4190.301; Spring 2008

Prof. Sang-goo Lee

(13:00pm: Mon & Wed: Room 302-208)

INTRODUCTION TO DATABASE SYSTEMS

Syllabus

- Text Book

  Database System Concepts, 5th Edition, A. Silberschatz, H. F. Korth, and S. Sudarshan, McGraw Hill, 2006.

- Reference
  - *Database Systems,* Atzeni, et al, McGraw Hill, 2000.
  - *데이타베이스 시스템,* 이 상구 외, 영지문화사, 2001.
  - *데이터베이스 이론과 실전사이-Oracle 9i 중심*, 심준호, 이한출판사, 2002.

- Lecture Notes
  - will be posted before class at http://europa.snu.ac.kr
  - username & password required
  - Please use only for personal use

- Exams (tentative dates)
  - Exam 1: 4/7 (Mon)
  - Exam 2: 5/19 (Mon)
  - Exam 3: 6/14 (Sat, 13:00)

- Term Project
  - 2~3 programming assignments
  - To be announced later

- Grades
  - Exams 1, 2, & 3: 20% each
  - Term Projects: 30% total
  - Quizzes, Assignments, and others: 10%

**** A score of 0 in**
• any one of your term projects,
• any one of the exams, or
• more than 50% of your assignments/quizzes
will result in F.

What is a Database?

- Registration info of 40,000 students
  - For each student: 50 courses, term, grade, … => 10KB
  - 10KB * 40,000 = 400MB
  - Others: library, health center, S-card, …
- National college aptitude test
  - 2005: 586,600 students;
2001: 872,300 students
  - Personal data, answer lists, scores, ranks …
  - 8KB * 586600 = 5GB (10^9)
- Telco call data
  - Over 3,6M subscribers
  - time, number, station, …
  - 36M * 60B * 5calls/day * 365days/year = 4TB (10^{12})
  - China: 500M subscribers
- The World Wide Web
  - Wikipedia: 1.74 billion words in 7.5 million articles in 250 languages (English Wiki: 250,000+)
  - Total number of web pages: 15~30 billion (2007)
  - Index of 10 billion pages: 10G * 1KB = 10TB

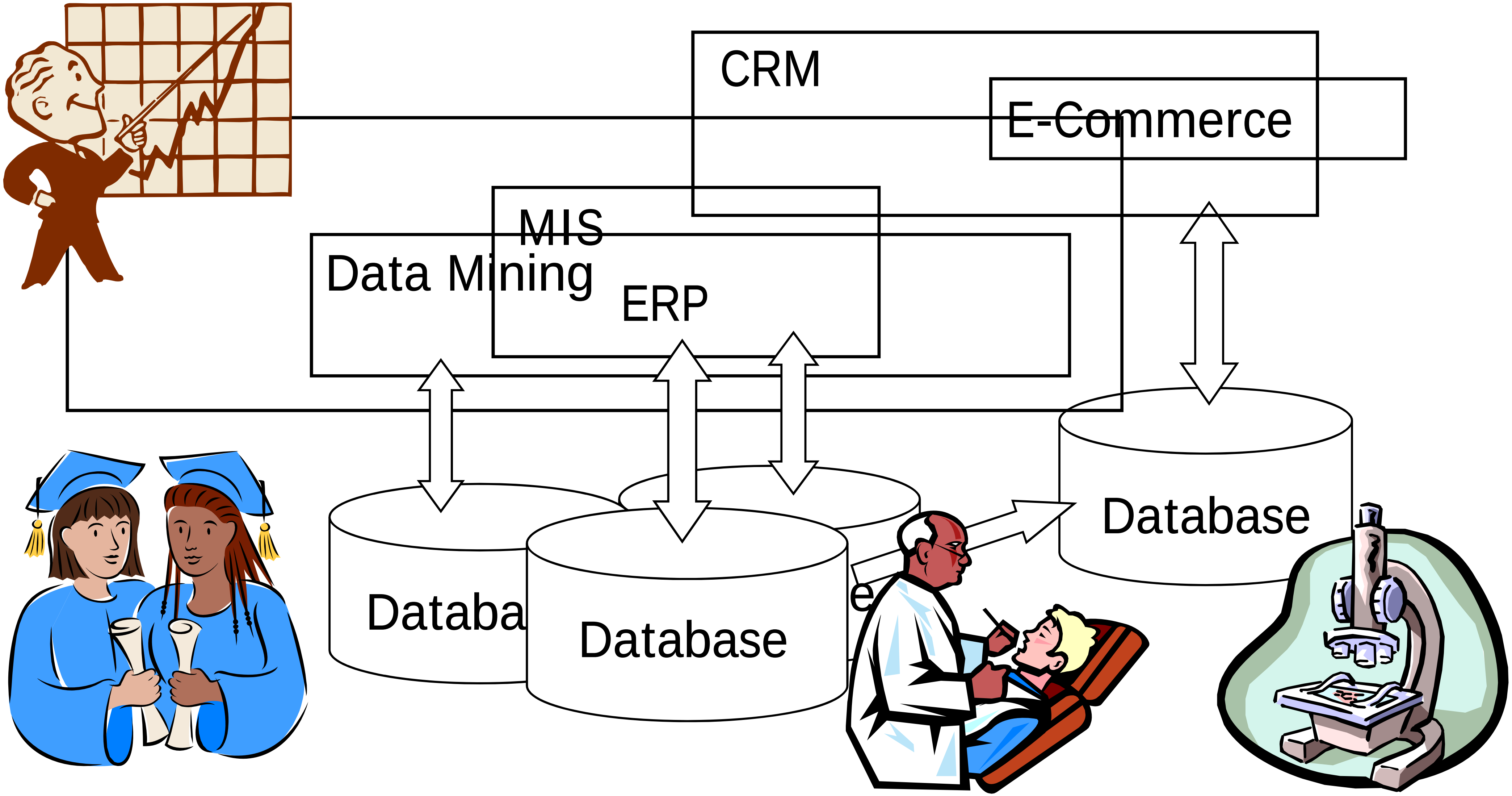
What do we do with it?

- Search
  - Math score for student no. 1234567
    - 586,000 * 5 = 2.9 M records
    - 12ms to fetch a record and check content

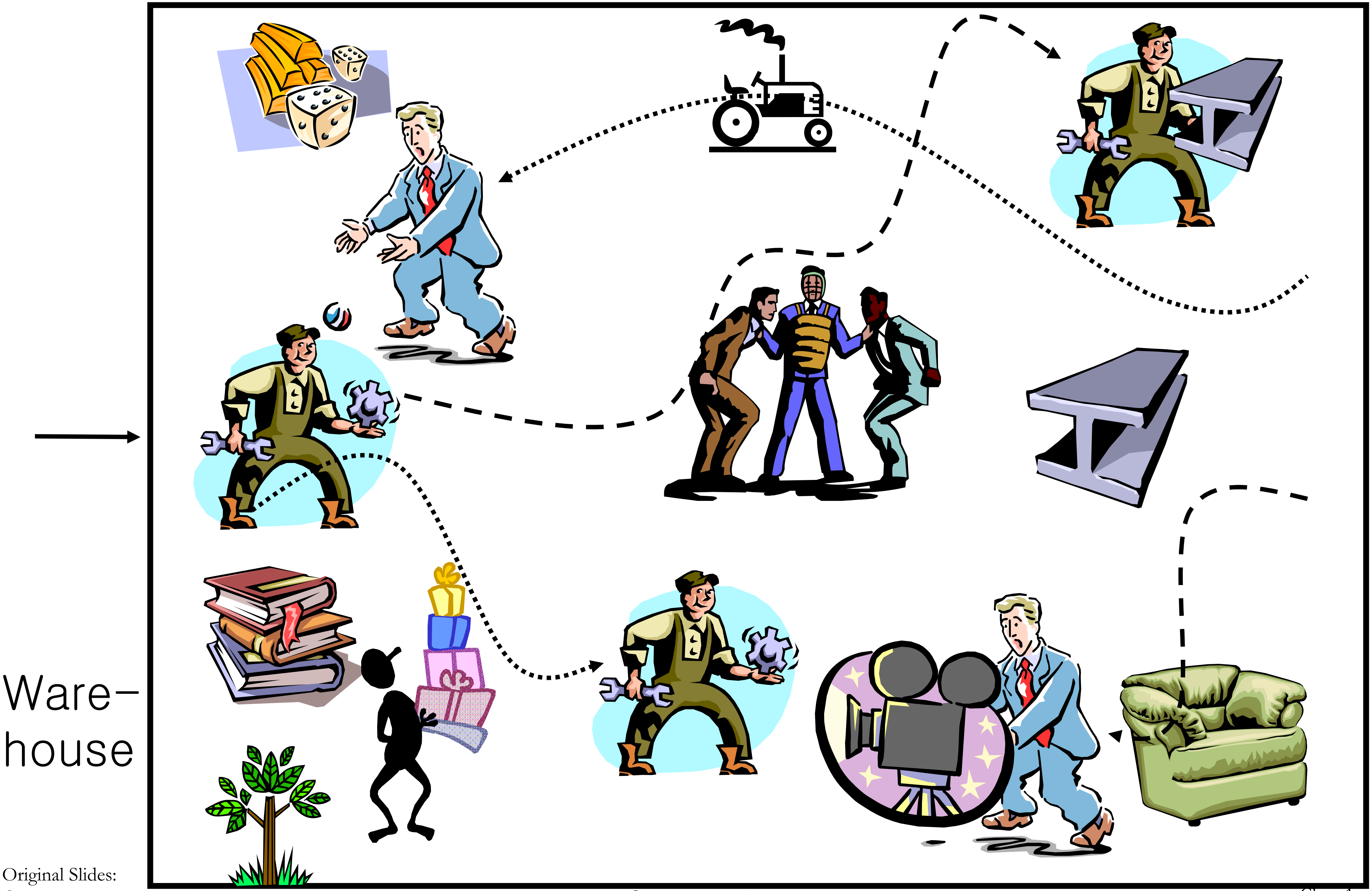
    ⇒2.9M * 12ms = 34.8Kseconds = over 9 hours
- Summarize and Mine for more information
  - Population by age and city
  - Gene patterns for diseases
  - Purchase patterns of classes of credit card customers
- Feed data to applications
  - Web sites
  - Telephone switching
  - Video stream

What do we do with it? (cont.)

- Most (all?) computing applications use some type of a database



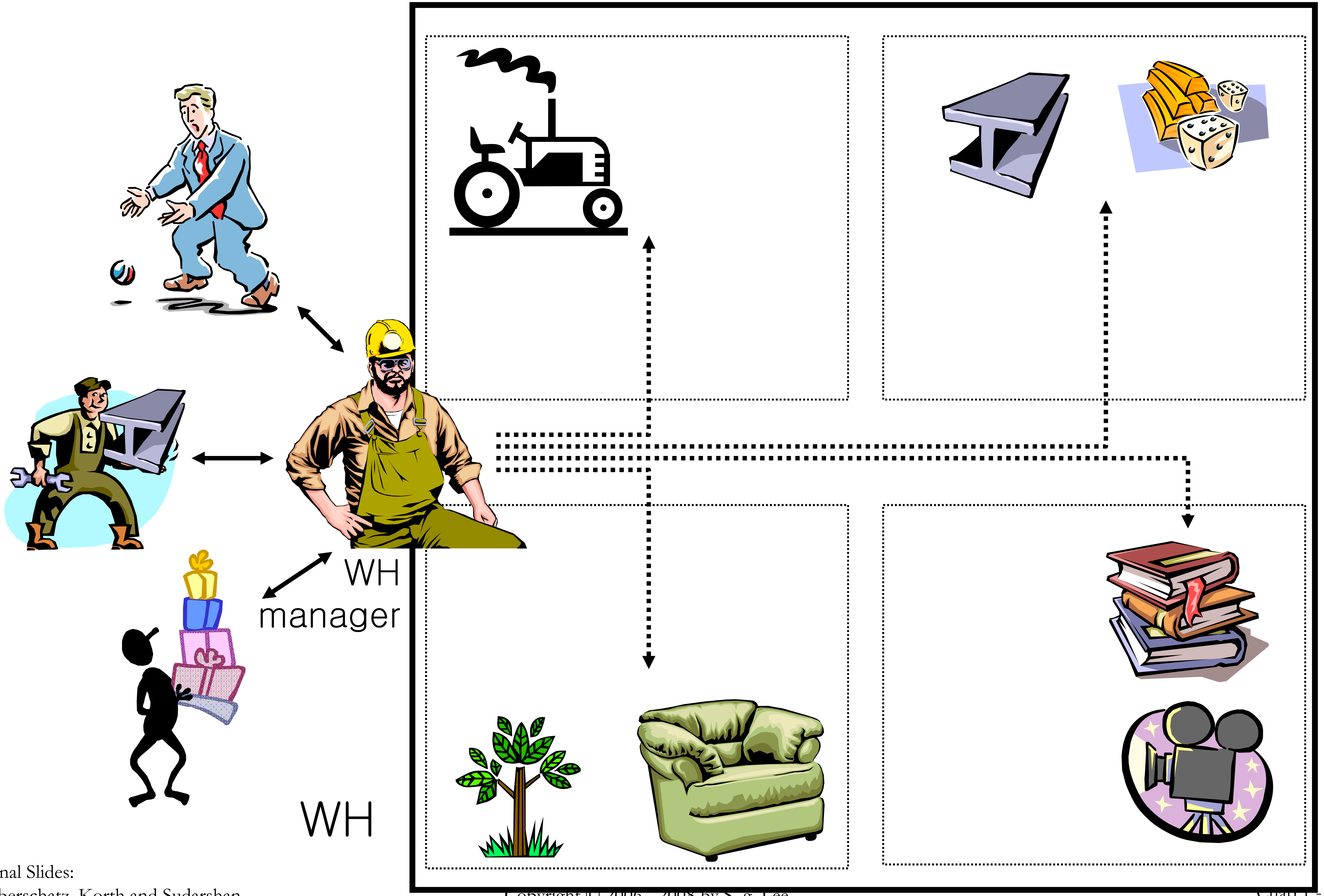
Database Management System (DBMS)



Original Slides:
© Silberschatz, Korth and Sudarshan

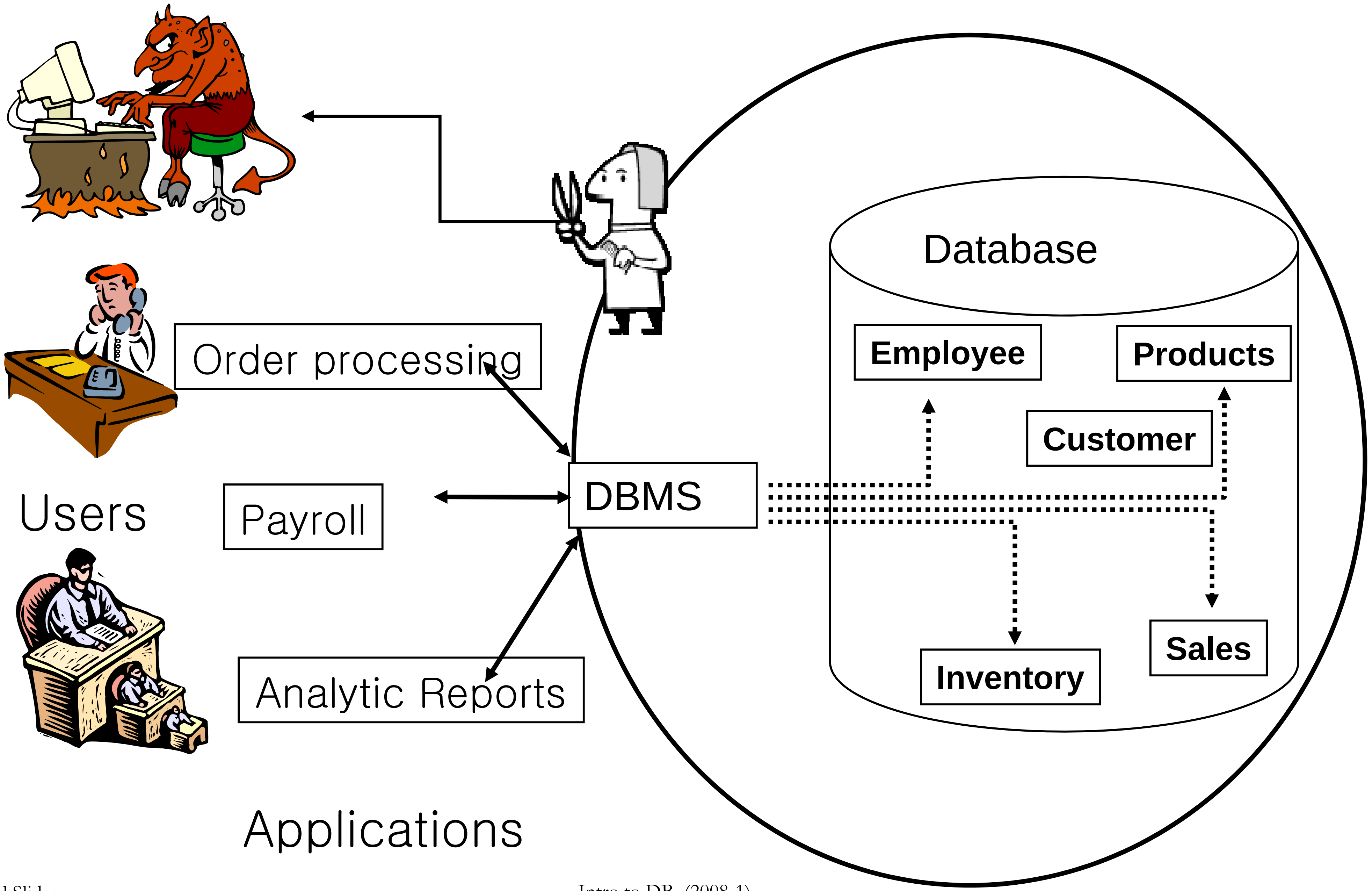
Copyright © 2006 - 2008 by S.-g. Lee

Database Management System (DBMS)



Original Slides:
© Silberschatz, Korth and Sudarshan

Database Management System (DBMS)



Original Slides:
© Silberschatz, Korth and Sudarshan

CHAPTER 1. INTRODUCTION

Contents

- Data & Database
- Database Management Systems
- View of Data
- Data Models
- Data Languages
- Database Users
- Transaction Management
- Storage Management
- Data Mining & Analysis
- Overall System Structure

Data & Database

- Data
  - A formal description of
    - an entity, event, phenomena, or idea
    - that is worth recording
- Database
  - An integrated collection of
  - persistent data
  - representing the information of interest
  - for various programs that compose the computerized information system of an organization.
  - Data are separated from the programs that use them

Database Management System (DBMS)

- Set of programs to access the data
- DBMS provides an environment that is both *convenient* and *efficient* to use.
- Database Applications (Information Systems):
  - Banking: all transactions
  - Airlines: reservations, schedules
  - Universities: registration, grades
  - Sales: customers, products, purchases
  - Manufacturing: production, inventory, orders, supply chain
  - Human resources: employee records, salaries, tax deductions
- Databases touch all aspects of our lives
- Commercial Systems
  - DB2, Oracle, Informix, BADA, MS SQL Server, Sybase,
  - dBase, FoxPro, Access

File Systems

- File System
  - Part of OS
  - Stores programs, data, documents, or anything
  - (in disk)
- In the early days, database applications were built on top of file systems
- Drawbacks of using file systems to store data:
  - Data redundancy and inconsistency
    - Multiple file formats, duplication of information in different files
  - Difficulty in accessing data
    - Need to write a new program to carry out each new task
  - Data isolation — multiple files and formats
  - Integrity problems
    - Integrity constraints (e.g. account balance > 0) become part of program code
    - Hard to add new constraints or change existing ones

File Systems (cont.)

- Drawbacks of using file systems (cont.)
  - Atomicity of updates
    - Failures may leave database in an inconsistent state with partial updates carried out
    - E.g. transfer of funds from one account to another should either complete or not happen at all
  - Concurrent access by multiple users
    - Concurrent accessed needed for performance
    - Uncontrolled concurrent accesses can lead to inconsistencies
      - E.g. two people reading a balance and updating it at the same time
  - Security problems
- Database systems offer solutions to all the above problems

Levels of Abstraction

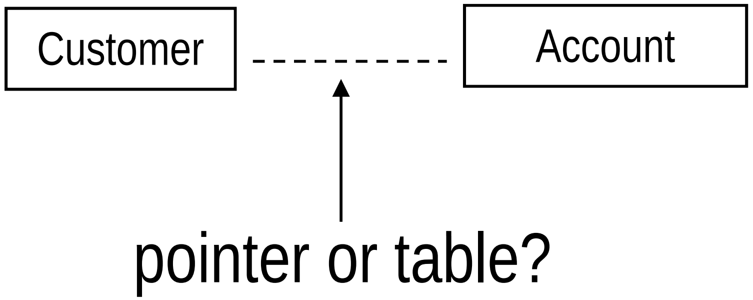
- Physical level describes how a record (e.g., customer) is stored in a physical device.
- Logical level: describes data stored in database, and the relationships among the data.

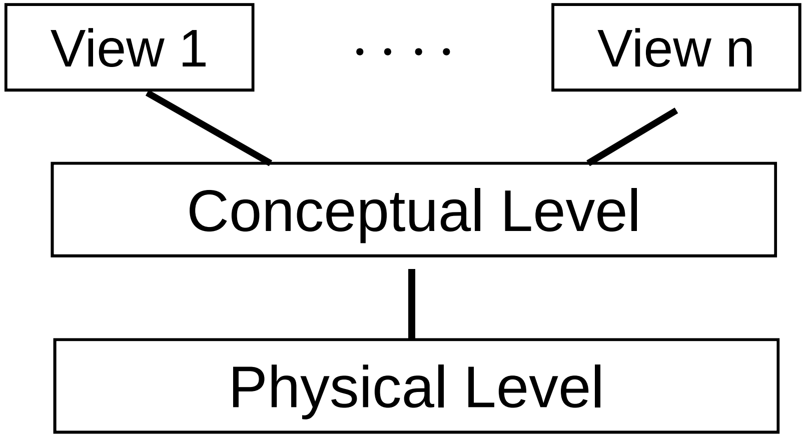
```
type customer = record
                    name : string;
                    street : string;
                    city : integer;
                end;
```

- View level: application programs hide details of data types. Views can also hide information (e.g., salary) for security purposes.

Data Independence

- ability to modify a schema in one level without affecting a schema definition in the next higher level
- physical data independence:
  - physical level - conceptual level
- logical data independence:
  - conceptual level - view level





Instances and Schemas

- Similar to types and variables in programming languages
- **Schema** – the logical structure of the database
  - e.g., the database consists of information about a set of customers and accounts and the relationship between them)
  - Analogous to type information of a variable in a program
  - **Physical schema**: database design at the physical level
  - **Logical schema**: database design at the logical level
- **Instance** – the actual content of the database at a particular point in time
  - Analogous to the value of a variable

- Scheme (schema)
  - the skeletal structure of the data content

Customer

Name	Address	Telephone

Account

No.	Type	Balance

- Instance
  - the actual content of the data at a given time
  - database status

2008/2/20/12:00
Customer

HS Kim	Seoul	323-3232
KS Lee	Busan	323-5454
PL Park	Seoul	553-3235
…		

2008/3/3/12:00
Customer

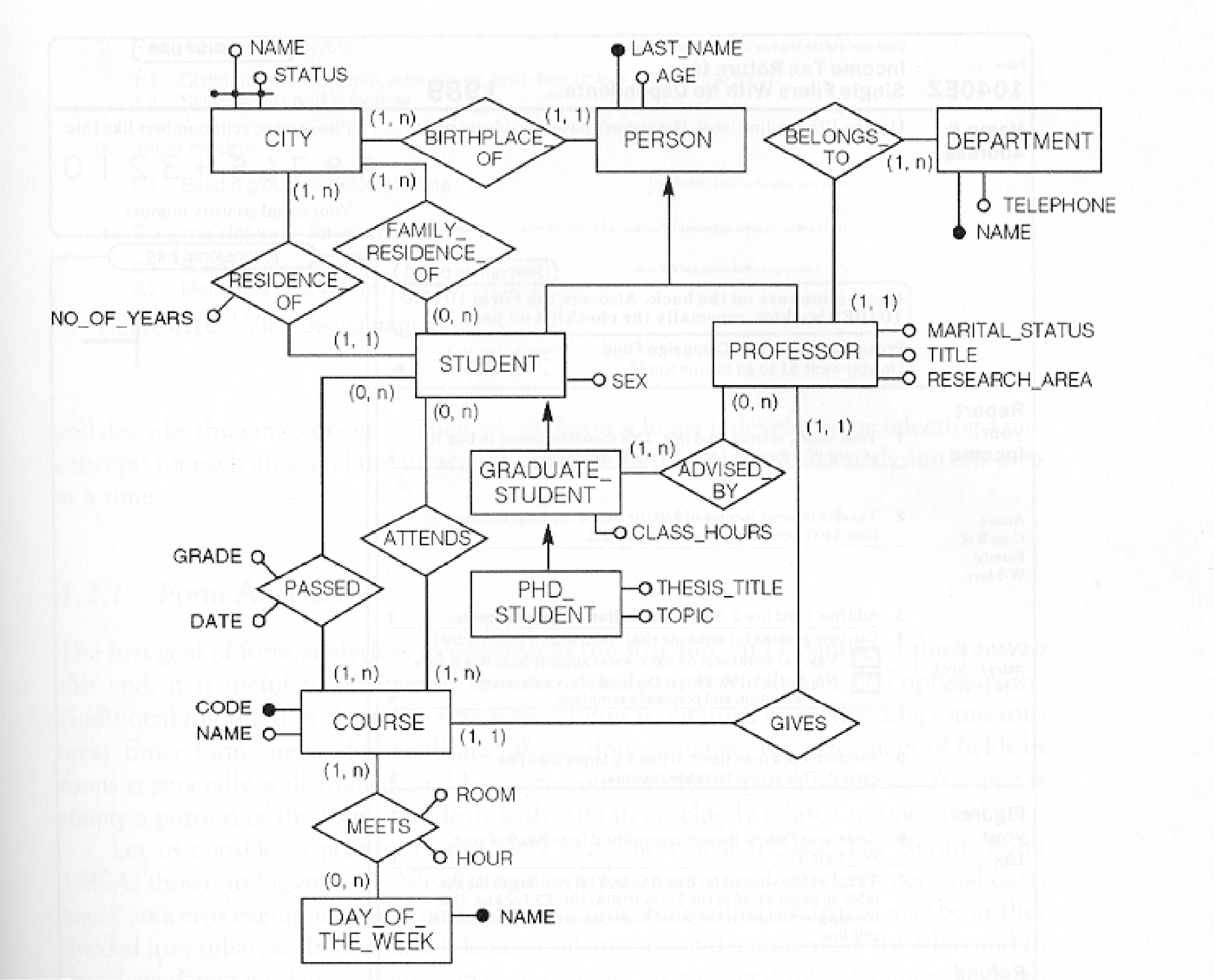
HS Kim	Suwon	323-3232
KS Lee	Busan	323-5454
MH Choi	Seoul	553-3235
KH Na	Yongin	545-5488
…		

Data Models

- A collection of tools for describing
  - data
  - data relationships
  - data semantics
  - data constraints
- Entity-Relationship model
- Relational model
- Other models:
  - object-oriented model
  - semi-structured data models
  - Older models: network model and hierarchical model

Entity-Relationship Model

Example of schema in the entity-relationship model

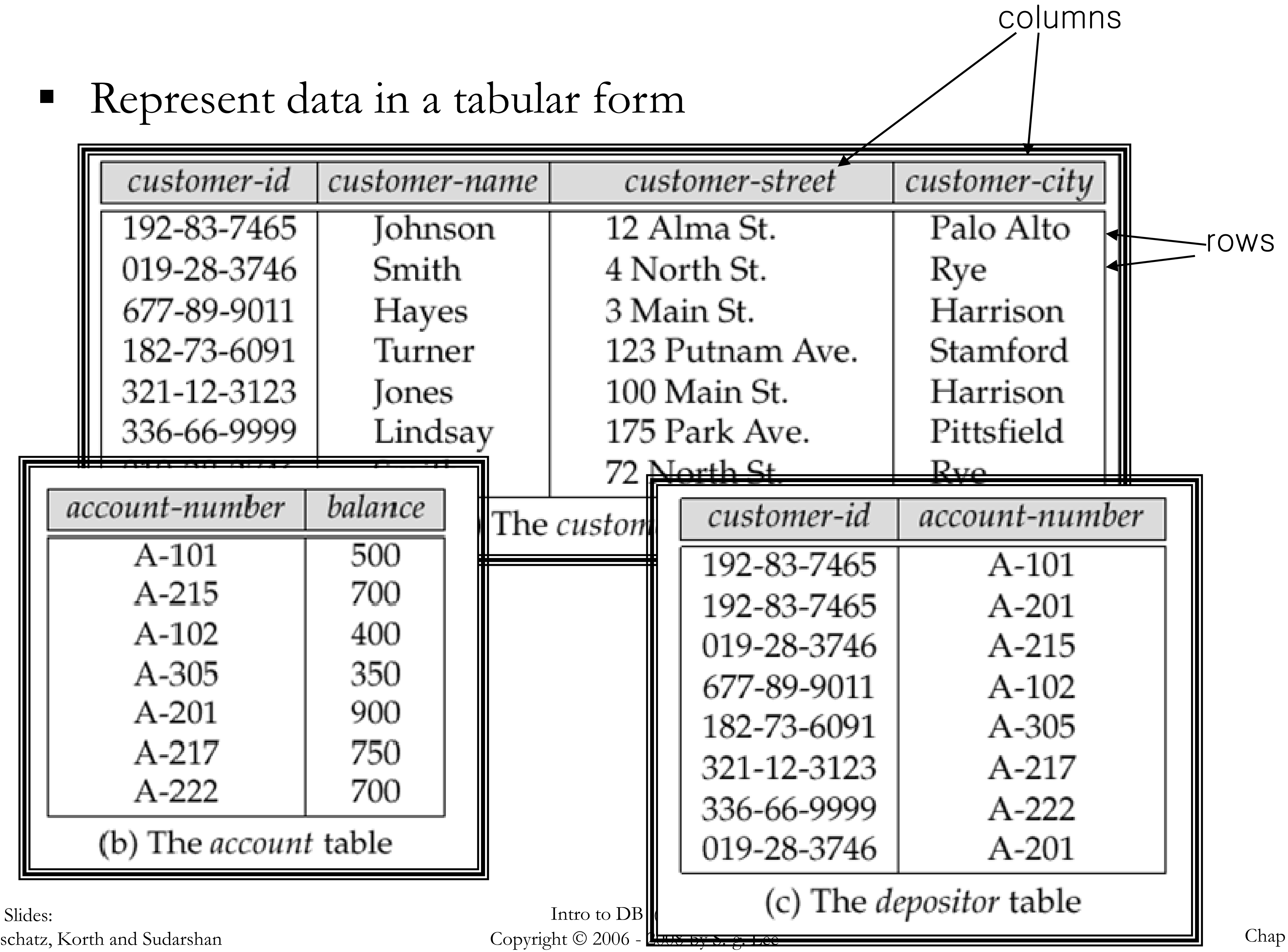


Entity Relationship Model (cont.)

- E-R model of real world
  - Entities (objects)
    - E.g. customers, accounts, bank branch
  - Relationships between entities
    - E.g. Account A-101 is held by customer Johnson
    - Relationship set *depositor* associates customers with accounts
- Widely used for database design
  - Database design in E-R model usually converted to design in the relational model (coming up next) which is used for storage and processing

Relational Model

- Represent data in a tabular form


columns
rows

customer-id	*customer-name*	*customer-street*	*customer-city*
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
[illegible]	[illegible]	72 North St.	Rye

The *custom*

account-number	*balance*
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

customer-id	*account-number*
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

Database Languages

- Data Definition Language (DDL)
  - Used for defining DB Schema
    - create table
    - drop column
- Data Manipulation Language (DML)
  - Used for operating the data in the DB (DB instance)
    - Retrieve
    - Insert
    - Delete
    - Change
- Query
  - a statement requesting the retrieval of information
  - query language: part of DML
  - sometimes "query language = DML"

SQL

- The most widely used 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
```

Database Users

- Users are differentiated by the way they expect to interact with the system
- Application programmers – interact with system through DML calls
- Sophisticated users – form requests in a database query language
- Specialized users – write specialized database applications that do not fit into the traditional data processing framework
- Naïve users – invoke one of the permanent application programs that have been written previously
    - E.g. people accessing database over the web, bank tellers, clerical staff

Database Administrator

- Coordinates all the activities of the database system; the database administrator has a good understanding of the enterprise's information resources and needs.
- Database administrator's duties include:
  - Schema definition
  - Storage structure and access method definition
  - Schema and physical organization modification
  - Granting user authority to access the database
  - Specifying integrity constraints
  - Acting as liaison with users
  - Monitoring performance and responding to changes in requirements

Transaction Management

- Transaction
  - a collection of operations that performs a single logical function in a database application
  - programmer is responsible for writing "correct" transactions
- DBMS must ensure the *atomicity* and *durability* of each transaction
  - atomicity : all-or-nothing
  - durability : effect should be persistent

Storage Management

- DBMS must effectively and efficiently manage storage (disk) space

- Storage manager
  - a program module
  - that provides the interface
  - between the low-level data stored in the database
  - and the application programs and queries submitted to the system

Data Mining & Analysis

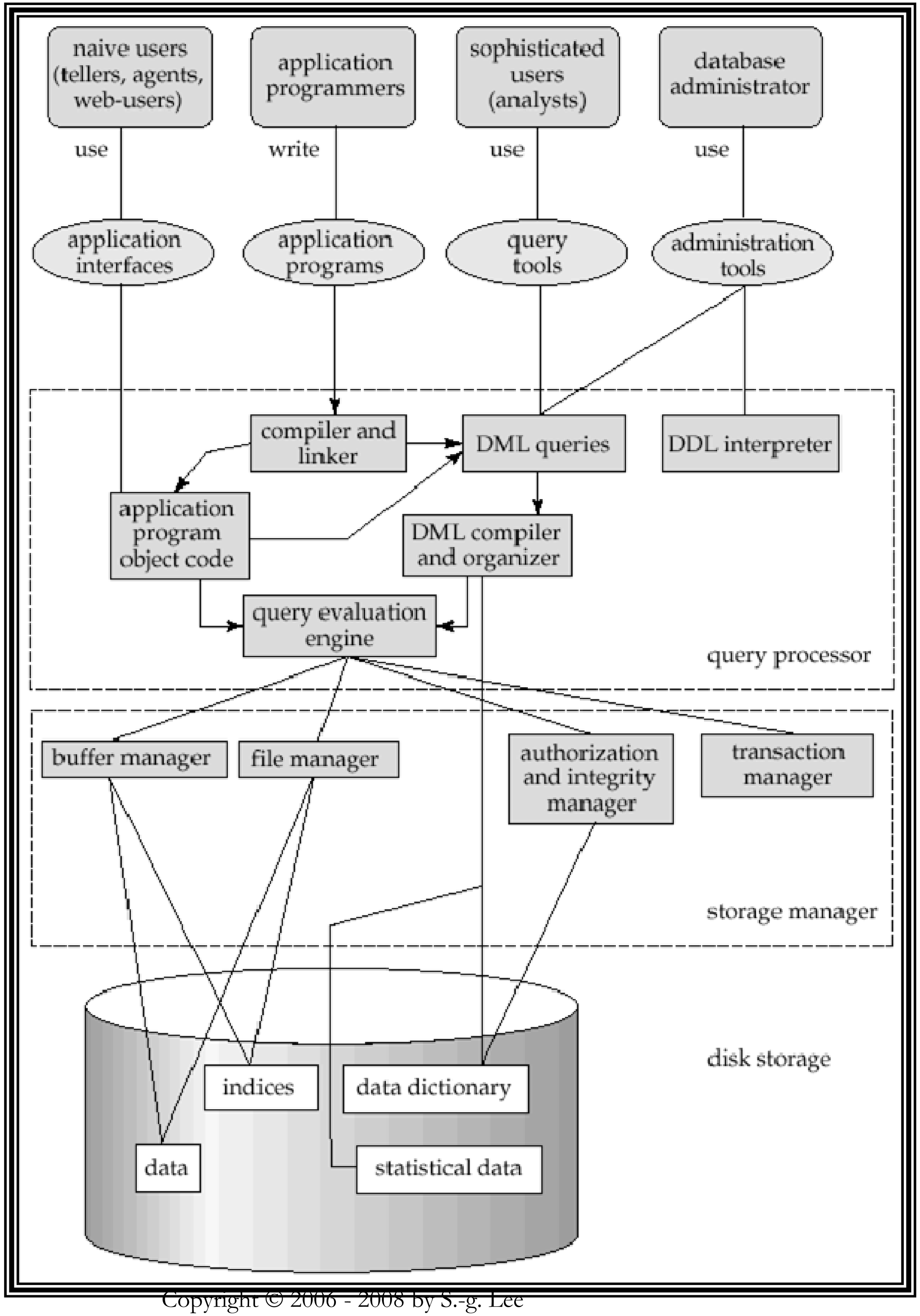
- The challenge is

  "getting information out"!

  - Knowledge Discovery in Databases
  - Extract information from the database

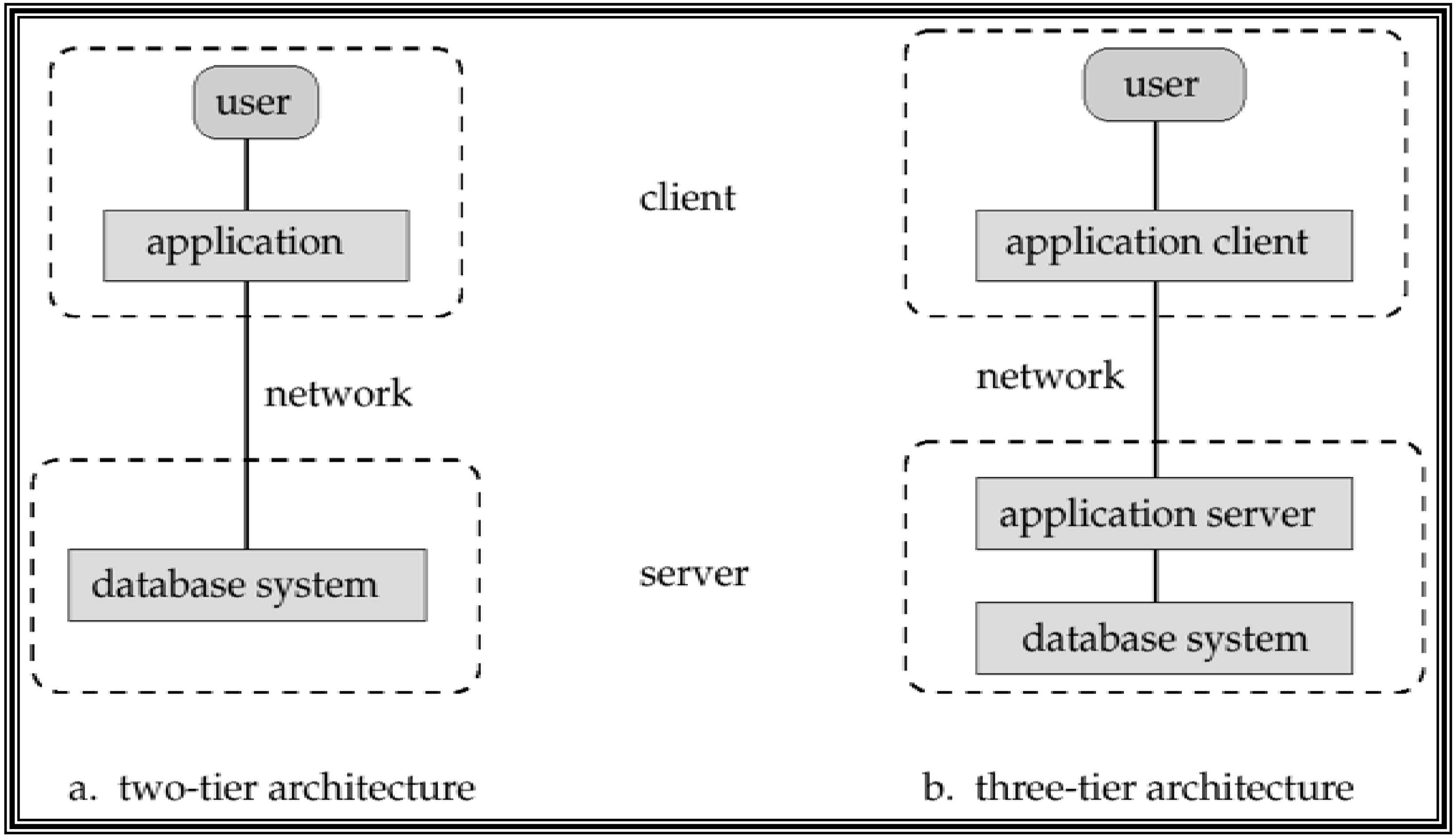
- Information retrieval
  - Textual data files (documents)
  - Find the most relevant document(s) for the given information need (query)

Overall System Structure



Original Slides:
© Silberschatz, Korth and Sudarshan

Application Architecture



a. two-tier architecture

b. three-tier architecture

- **Two-tier architecture**: E.g. client programs using ODBC/JDBC to communicate with a database
- **Three-tier architecture**: E.g. web-based applications, and applications built using "middleware"

END OF CHAPTER 1