CHAPTER 1. INTRODUCTION

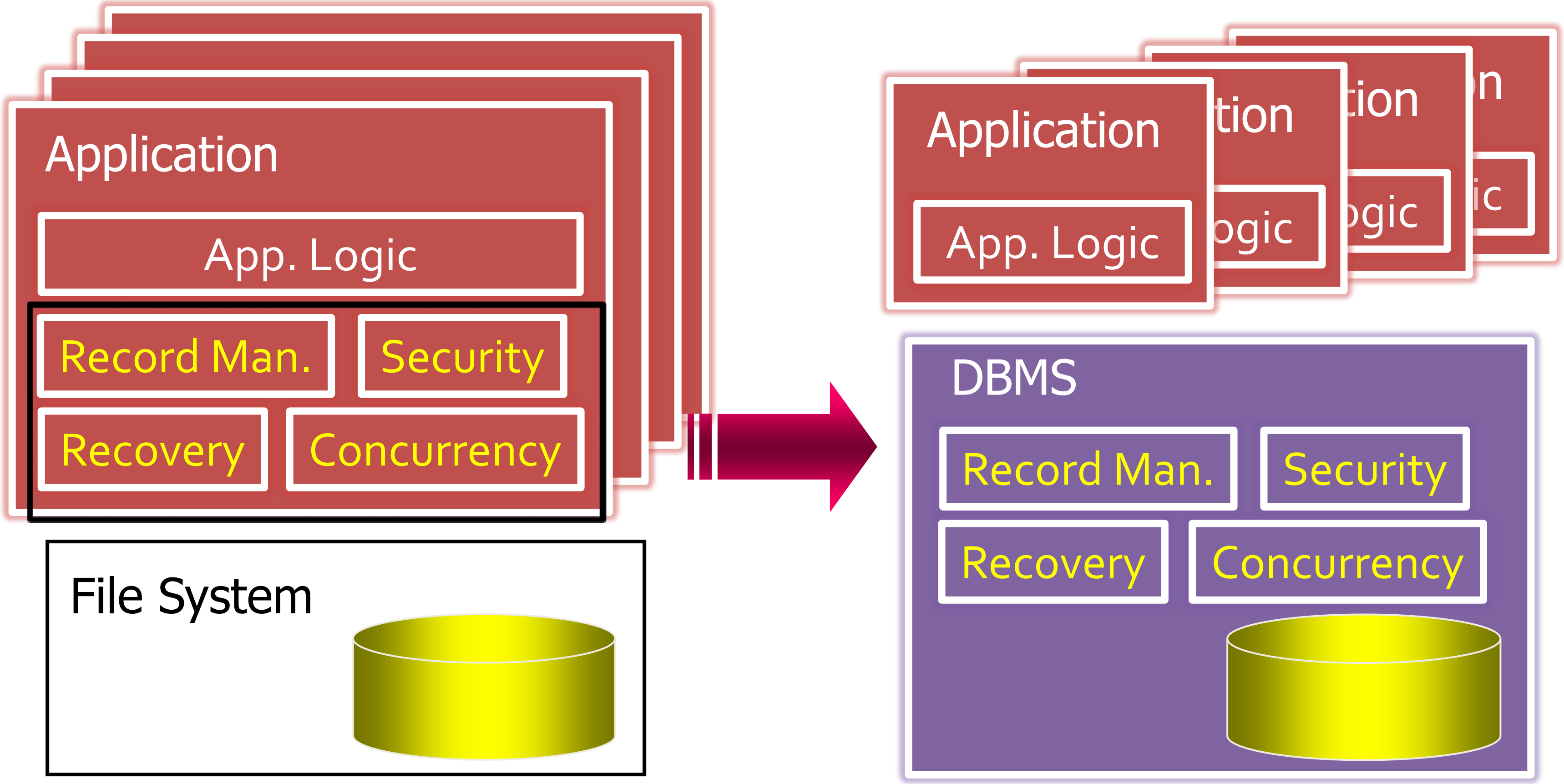
Chapter 1: Introduction

- Data & Database
- Database Management Systems
- File Systems
- View of Data
- Data Models
- Data Languages
- Database Users
- Overall System Structure

What is a Database?

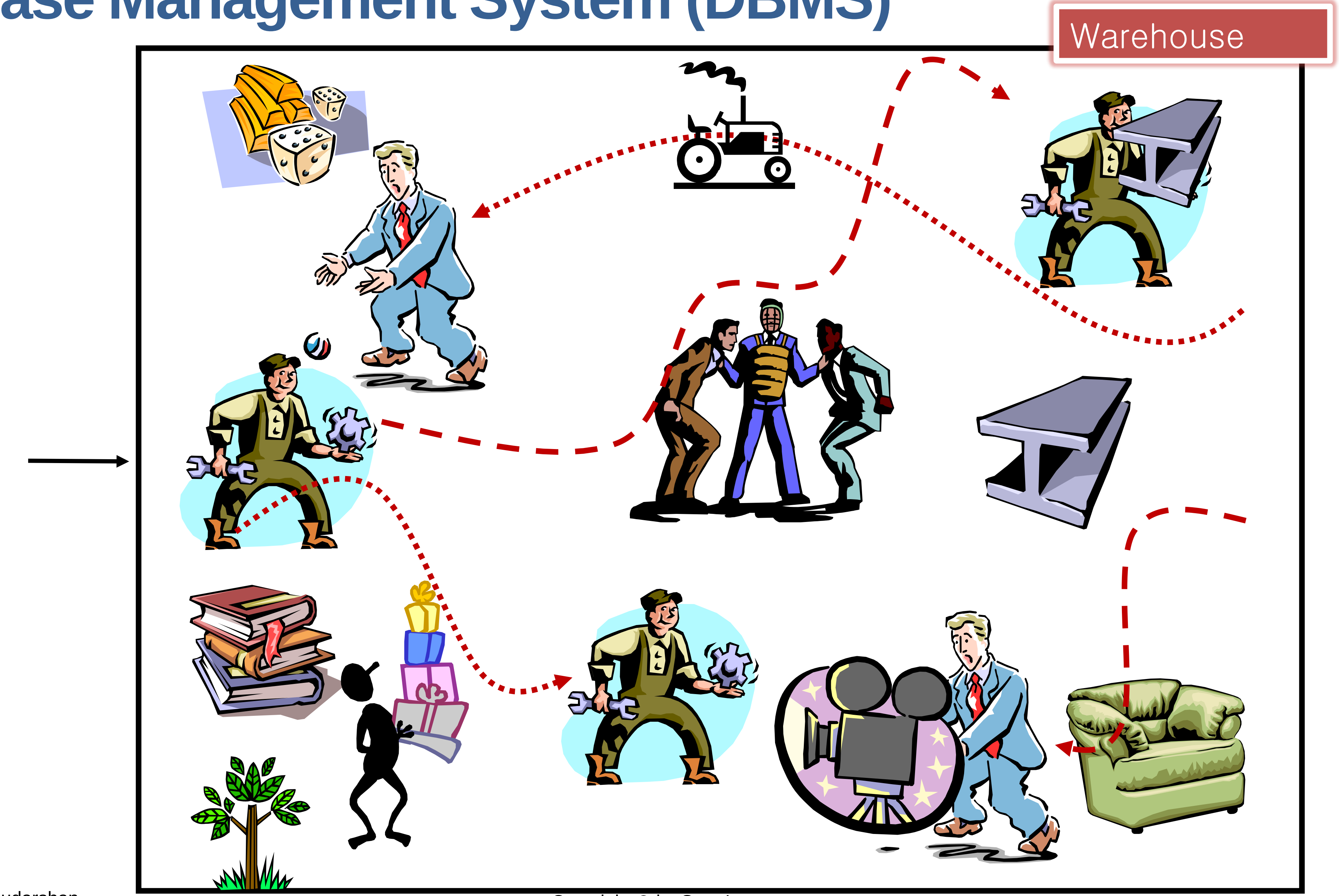
- **Data**
  - A formal description of
    - an entity, event, phenomena, or idea
    - that is worth recording
- **Database**
  - An integrated collection of
  - persistent data
  - representing the information of interest
  - for various programs that compose the computerized information system of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

DBMS



- Database Management System
  - A collection of program modules that store, process, and manage data
  - Abstraction

Database Management System (DBMS)



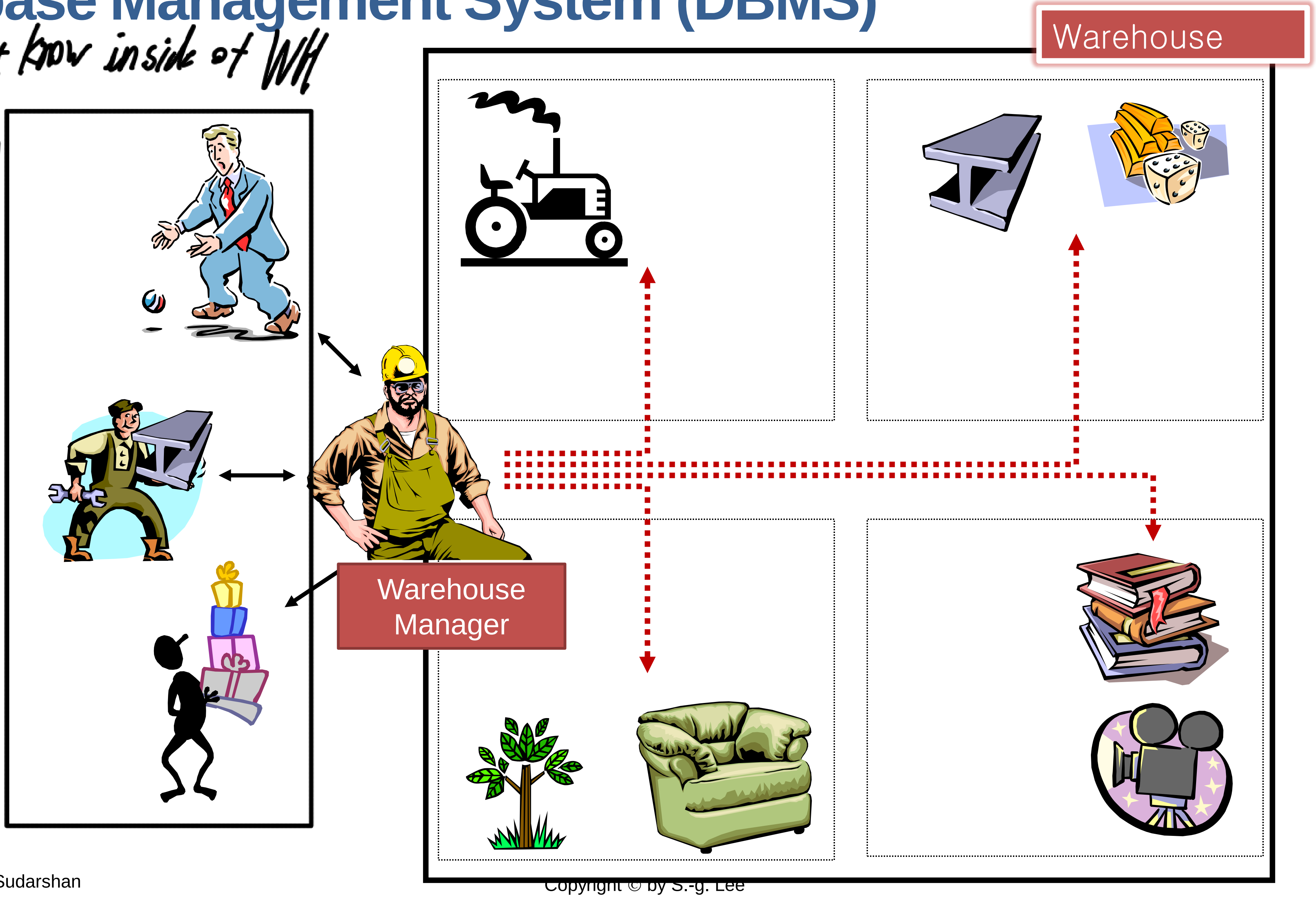
Original Slides:
© Silberschatz, Korth and Sudarshan

Copyright © by S.-g. Lee

Database Management System (DBMS)

Users don't know inside of WH

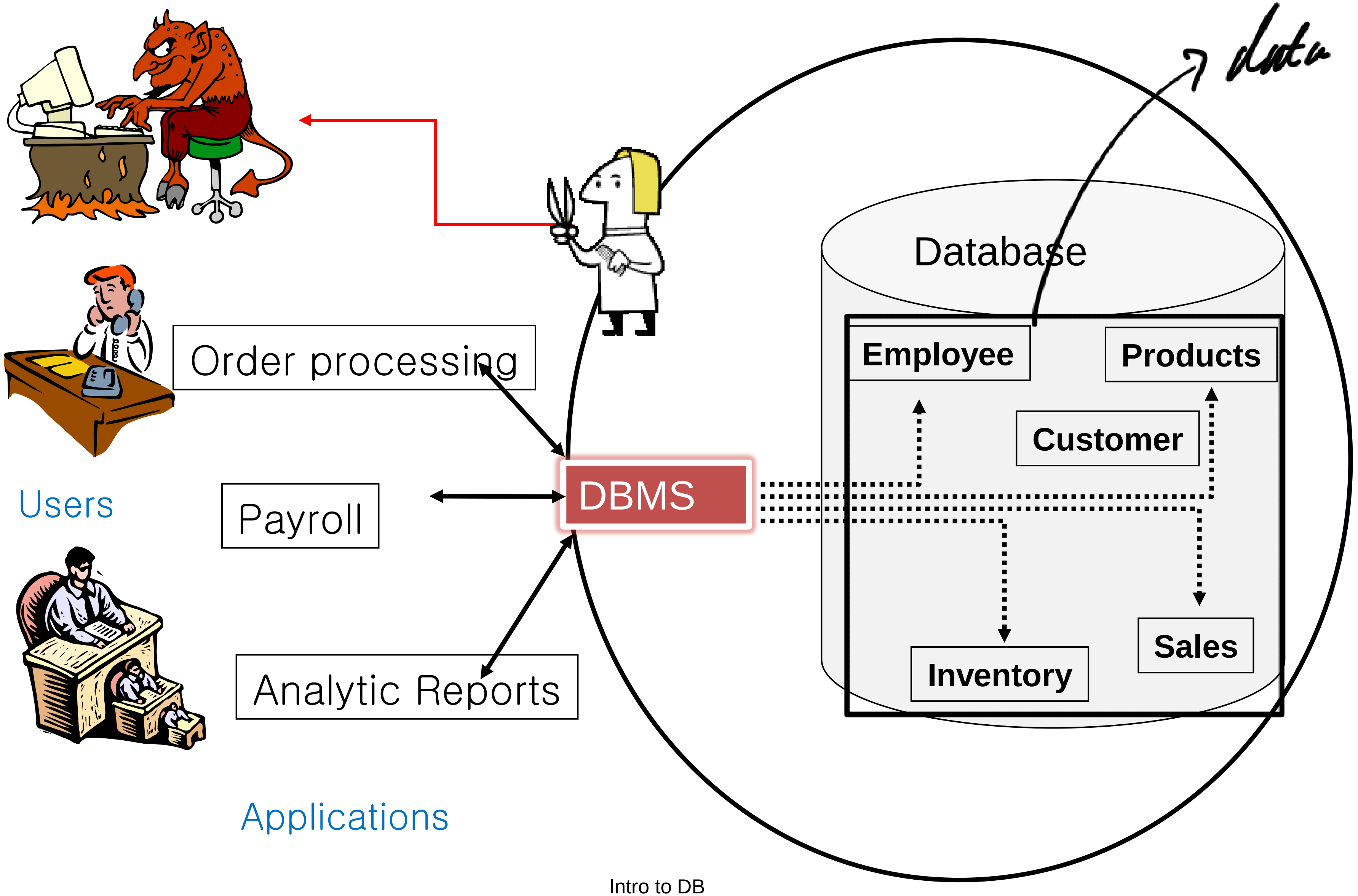
=> Abstraction



Warehouse

Warehouse Manager

Database Management System (DBMS)



Order processing

Payroll

Analytic Reports

Users

Applications

DBMS

Database

Employee

Products

Customer

Inventory

Sales

→ data fragment

Database Management System (DBMS)

- Set of programs to manage/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, MS SQL Server, MySQL, Hana
  - MS Access

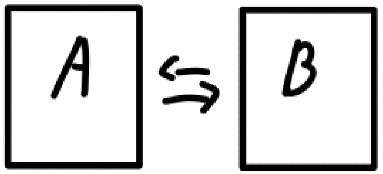
File Systems

extra function

- File System
  - Part of OS
  - Stores programs, data, documents, or anything
  - (in disk)

overlapped data for back up

- 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 → data change propagation : if one data changed, then all copy should reflect that
    - Multiple file formats, duplication of information in different files
  - Difficulty in accessing data : non-type safe of data
    - Need to write a new program to carry out each new task
  - Integrity problems : data error check (ex) age must over zero)
    - Integrity constraints (e.g. account balance > 0) become part of program code
    - Hard to add new constraints or change existing ones



these problem solved by
using DBMS

File Systems (cont.)

prevent dreadlock, race-condit :, Concurrence I/O to data, overwrite

- 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

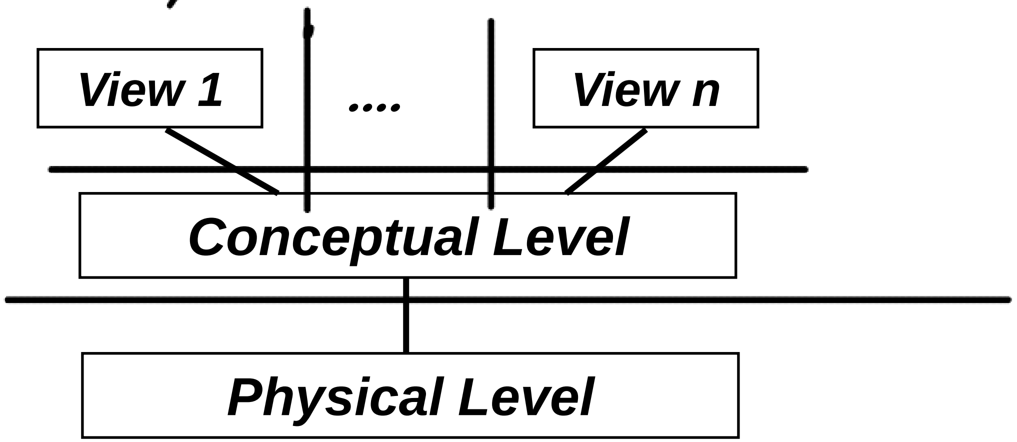
Levels of Abstraction

- **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.

View part of 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.

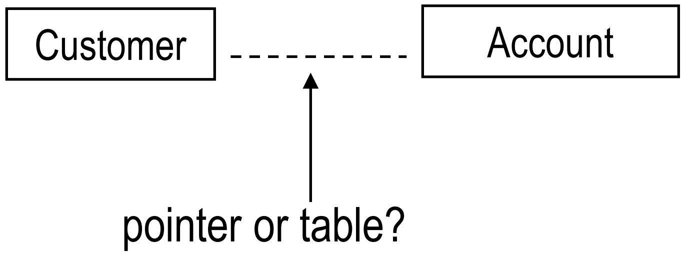
Data Independence

:each level after abstraction to upper level

- 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

| location, storing method

- 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

Instances and Schemas – Examples

- 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

2022/2/20/12:00
Customer

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

2022/3/3/12:00
Customer

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

Data Models

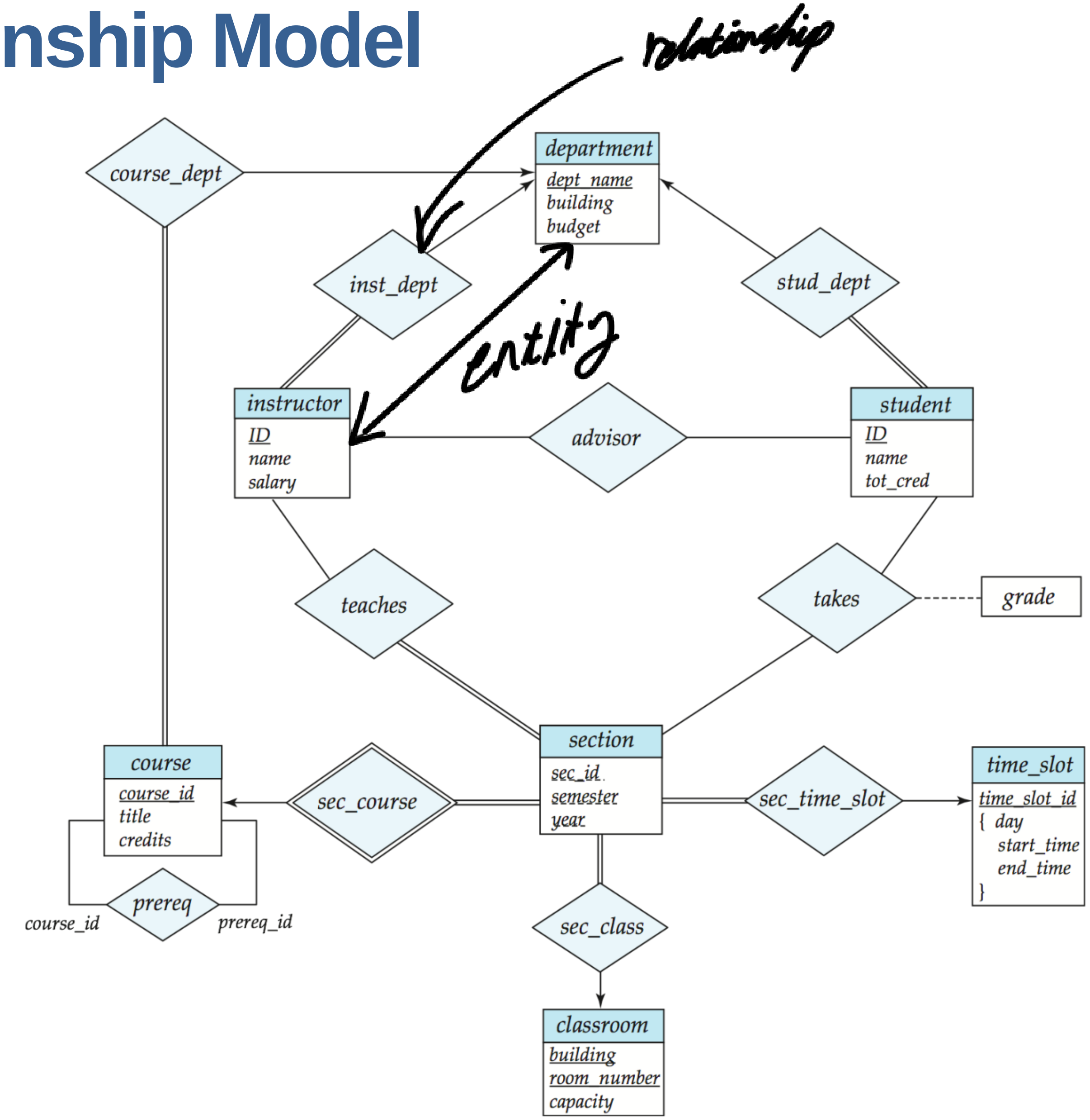
- A collection of tools for describing
  - data
  - data relationships
  - data semantics
  - data constraints

→ Schema of DB + relationship between instance

- Entity-Relationship model
- Relational model
- Other models:
  - object-oriented model
  - semi-structured data models
  - Older models: network model and hierarchical model

Entity-Relationship Model

(E-R 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

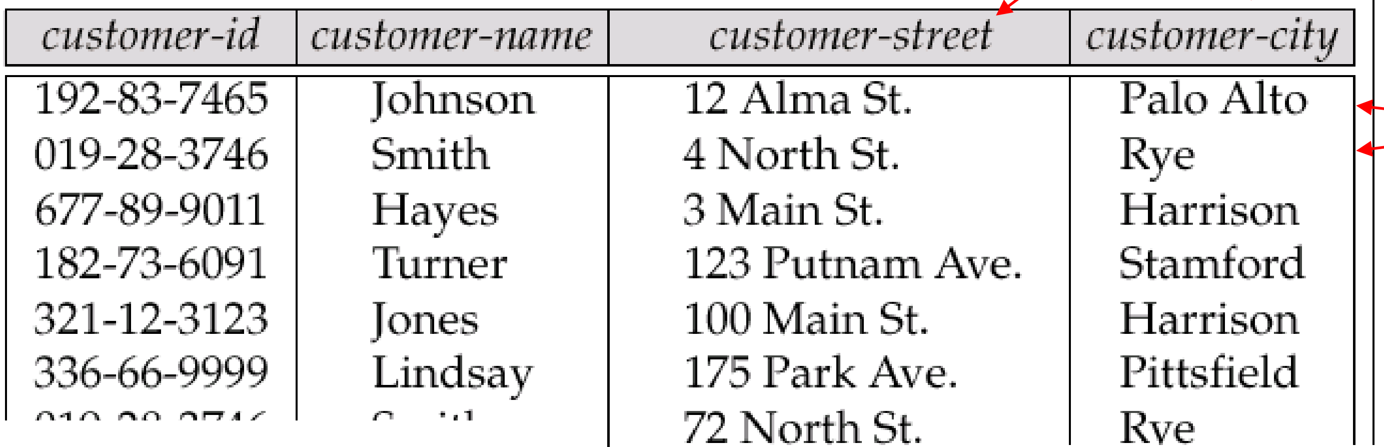
Relational Model (represent data as table)

- Represent data in a tabular form

columns

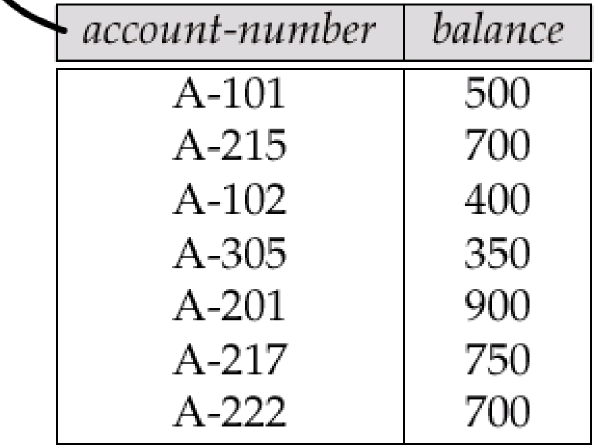
rows

attribute



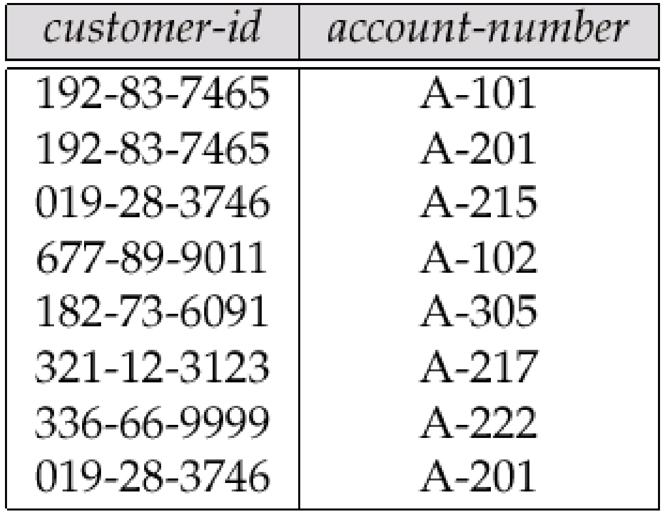
customer-id	customer-name	customer-street	customer-city
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746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

read

Original Slides:
© Silberschatz, Korth and Sudarshan

Intro to DB
Copyright © by S.-g. Lee

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

DISCUSSIONS – CHAPTER 1

Discussion 1-1

- List four applications you have used that most likely employed a database system to store persistent data.
  (other than the ones mentioned in class)

Search-history Log

Save user information of service

Discussion 1-2

- Explain the concept of *physical data independence*, and its importance in database systems.

— the ability to modify the physical schema without changing logical schema

⇒ Applications depend on the logical schema

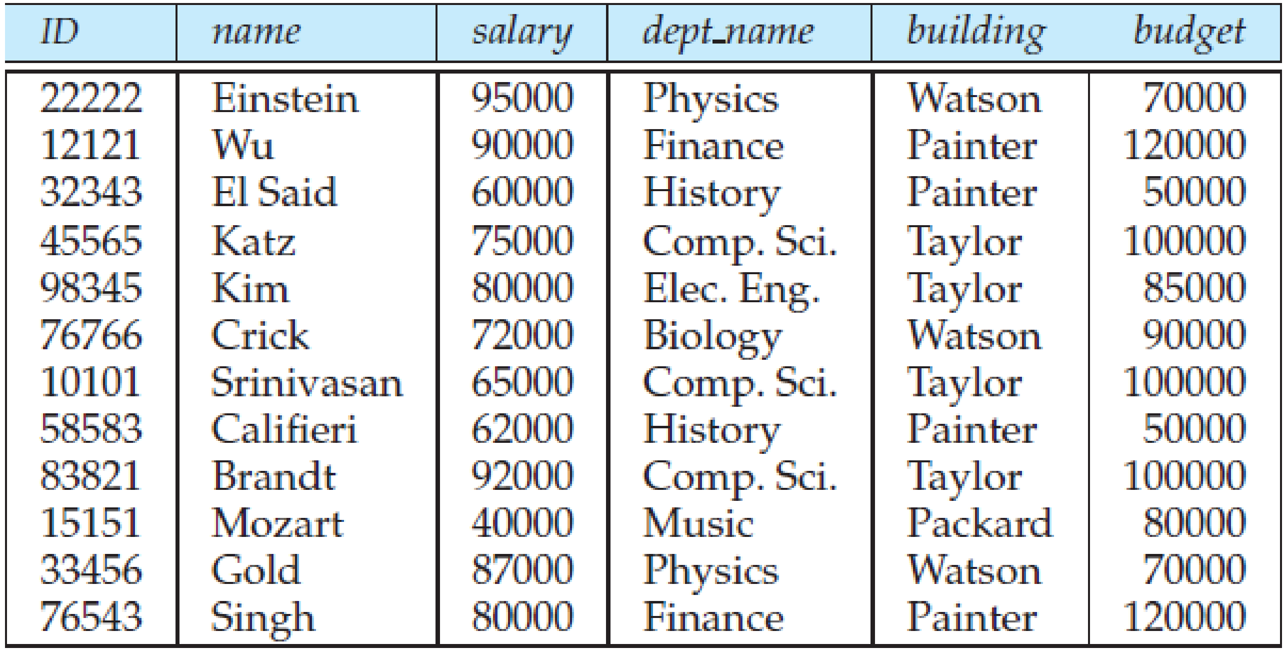
⇒ In general, the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others

Discussion 1-3

A. Describe the information content represented by the table in Figure 1.4.

B. Explain what problems may arise by this design.

분리



ID	*name*	*salary*	*dept_name*	*building*	*budget*
22222	Einstein	95000	Physics	Watson	70000
12121	Wu	90000	Finance	Painter	120000
32343	El Said	60000	History	Painter	50000
45565	Katz	75000	Comp. Sci.	Taylor	100000
98345	Kim	80000	Elec. Eng.	Taylor	85000
76766	Crick	72000	Biology	Watson	90000
10101	Srinivasan	65000	Comp. Sci.	Taylor	100000
58583	Califieri	62000	History	Painter	50000
83821	Brandt	92000	Comp. Sci.	Taylor	100000
15151	Mozart	40000	Music	Packard	80000
33456	Gold	87000	Physics	Watson	70000
76543	Singh	80000	Finance	Painter	120000

Figure 1.4 The *faculty* table.

Discussion 1-4

A. Describe at least 5 tables that might be used to store information in an online social communication system such as *KakaoTalk*.

B. What would be the top queries issued to these tables by the application?

Chatting Log, Friends list, Users List, Thread List,

Discussion 1-5

- If the data in the previous problem were kept in a file system (without the use of a DBMS), describe a situation where things may go wrong.
  - Data redundancy and inconsistency
  - Difficulty in accessing data
  - Data isolation
  - Integrity problems
  - Atomicity of Updates
  - Concurrently access by multiple users
  - security problem

Discussion 1-6

- Keyword queries used in *Web search* are quite different from database queries. List key differences between the two, in terms of the way the queries are specified, and in terms of what is the result of a query.

Discussion 1-7

- How would you define *big data*?
- What *values* does (big) data bring?

Original Slides:
© Silberschatz, Korth and Sudarshan

END OF CHAPTER 1