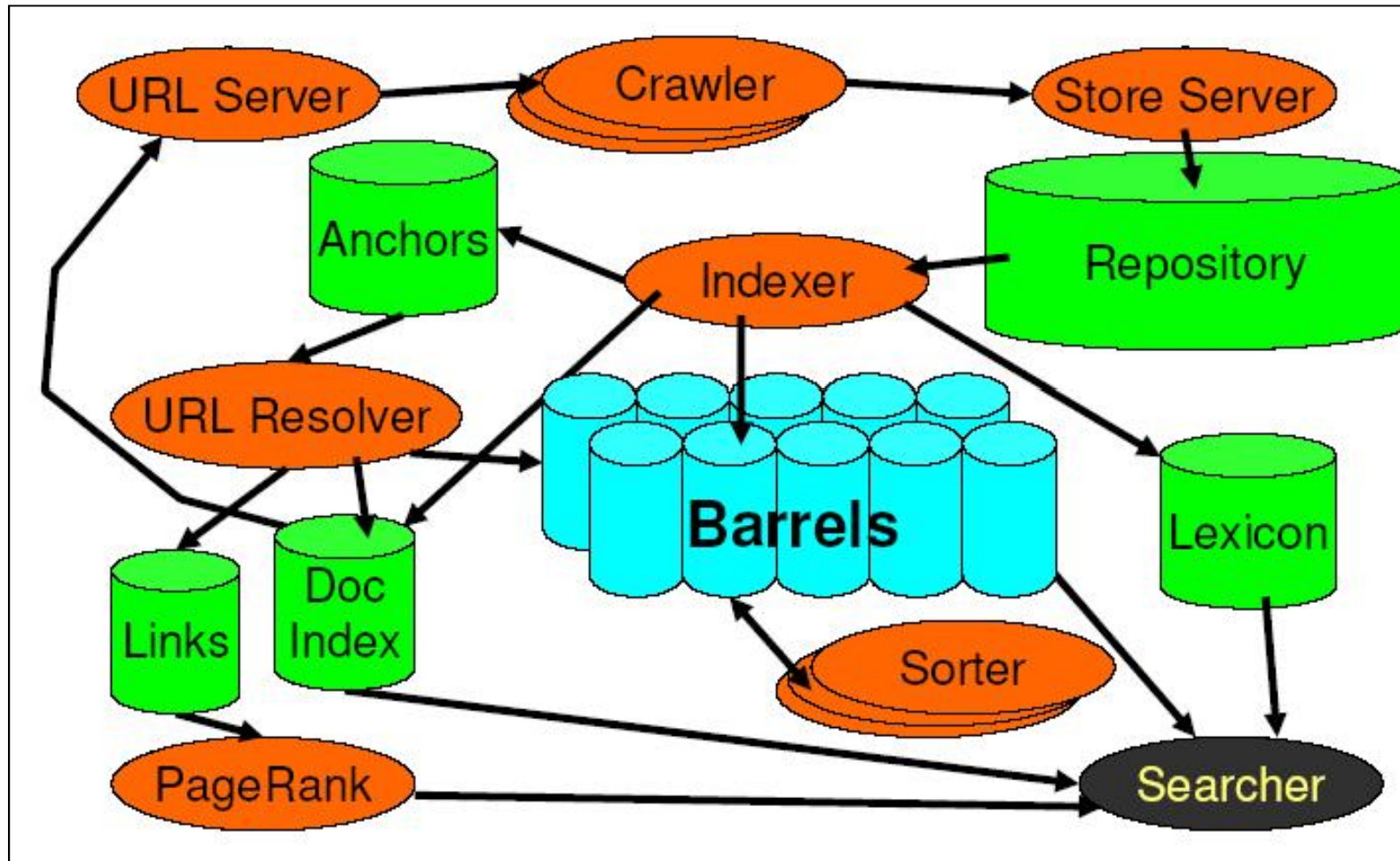
```
potato: (url_zz; 3, 101, 178, 2009); (url_pq; 1; 809); ...  
quake: (url_ds; 1; 16); (url_lk; 4; 3, 11, 12, 678); ...
```



Figure 5.3 In Google's data center, Dalles, Oregon. A search engine's index is huge, because *in principle* it keeps URLs for most of the words used on the Web; Google's index has been reported to be "100 million gigabytes" = 10^{17} bytes. However big it is, they can't store just one copy, because they need a backup in case some of those LEDs go dead.

Large Scale Web Search

** Google Search Engine Architecture



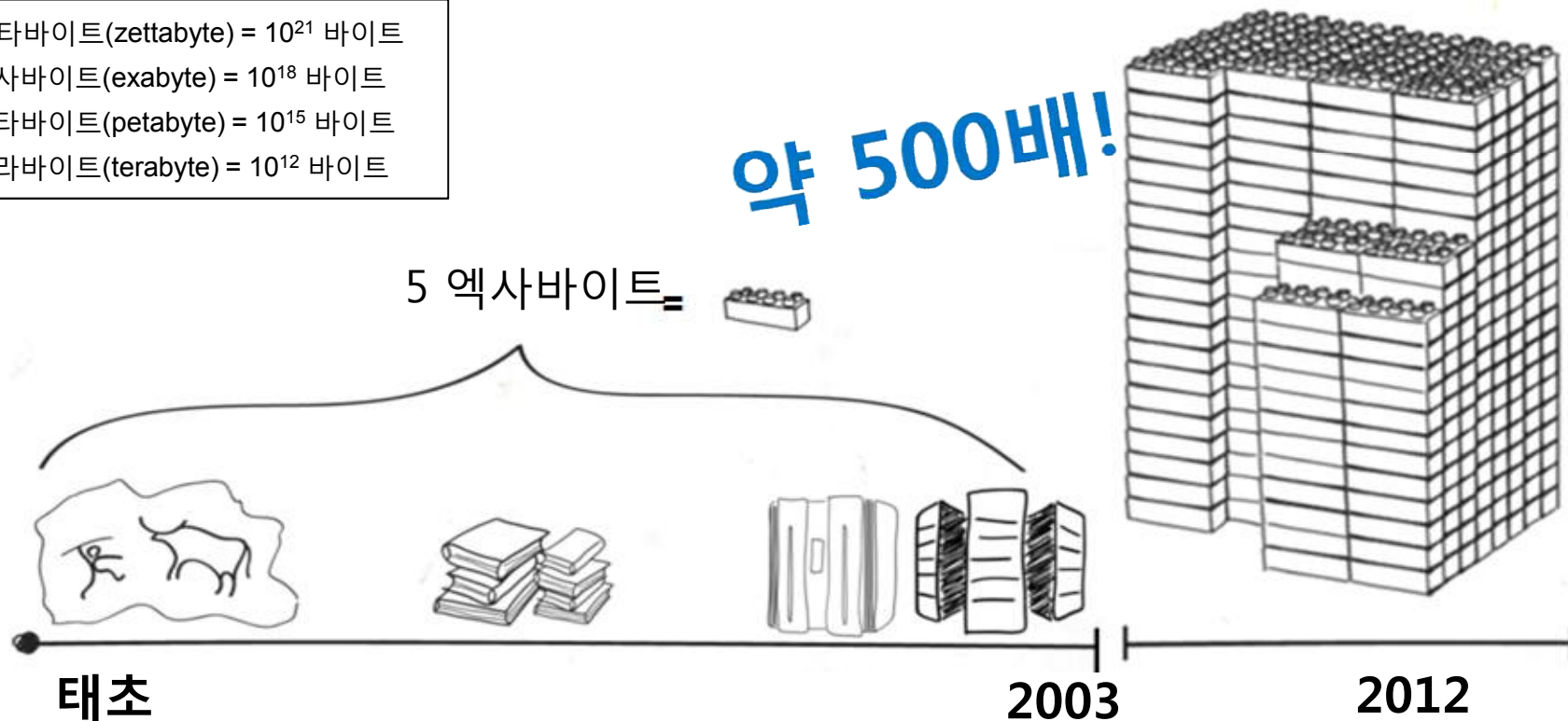
** Google Map-Reduce Framework! → Big Data Processing

Big Data의 시대의 도래!

- 2012년 한 해동안 생성된 디지털 데이터
➔ 2,700,000,000,000,000,000,000 바이트 (2.7 ZB)
 - 원인: 정보화 가속, 모바일·소셜·센서 데이터의 급증
 - 1.5년마다 2배로 증가 ➔ 2020년엔 지금의 20배

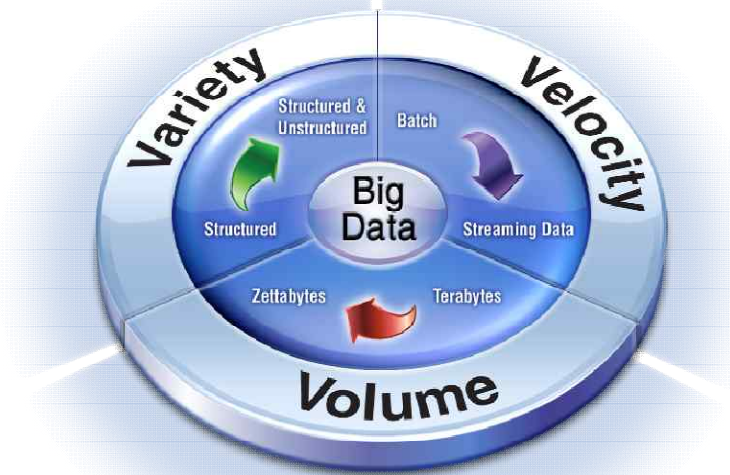
2.7 제타바이트

* 제타바이트(zettabyte) = 10^{21} 바이트
엑사바이트(exabyte) = 10^{18} 바이트
페타바이트(petabyte) = 10^{15} 바이트
테라바이트(terabyte) = 10^{12} 바이트



Big Data의 특징

- “빅 데이터”의 속성: **3V** 또는 **4V**
 - 크기(Volume)
 - 속도(Velocity)
 - 다양성(Variety)
 - 가치(Value)



3V

– Gartner –

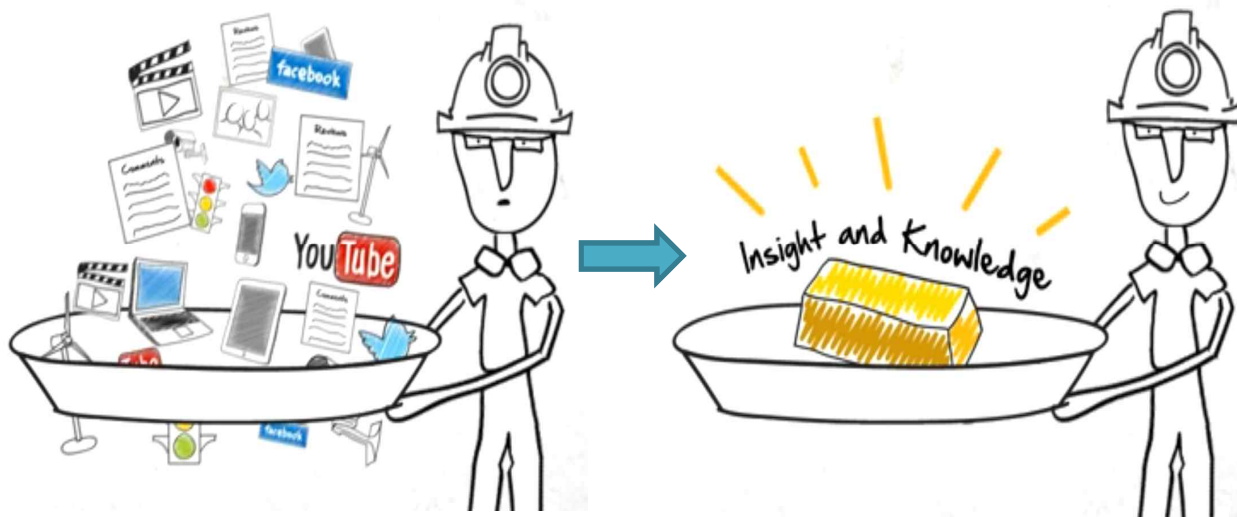


The 4th V

– Oracle –

Big Data가 주는 가치

- 데이터: 의미를 담고 있는 기록된 사실 [Elmasri and Navathe. Fundamentals of Database Systems]
- 그렇다면, 다양하고 많은 “빅 데이터” → 다양하고 많은 의미?
 - “빅 데이터”를 처리, 분석하여 의미를 제대로 찾아낼 때에만!




Big Data 101: How Big Data Makes Big Impacts, Intel
<http://www.intel.com/content/www/us/en/big-data/big-data-101-animation.html>

기계화/자동화 → 제조 프로세스 혁신
빅 데이터 분석 → 판단 프로세스 혁신

MapReduce란?

- **구글(Google)**이 대용량 Web Data처리를 위한 **분산 처리** 프레임워크
 - 큰 작업을 잘게 **나누고(Map)** 종류별로 **모아서(Reduce)** 처리하는 방식
 - MapReduce는 비공개된 Google의 SW
 - **2004년**에 논문을 통해 세상에 알려짐

- MapReduce가 널리 쓰이게 된 이유는? **Hadoop의 등장** 
 - Hadoop: 구글의 공식 허가를 받은 MapReduce의 **오픈 소스** 버전
몇몇 회사는 자체 프레임워크 사용 → 개발 및 유지보수 비용 부담
 - 2006년 초: **Don Cutting**이 **Yahoo!**에서 **Apache Hadoop 프로젝트**

* Hadoop은 Cutting의 아들이 좋아하는 코끼리 장난감 이름



MapReduce 예제: 단어 세기

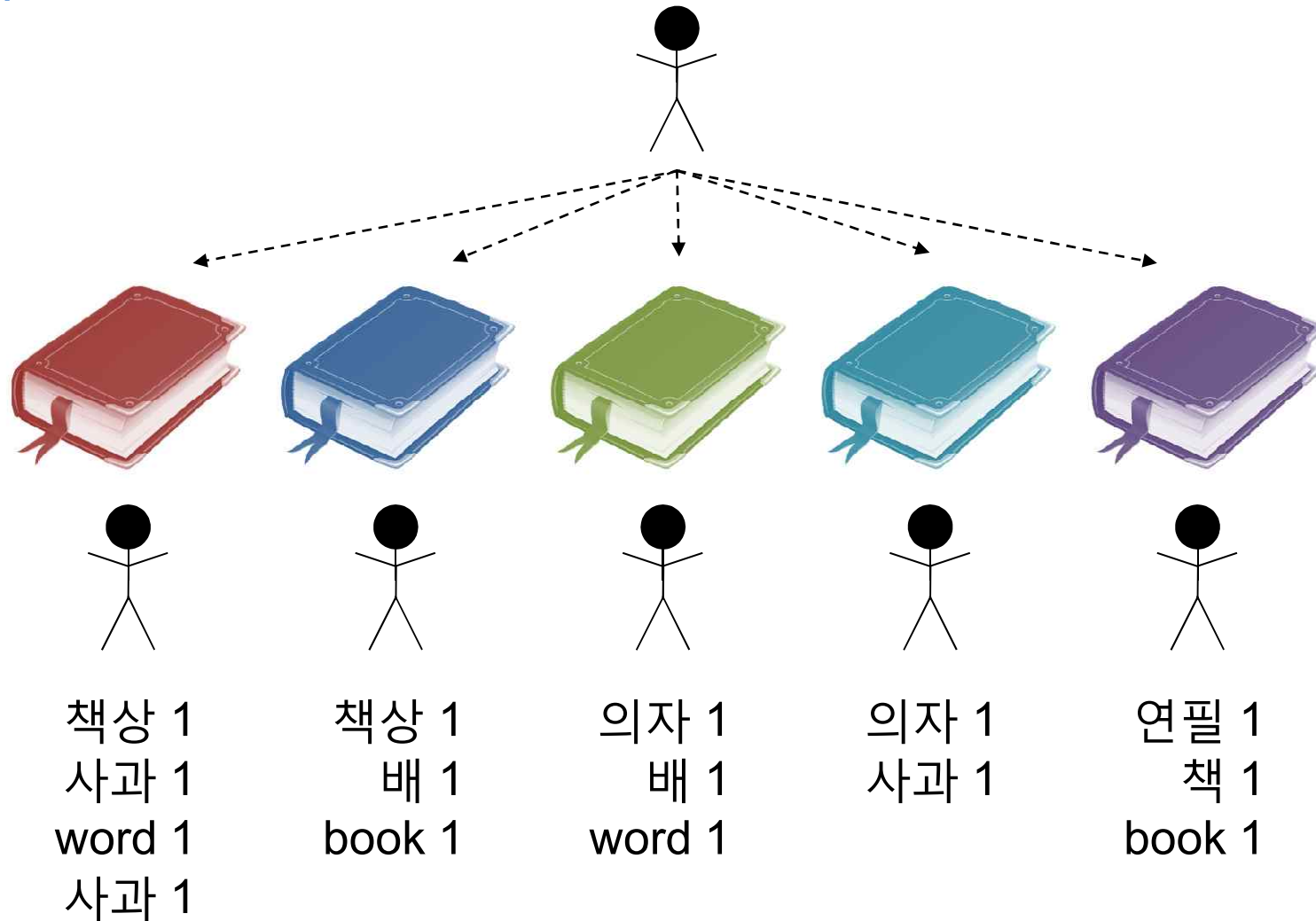
- 임무: 아래 책들에서 각 단어가 몇 번 나오는지 세어주세요.



- 문제점: 양이 너무 많아서 혼자 세면 너무 오래 걸림
- 해결방법: 여럿이서 나누어 하기
 - 여럿에게 일을 **나누어서** 시키고 (Map)
 - 몇 명이 각 결과를 **모아서** 작업을 완료 (Reduce)

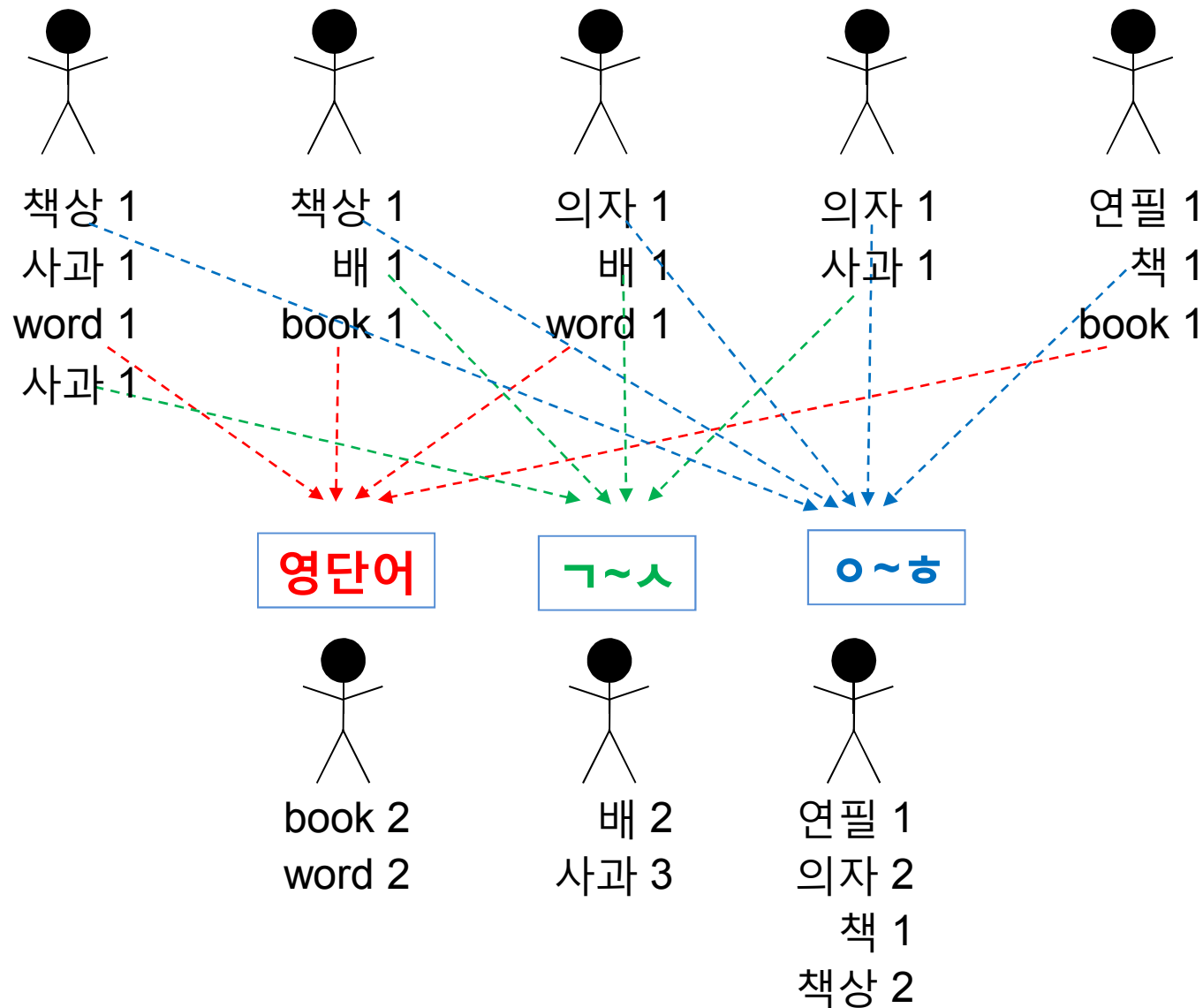
MapReduce 예제: 단어 세기 – Map 단계

- Map: 조금씩 나누어서 일 시키기

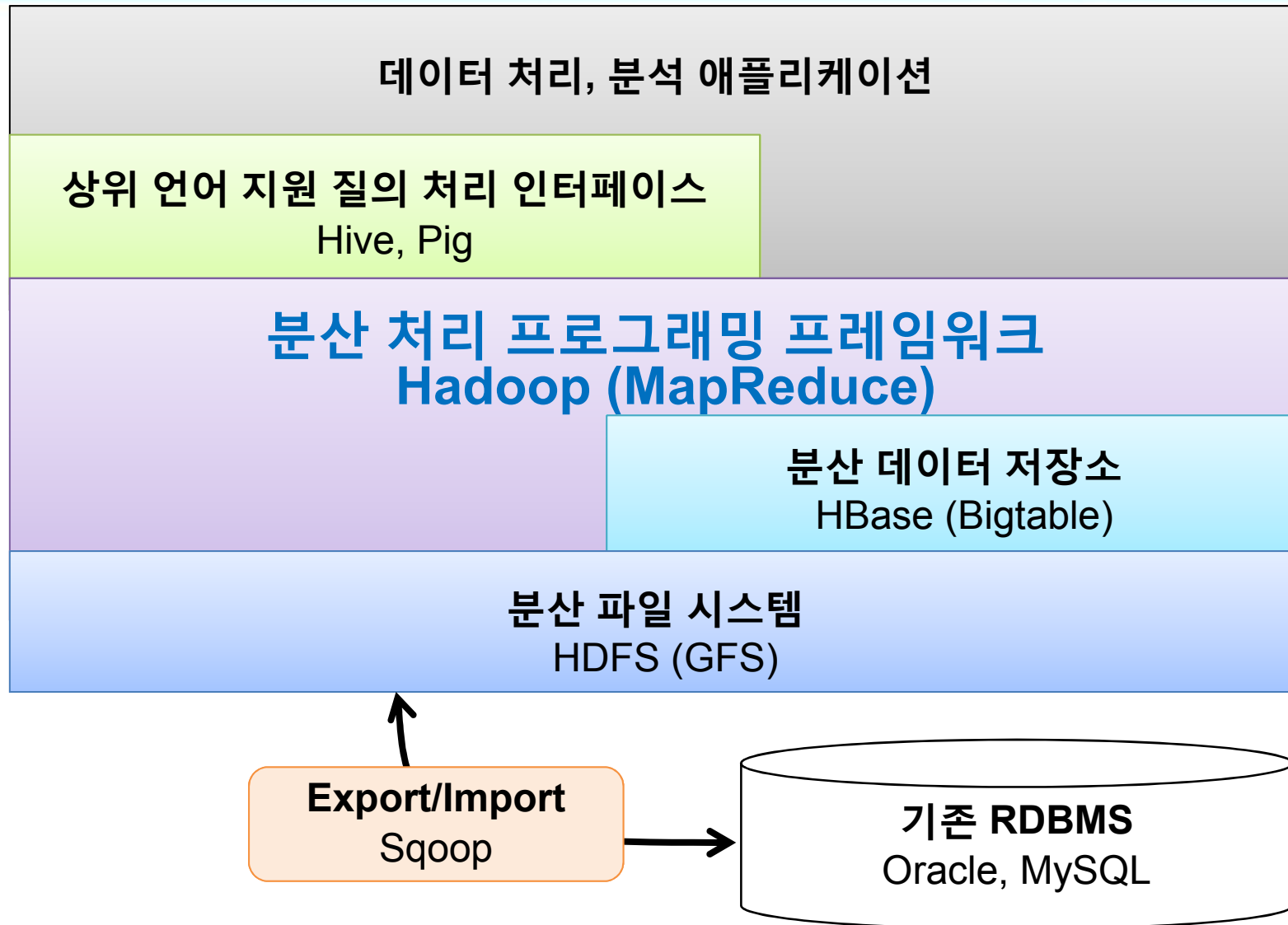


MapReduce 예제: 단어 세기 – Reduce 단계

- Reduce: 몇 명이 각 결과를 **모으기**



Hadoop 관련 시스템 구조



Database System Concepts

Chapter 1: Introduction

Part 1: Relational databases

Chapter 2: Introduction to the Relational Model

Chapter 3: Introduction to SQL

Chapter 4: Intermediate SQL

Chapter 5: Advanced SQL

Chapter 6: Formal Relational Query Languages

Part 2: Database Design

Chapter 7: Database Design: The E-R Approach

Chapter 8: Relational Database Design

Chapter 9: Application Design

Part 3: Data storage and querying

Chapter 10: Storage and File Structure

Chapter 11: Indexing and Hashing

Chapter 12: Query Processing

Chapter 13: Query Optimization

Part 4: Transaction management

Chapter 14: Transactions

Chapter 15: Concurrency control

Chapter 16: Recovery System

Part 5: System Architecture

Chapter 17: Database System Architectures

Chapter 18: Parallel Databases

Chapter 19: Distributed Databases

Part 6: Data Warehousing, Mining, and IR

Chapter 20: Data Mining

Chapter 21: Information Retrieval

Part 7: Specialty Databases

Chapter 22: Object-Based Databases

Chapter 23: XML

Part 8: Advanced Topics

Chapter 24: Advanced Application Development

Chapter 25: Advanced Data Types

Chapter 26: Advanced Transaction Processing

Part 9: Case studies

Chapter 27: PostgreSQL

Chapter 28: Oracle

Chapter 29: IBM DB2 Universal Database

Chapter 30: Microsoft SQL Server

Online Appendices

Appendix A: Detailed University Schema

Appendix B: Advanced Relational Database Model

Appendix C: Other Relational Query Languages

Appendix D: Network Model

Appendix E: Hierarchical Model