Principles of Database Systems

Introduction

主讲教师：薛昕惟

办公室：信息楼320B

xuexinwei@dlut.edu.cn

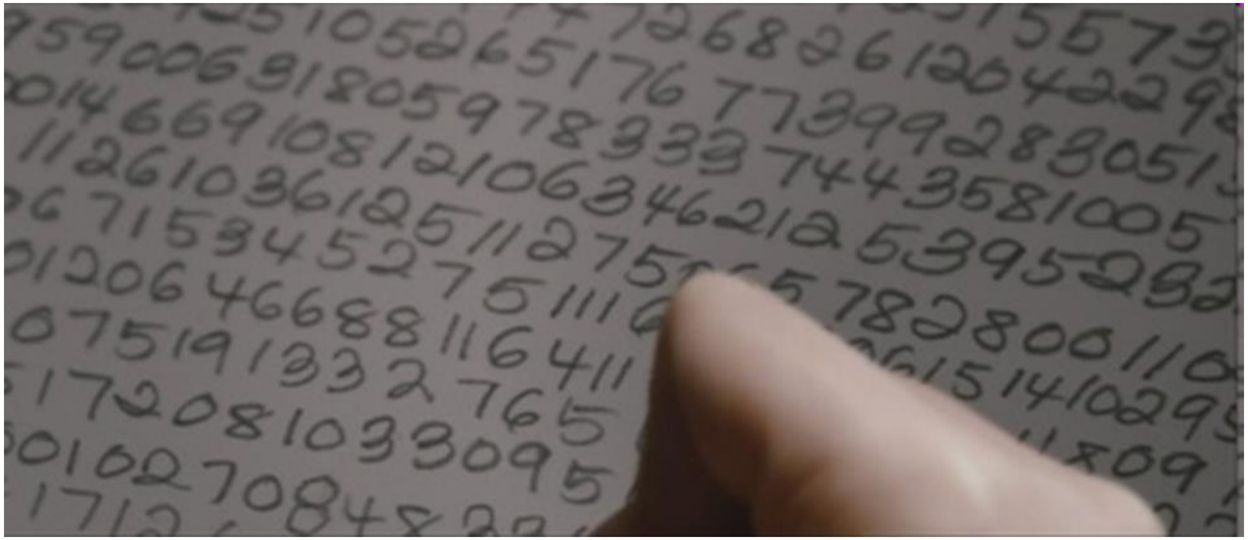
Basic Concepts

Basic Concepts

- Data
  - **Values** of qualitative or quantitative variables, belonging to a set of items (wikipedia)
  - Data as an **abstract concept** (抽象概念) can be viewed as the lowest level of abstraction from which **information** and then **knowledge** are derived (wikipedia)

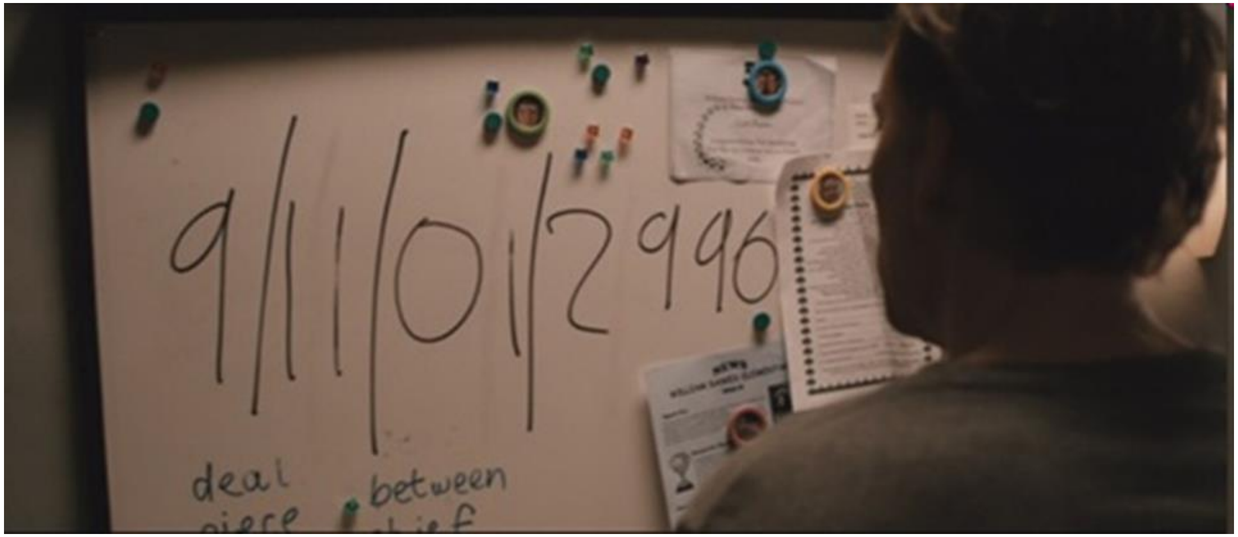
Basic Concepts

- Data
  - Does data always be useful?



Basic Concepts

- Data
  - Does data always be useful?



Basic Concepts

- Data
  - Data on its own carries no meaning. For data to become information, it must be interpreted and take on a meaning.

档案中的记录

（李明，男，1972，江苏，计算机系，1990）

数据的解释

语义：学生姓名、性别、出生年月、籍贯、所在系别、入学时间

解释：李明是个大学生，1972年出生，江苏人，1990年考入计算机系

请给出另一个解释和语义

Basic Concepts

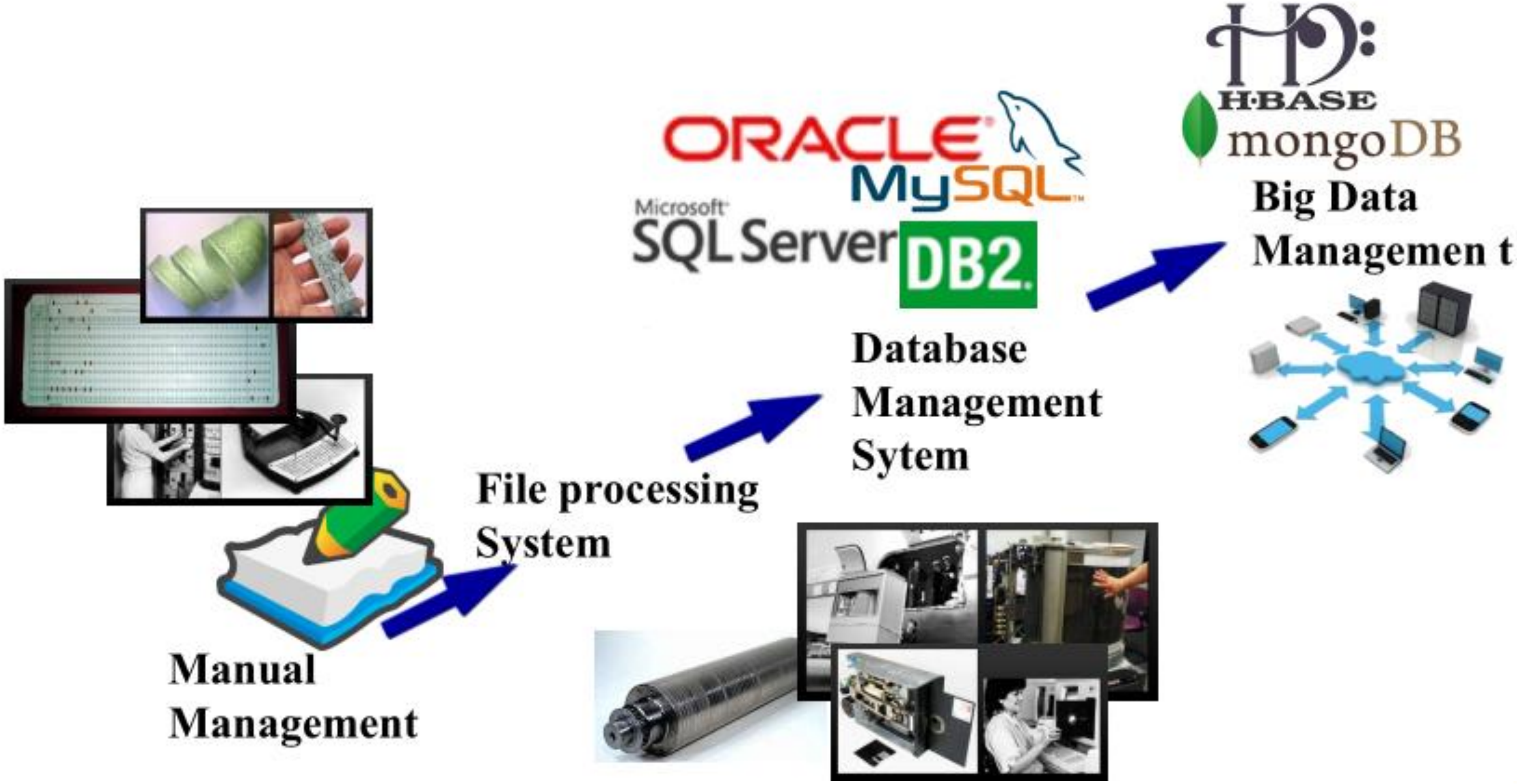
- Another Example
  - the height of Mt. Everest is generally considered as "data"
  - a book on Mt. Everest geological characteristics may be considered as "information"
  - a report containing practical information on the best way to reach Mt. Everest's peak may be considered as "knowledge".

8848.86

Summit
Northeast Ridge
West Ridge
Northeast Shoulder
4
3
2
Ice Cliff
Changtse
Cliffs
North Col
1
Advanced Base Camp
NORTH FACE
MOUNT EVEREST

Basic Concepts

- Data
  - Data Management Issues



Manual Management

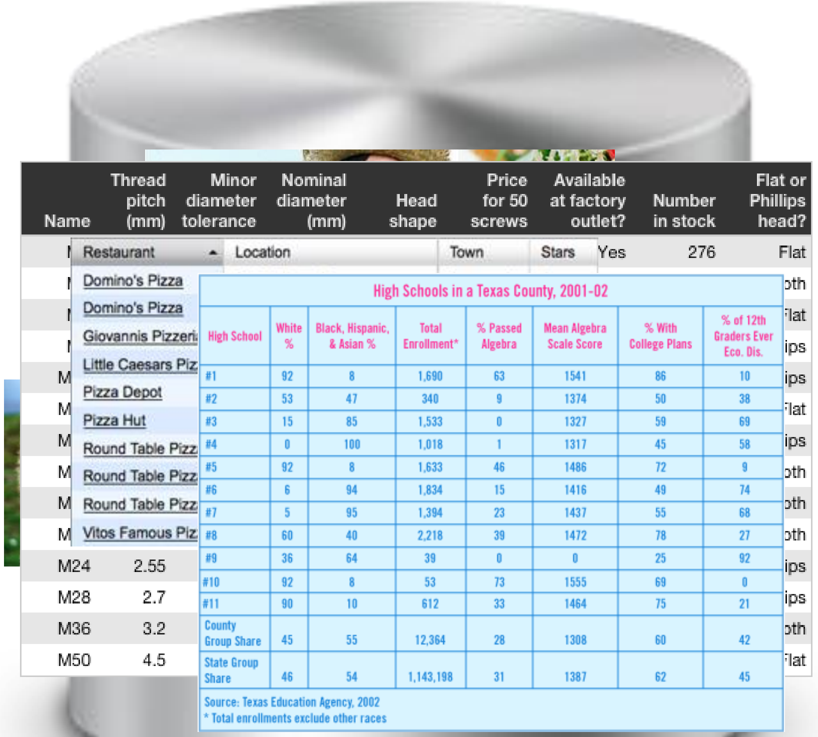
File processing System

Database Management Sytem

Big Data Managemen t

Basic Concepts

- Database (DB)
  - Collection of **interrelated** data



High Schools in a Texas County, 2001-02

High School	White %	Black, Hispanic, & Asian %	Total Enrollment*	% Passed Algebra	Mean Algebra Scale Score	% With College Plans	% of 12th Graders Ever Eco. Dis.
#1	92	8	1,690	63	1541	86	10
#2	53	47	340	9	1374	50	38
#3	15	85	1,533	0	1327	59	69
#4	0	100	1,018	1	1317	45	58
#5	92	8	1,633	46	1486	72	9
#6	6	94	1,834	15	1416	49	74
#7	5	95	1,394	23	1437	55	68
#8	60	40	2,218	39	1472	78	27
#9	36	64	39	0	0	25	92
#10	92	8	53	73	1555	69	0
#11	90	10	612	33	1464	75	21
County Group Share	45	55	12,364	28	1308	60	42
State Group Share	46	54	1,143,198	31	1387	62	45

Source: Texas Education Agency, 2002

* Total enrollments exclude other races

Basic Concepts

- How to access and manage database?

Name	Thread pitch (mm)	Minor diameter tolerance	Nominal diameter (mm)	Head shape	Price for 50 screws	Available at factory outlet?	Number in stock	Flat or Phillips head?
M24	2.55							
M28	2.7							
M36	3.2							
M50	4.5							

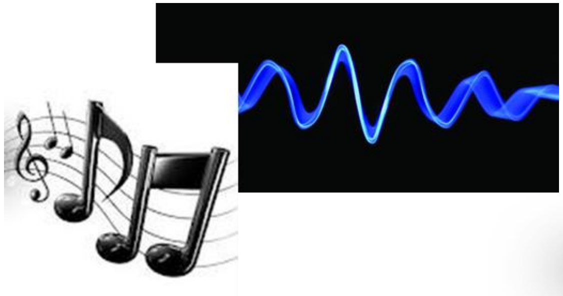
Restaurant | Location | Town | Stars | Yes | 276 | Flat

Domino's Pizza
Domino's Pizza
Giovannis Pizzeri
Little Caesars Piz
Pizza Depot
Pizza Hut
Round Table Pizz
Round Table Pizz
Round Table Pizz
Vitos Famous Piz

High Schools in a Texas County, 2001-02

High School	White %	Black, Hispanic, & Asian %	Total Enrollment*	% Passed Algebra	Mean Algebra Scale Score	% With College Plans	% of 12th Graders Ever Eco. Dis.
#1	92	8	1,690	63	1541	86	10
#2	53	47	340	9	1374	50	38
#3	15	85	1,533	0	1327	59	69
#4	0	100	1,018	1	1317	45	58
#5	92	8	1,633	46	1486	72	9
#6	6	94	1,834	15	1416	49	74
#7	5	95	1,394	23	1437	55	68
#8	60	40	2,218	39	1472	78	27
#9	36	64	39	0	0	25	92
#10	92	8	53	73	1555	69	0
#11	90	10	612	33	1464	75	21
County Group Share	45	55	12,364	28	1308	60	42
State Group Share	46	54	1,143,198	31	1387	62	45

Source: Texas Education Agency, 2002
* Total enrollments exclude other races

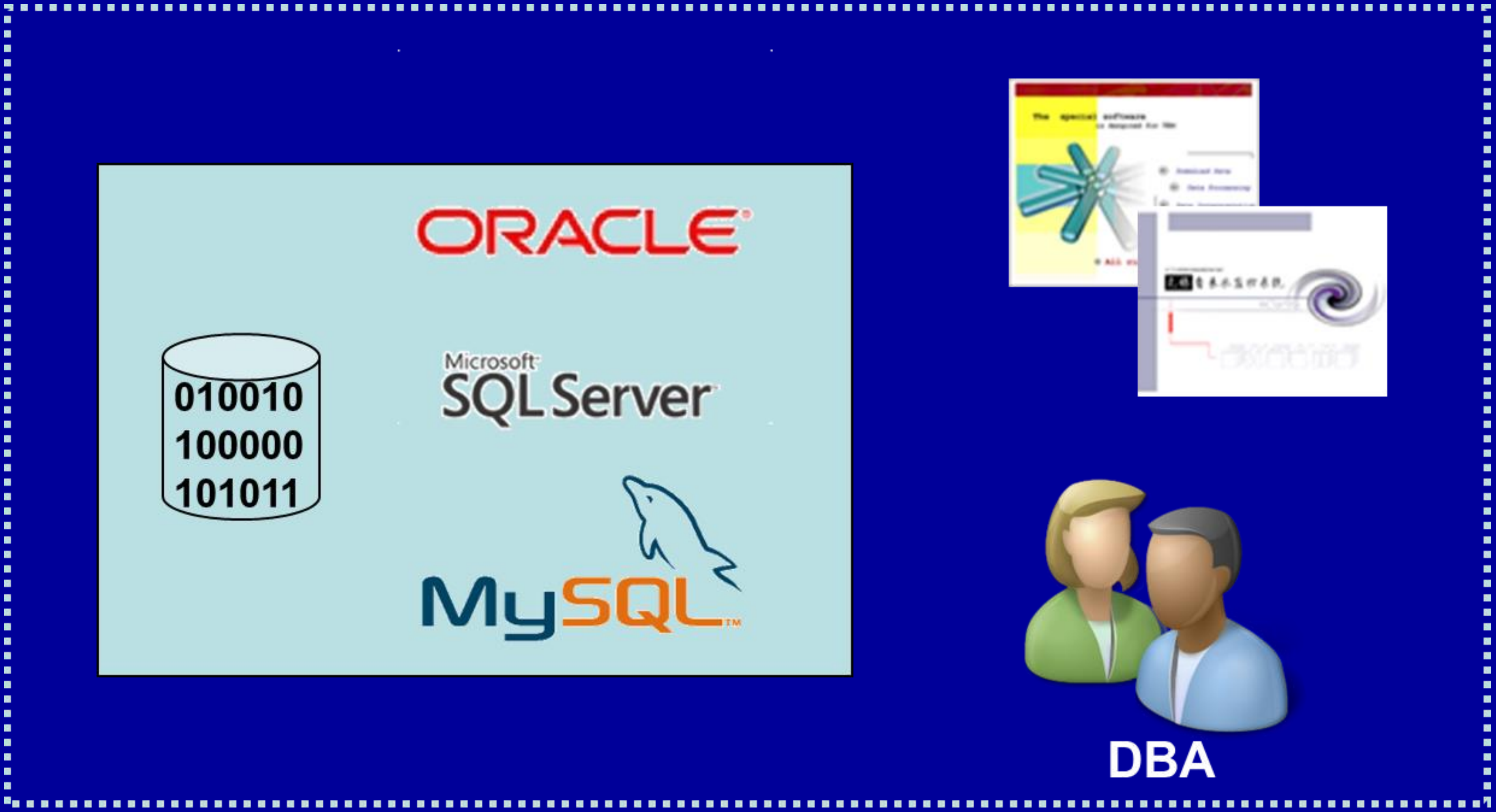


Basic Concepts

- Database Management System (DBMS)
  - a collection of **interrelated data** and a set of **programs** to access those data
  - An environment that is both *convenient* and *efficient* to use
  - DBMS is the core of **database systems**.
  - Database systems are designed to **manage large bodies of information**. Management of data involves both **defining structures for storage of information** and **providing mechanisms for the manipulation of information**.

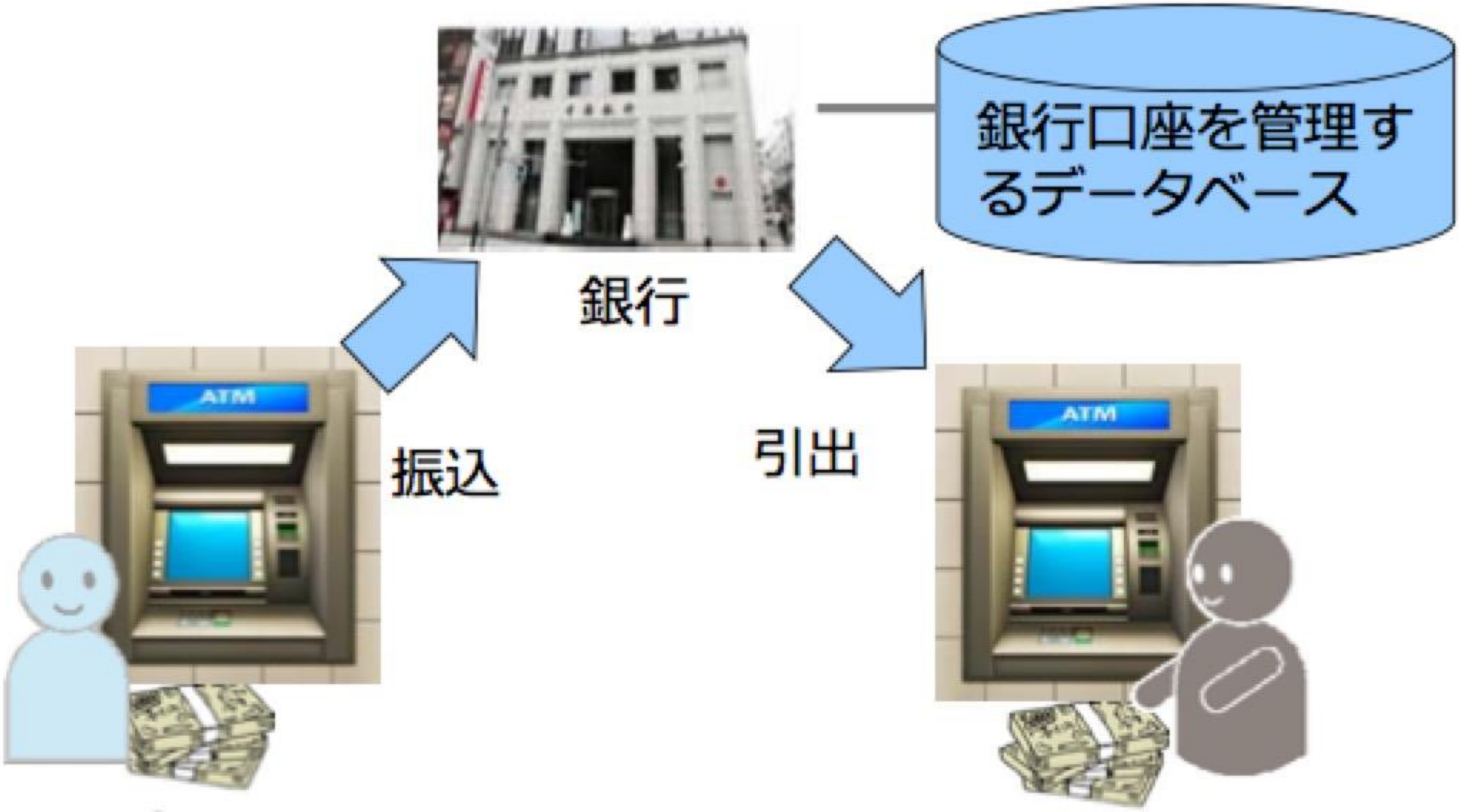
Basic Concepts

- **DBS**



Database Management System (DBMS)

- Database Applications:



x

- Databases can be very large.
- Databases touch all aspects of our lives

Section Review

– **Data**

– **Data management issues**

– **Database(DB)**:a collection of interrelated data , stored in systems as files (データベース)

– **Database management system (DBMS)**: a system/mechanism to manage data in DB
or: set of programs to access the data in DB
（データベース管理システム）

– **Database system(DBS)**: DB + DBMS + Users/Administers （データベースシステム）

– **Database application system**: DB + DBMS + Application programs + Users/Administers
（データベースアプリケーションシステム）

University Database Example

- Application program examples
  - Add new students, instructors, and courses
  - Register students for courses, and generate class rosters
  - Assign grades to students, compute grade point averages (GPA) and generate transcripts

- In the early days, database applications were built directly on top of file systems

- In the case, if new need arises, e.g. new major is to be created, then new files or even new application might be developed to fulfill the requirement.

Drawbacks of using file systems to store data (Cont.)

- This typical file-processing system is supported by a conventional operating system.
- Before database management systems (DBMSs) were introduced, organizations usually stored information in such systems.
- Keeping organizational information in a file-processing system has a number of major disadvantages

Drawbacks of using file systems to store data (Cont.)

- Data redundancy and inconsistency (数据冗余和不一致)
  - Multiple file formats, duplication of information in different files
- Difficulty in accessing data(数据访问困难)
  - Need to write a new program to carry out each new task
- Data isolation (数据孤立)
  - multiple files and formats
- Integrity problems (完整性问题)
  - Integrity constraints (e.g., account balance > 0) become "buried" in program code rather than being stated explicitly
  - Hard to add new constraints or change existing ones

Drawbacks of using file systems to store data (Cont.)

- Atomicity of updates (更新操作的原子性)
  - Failures may leave database in an inconsistent state with partial updates carried out
  - Example: Transfer of funds from one account to another should either complete or not happen at all

- Concurrent access by multiple users (多用户并发访问)
  - Concurrent access needed for performance
  - Uncontrolled concurrent accesses can lead to inconsistencies
  - Example: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time

- Security problems (安全性问题)
  - Hard to provide user access to some, but not all, data

Section Review

- Data redundancy and inconsistency (数据冗余和不一致)
- Difficulty in accessing data(数据访问困难)
- Data isolation (数据孤立)
- Integrity problems (完整性问题)
- Atomicity of updates (更新操作的原子性)
- Concurrent access by multiple users (多用户并发访问)
- Security problems (安全性问题)

Database systems offer solutions to all the above problems

View of Data

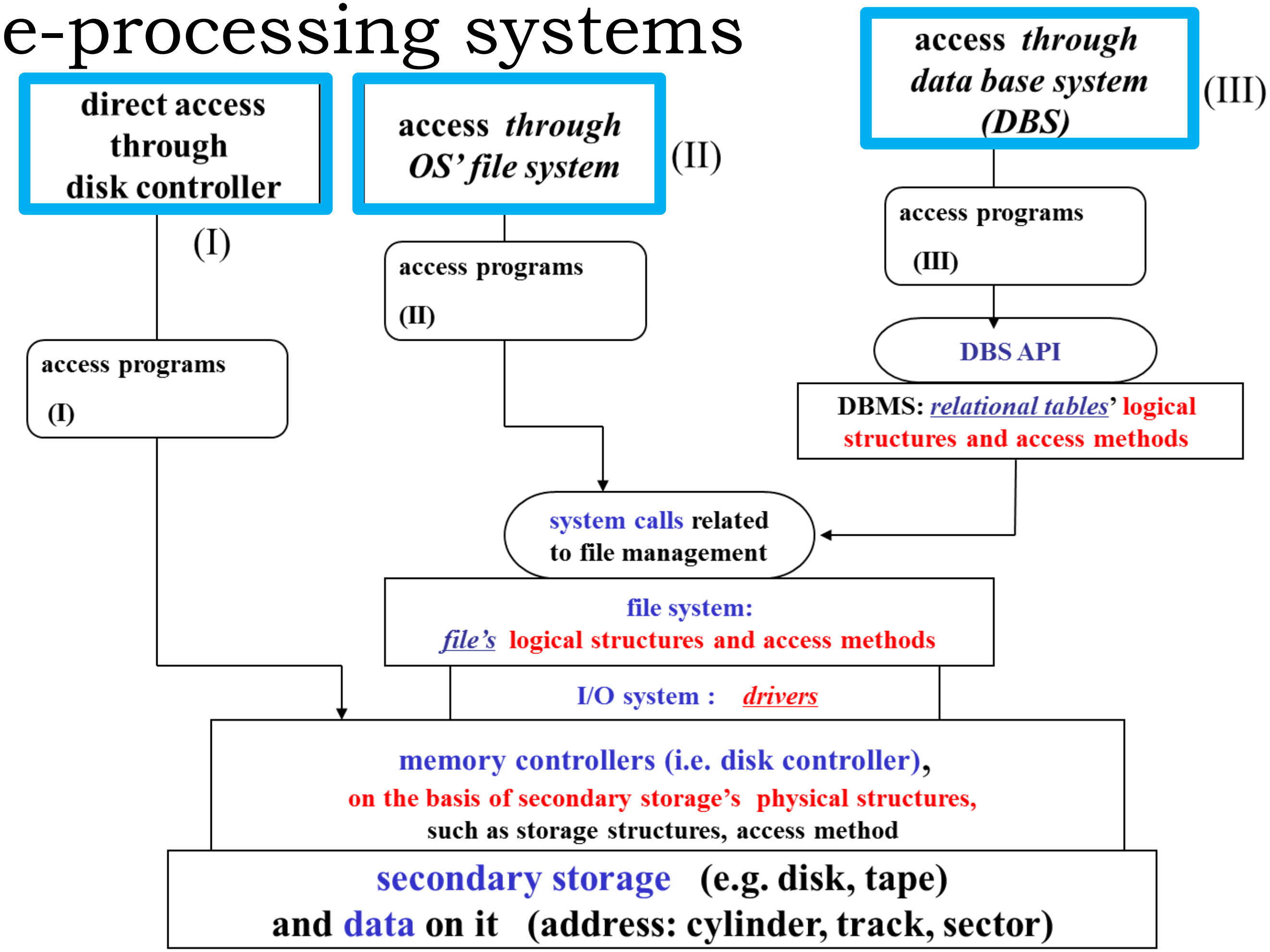
View of Data

- A major purpose of a database system is to provide users with an **abstract view of the data**(抽象的数据视图).

- That is, the system **hides certain details** of how the data are stored and maintained.(隐藏存储细节)

View of Data

- Recall the methods used in typical file-processing systems



direct access through disk controller

(I)

access programs (I)

access *through* *OS' file system*

(II)

access programs (II)

access *through* *data base system* *(DBS)*

(III)

access programs (III)

DBS API

DBMS: *relational tables*' logical structures and access methods

system calls related to file management

file system: *file's* logical structures and access methods

I/O system : *drivers*

memory controllers (i.e. disk controller), on the basis of secondary storage's physical structures, such as storage structures, access method

secondary storage (e.g. disk, tape) and data on it (address: cylinder, track, sector)

View of Data

- Most drawbacks of typical file-processing systems are due to the dependency among application programs and data (程序与数据的非独立/依存性）.

- On the contrary, DBMS is proposed to allow users to access and modify data more easily and more efficient.

- How can DBMS make it works?
  - Key: **Data abstraction**

Data Abstraction

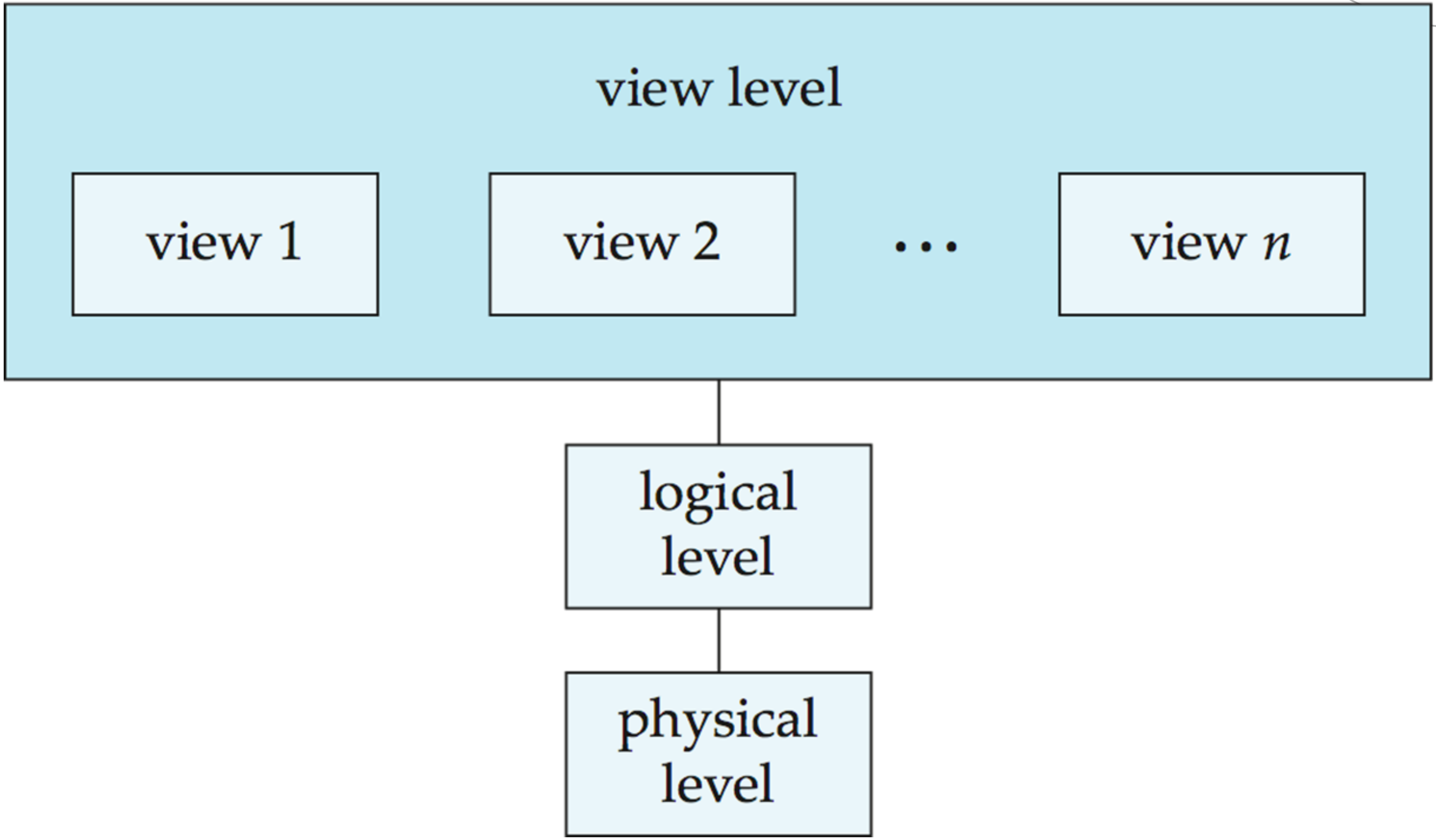
- For the system to be usable, it must retrieve data efficiently. (高效的检索数据)

- The need for efficiency has led designers to use **complex data structures**（数据结构） to represent data in the database.

- Since many database-system users are not computer trained, developers hide the complexity from users through several levels of abstraction, to simplify users’ interactions with the system.

Data Abstraction

- An architecture for a database system



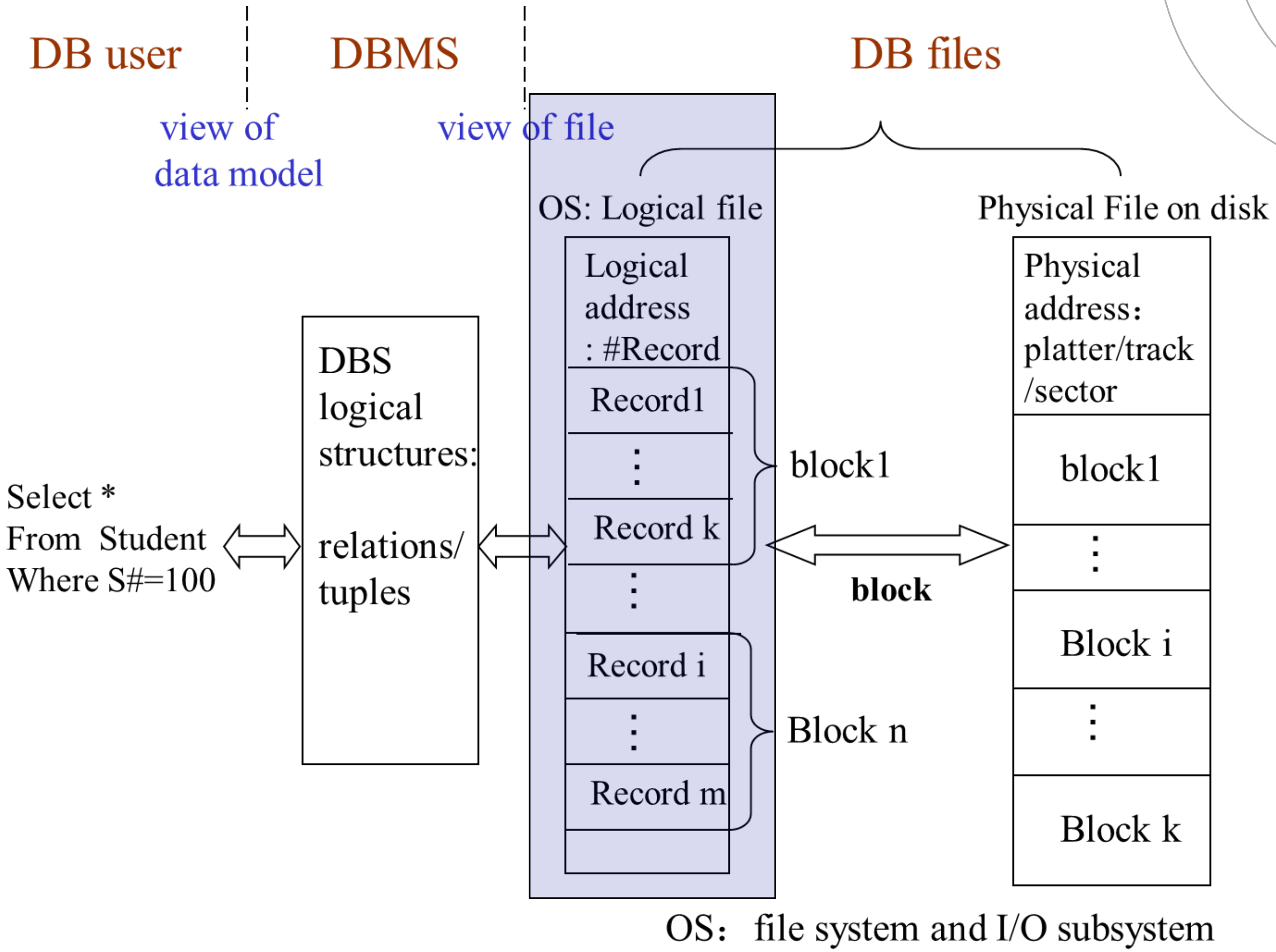
Data Abstraction

- **Physical level (物理层)** : describes how data (e.g., customer) is stored are actually stored in files (or in secondary storage)

- description results
  - physical/internal schema (物理模式，内模式)
  - i.e. storage structure and access methods, such as index, physical blocks, access methods for secondary memory, etc.

- description procedure/Physical DB design
  - physical abstraction

Data Abstraction



Data Abstraction

- **Logical level (逻辑层)** : describes data stored in database, and the relationships among the data.

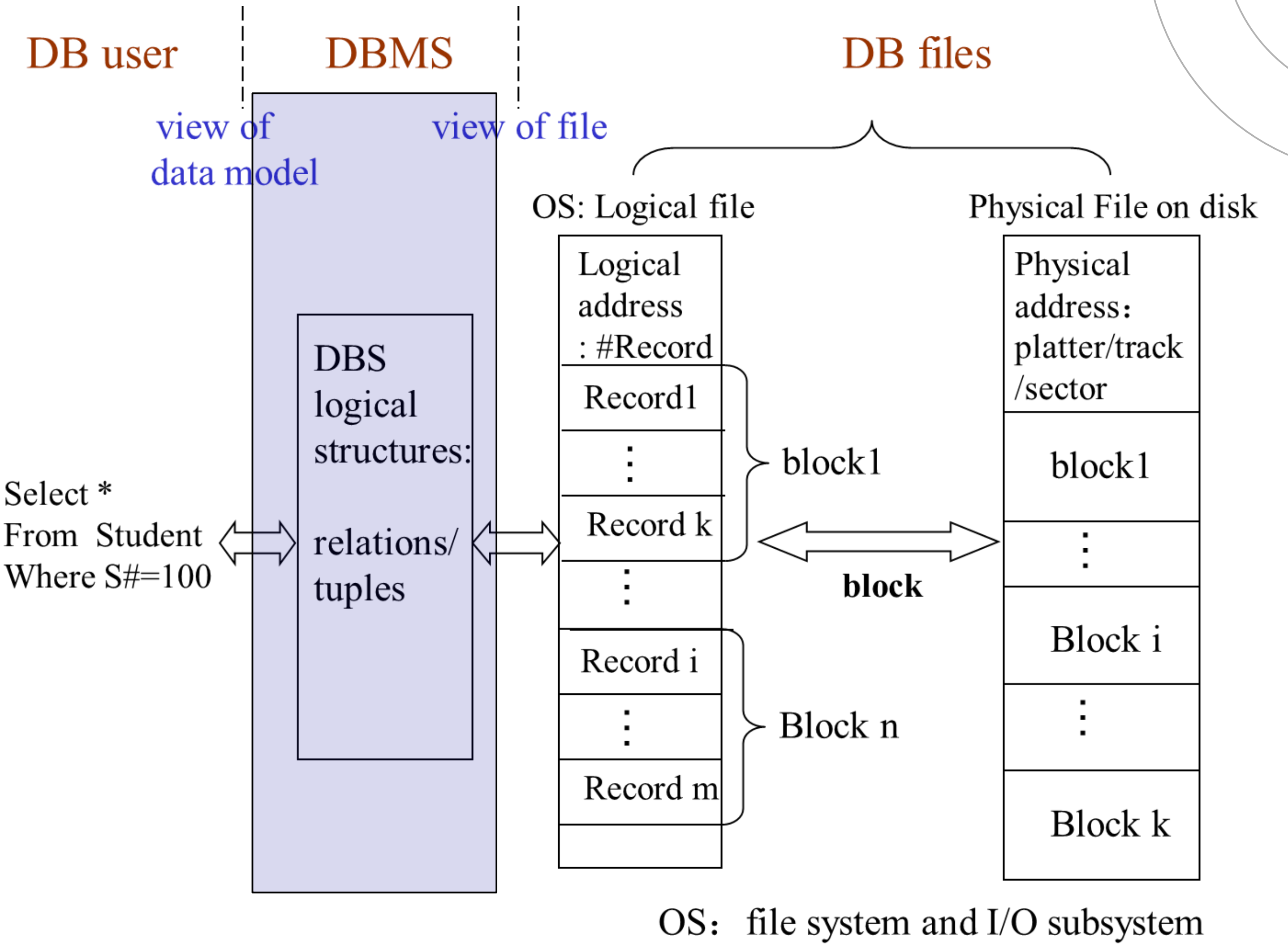
```
type instructor = record
                        ID : string;
                        name : string;
                        dept_name : string;
                        salary : integer;
                        end;
```

- description results
  - logical schema (逻辑模式)，e.g. relational tables

- description procedure/Logical DB design
  - logical abstraction

- **Merits: hiding physical implementation details (隐藏物理实现细节)**

Data Abstraction



Data Abstraction

- **View level (视图层)** : describes data from different view of data
  - in application areas, data item and associations among them
  - several views for one datum

- description results
  - external schema (外模式)={view}, set of views

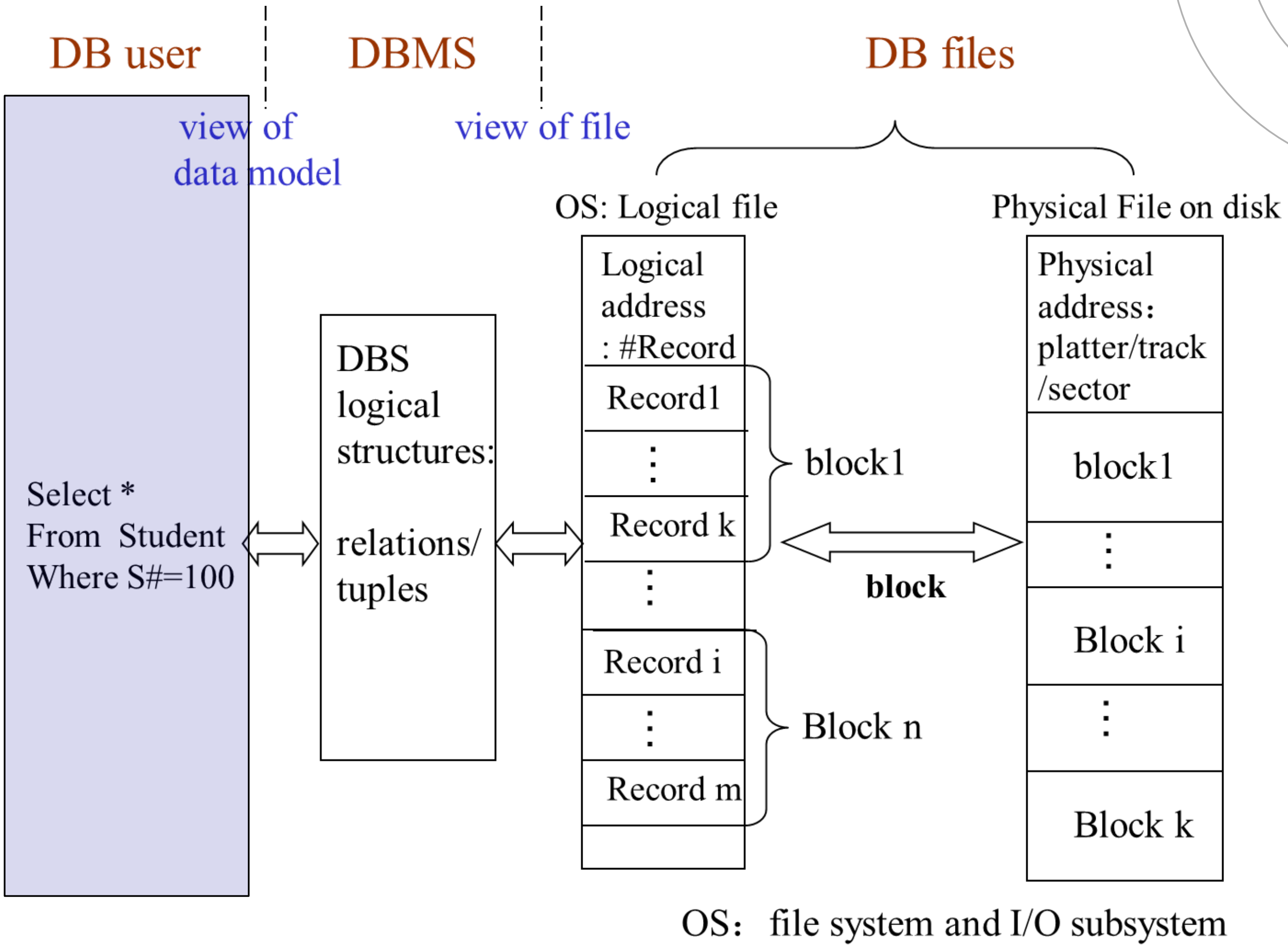
- description procedure/Logical DB design
  - view abstraction

- **merits: application programs** are programmed according to views, **hiding details of data types**. Views can also hide information (e.g., salary) for **security purposes**.

Data Abstraction

- E.g. Banking Application Areas
  - in view-level, finding objects, object’s features, and associations among objects
  - from more than one viewpoints, view integration
  - view1: customer<id, name, street, city>
  - view2: loan<loan-number, amount>
  - view3: account<account-number, balance>
  - view4: customer <borrower, loan>
  - view5: customer <depositor, account>
  - view6: ….
  - …….

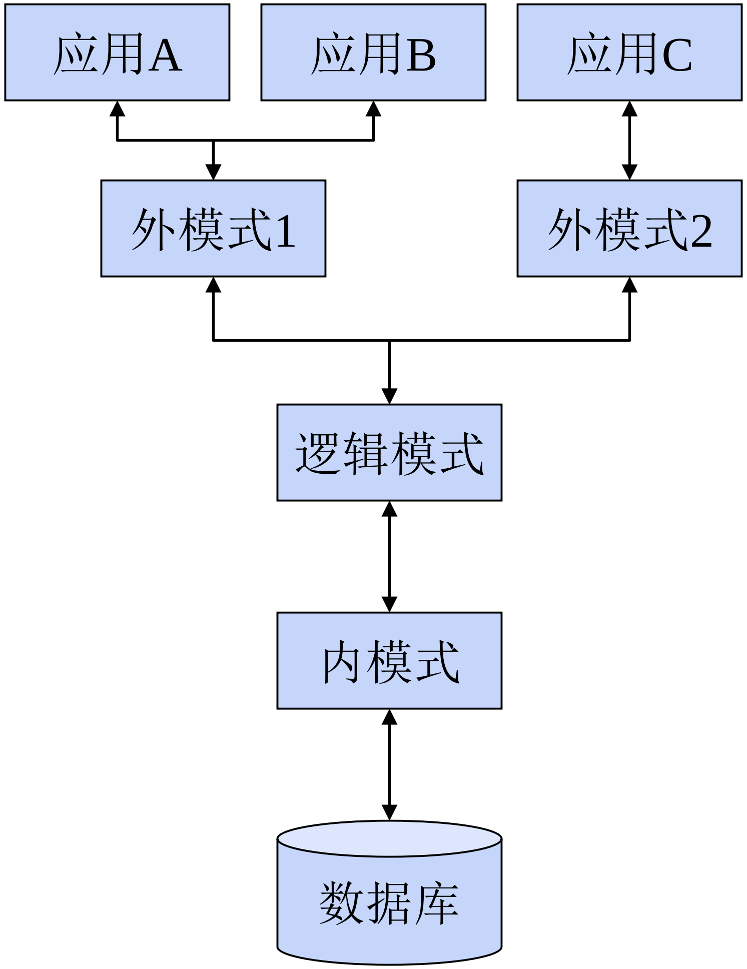
Data Abstraction

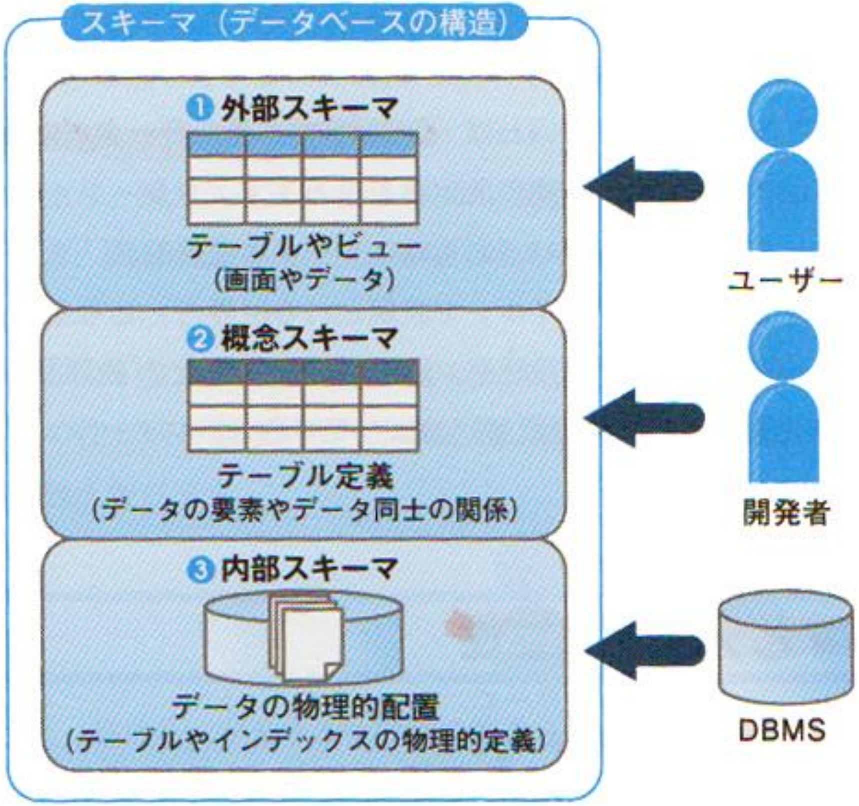


数据库系统的模式结构

- 三级模式

3層スキーマ





Instances & Schemas (实例&模式)

- **Schema(模式)** – the logical structure of the database
  - Example: 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 (物理模式): database design at the physical level
- Logical schema (逻辑模式): database design at the logical level
- Subschema (子模式): database design at the view level

Instances & Schemas (实例&模式)

- **Instance (实例)** – the actual content of the database at a particular point in time
  - Analogous to **the value of a variable**

Instances & Schemas (实例&模式)

- Schema and Instance
- E.g.
  - schema：　customer=<c_name, c_id, street, city>
  - instance of schema：　<Tom, 1001, Manhatton , New York >

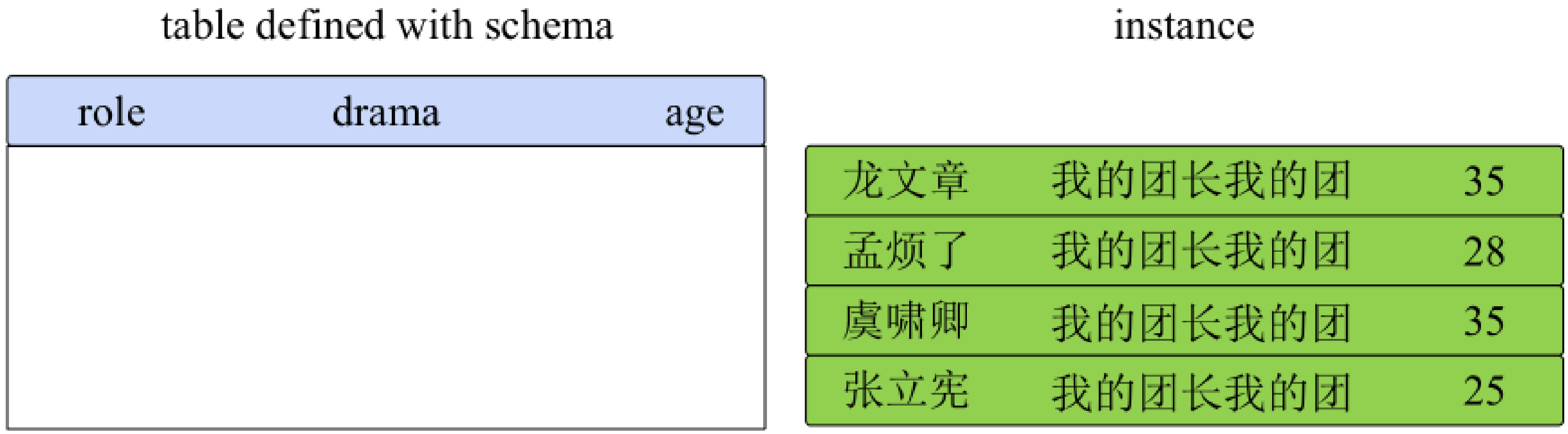


table defined with schema

role	drama	age

instance

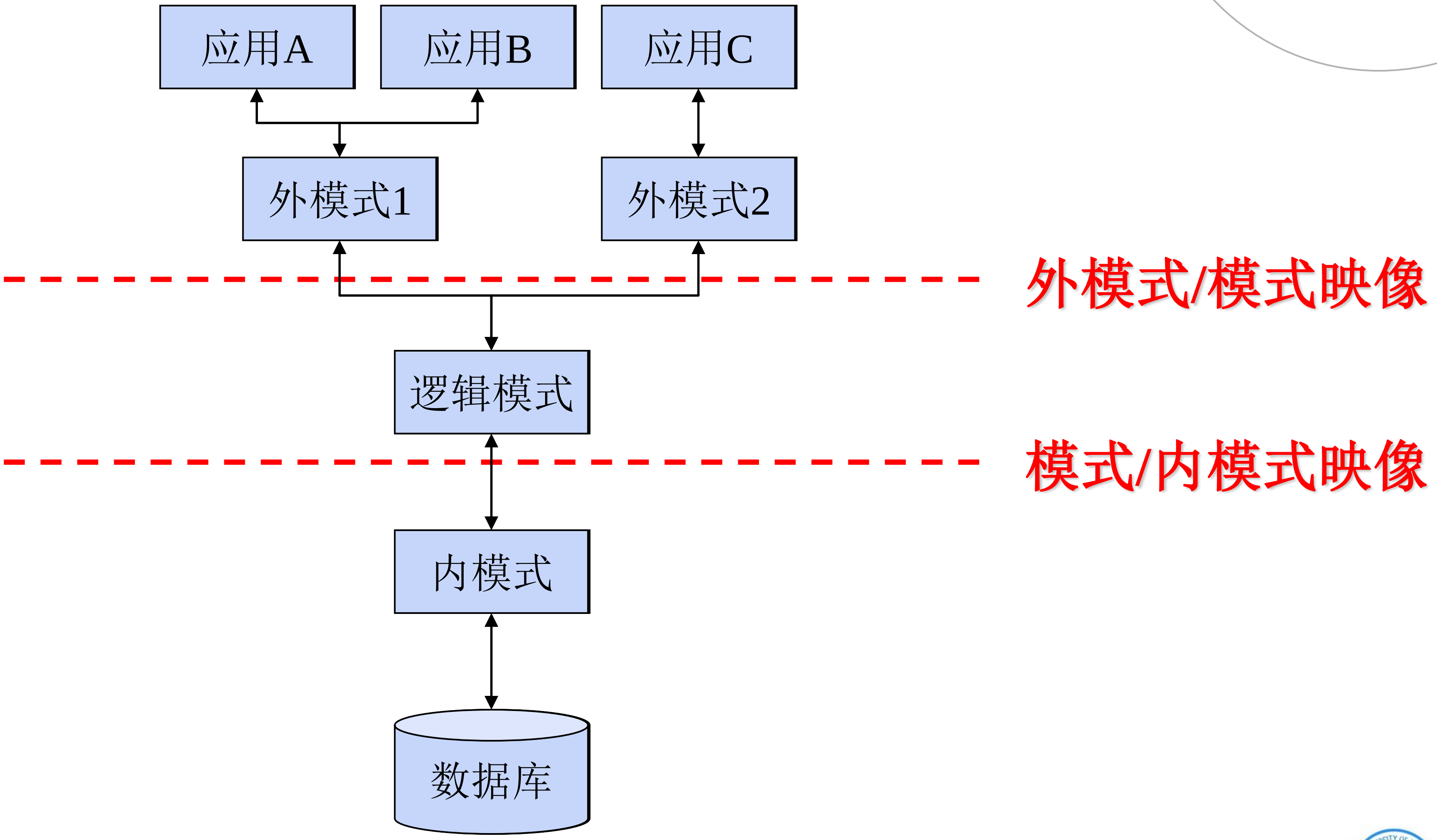
龙文章	我的团长我的团	35
孟烦了	我的团长我的团	28
虞啸卿	我的团长我的团	35
张立宪	我的团长我的团	25

Instances & Schemas (实例&模式)

- **Physical Data Independence(物理数据独立性)**
  - the ability to modify the physical schema without changing the 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.

数据库系统的模式结构

- 三级模式与数据独立性

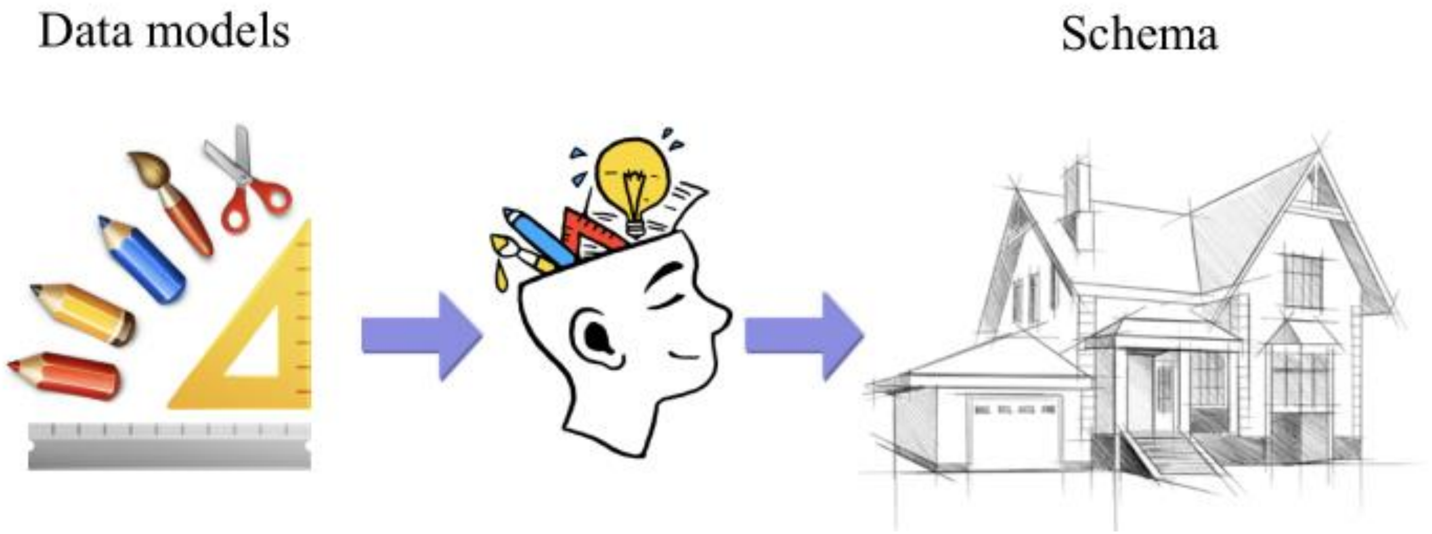


Section Review

- View of Data
  - Data Abstraction
    - architecture for a database system
      - Physical level
      - Logical level
      - View level
  - Schema (スキーマ)
    - Physical schema (内部スキーマ)
    - Logical schema　(概念スキーマ）
    - Subschema　(外部スキーマ）
  - Instance (インスタンス）
  - Physical Data Independence (物理的データ独立性)

Data Model (数据模型)

- Data descriptions/abstractions in three levels must obey three types of specification, i.e. three types of **data models**



Data Model

- Definition of data model:
  - a collection of conceptual tools for describing
    - data
    - data relationships (数据联系)
    - data semantics (数据语义)
    - consistency constraints (一致性约束)

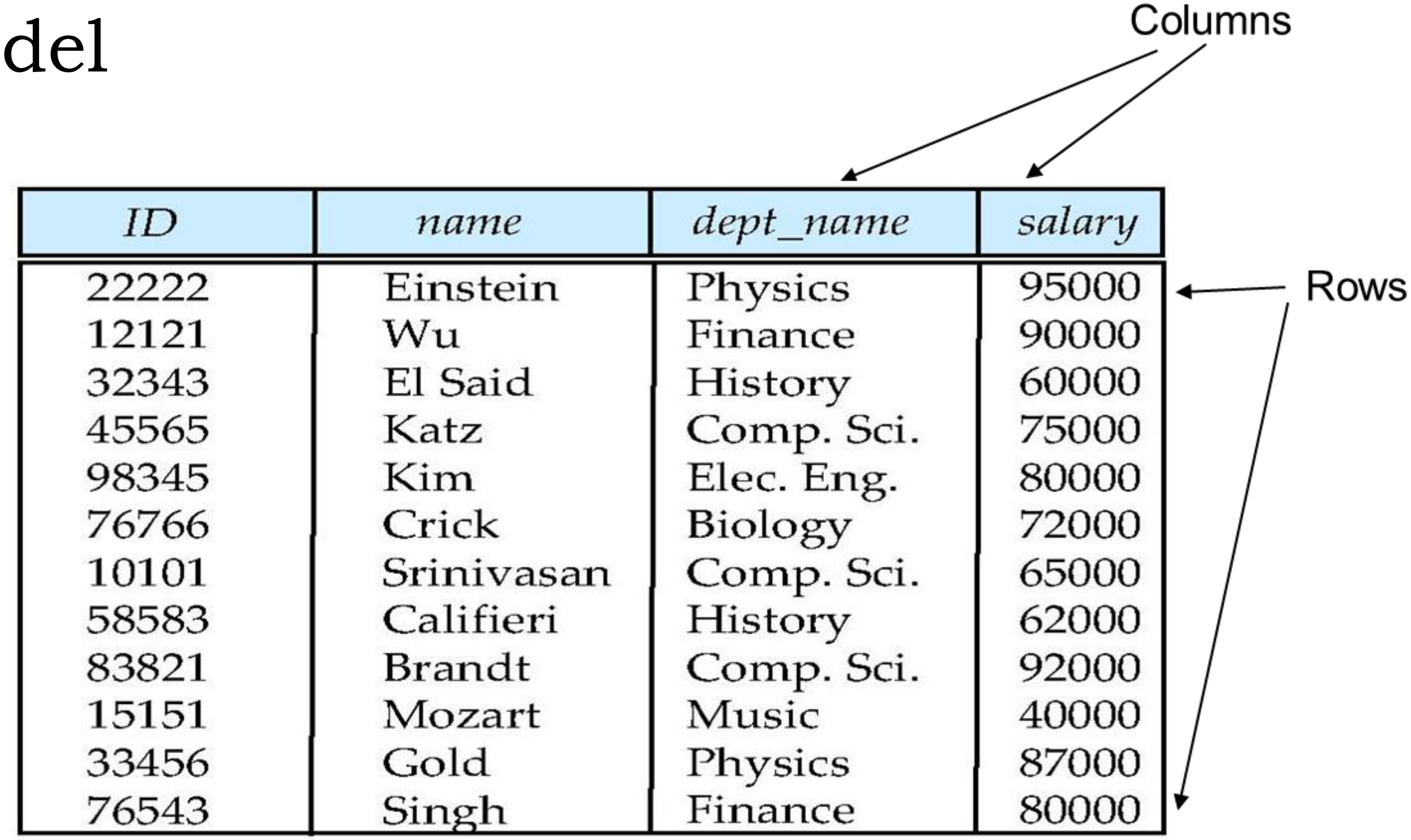
- A data model provides a way to describe the design of a database at the physical, logical, and view levels.

Data Model (数据模型)

- In the course, data models can be classified as
  - *Conceptual Data Model（概念数据模型）*
    - *Entity-Relationship Model (实体-联系模型)*
  - *Logical Data Model（逻辑数据模型）*
    - *Relational model (关系模型)*
    - *network data model (网状模型)*
    - *hierarchical data model (层次模型)*
    - *Object-based data model (基于对象的数据模型)*
    - *Semistructured data model (半结构化数据模型)*
  - *Physical Data Model（物理数据模型）*
    - *B* tree model…*

Data Model (数据模型)

- Relational model (Chapter 2)
- Example of tabular data in the relational model



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

Columns
Rows

(a) The *instructor* table

Database Languages

Database Languages

- Database Languages as human-machine interfaces
  - **Data-Manipulation Language, DML** (数据操纵语言)
  - **Data-Definition Language, DDL** (数据定义语言)

Database Languages

- **Data Manipulation Language (DML)**
  - Language for **accessing** and **manipulating** the data organized by the appropriate data model
  - DML also known as query language

- Two classes of languages
  - **Procedural (过程化DML)**– user specifies what data is required and how to get those data
  - **Declarative (nonprocedural) (声明式DML)** – user specifies what data is required without specifying how to get those data

- Query (查询):a statement requesting the retrieval of information

- SQL is the most widely used query language

Database Languages

- Data Definition Language (DDL)
  - DDL is used for specifying the database schema and additional properties of the data
    - E.g.

```
create table account (
              account-number    char(10),
              balance           integer);
```

  - DDL can also be used to define integrity constraints in DB
    - domain integrity（域约束）, referential integrity（参照完整性）, assertions（断言）, authorization（授权）, etc.

Database Languages

- Data Definition Language (DDL)
  - just like any other programming language, the DDL gets as input some instructions (statements) and generates some output.
  - The output of the DDL is placed in the **data dictionary(数据字典)**, which contains **metadata(元数据)**

Database Languages

- Metadata: data about data
  - The structures /**schemas** of the database defined by DDL
  - **Integrity constraints** (完整性约束)
  - **Primary key** (主键) (ID uniquely identifies instructors)
  - **Referential integrity** (参照完整性) (references constraint in SQL)
    - e.g. dept_name value in any instructor tuple must appear in department relation
  - **Authorization** (授权)

Database Languages

- SQL: widely used non-procedural language
  - Example: Find the name of the instructor with ID 22222

```
select  name
from    instructor
where   instructor.ID = '22222'
```

  - Example: Find the ID and building of instructors in the Physics dept.

```
select instructor.ID, department.building
 from instructor, department
 where instructor.dept_name=department.dept_name
   and department.dept_name = 'Physics'
```

Database Languages

- SQL
  - Application programs generally access databases through one of Language extensions to allow embedded SQL
  - Application program interface (e.g., ODBC/JDBC) which allow SQL queries to be sent to a database

Relational Databases

Relational Databases

- Relational database
  - based on the relational model
  - uses a collection of tables to represent both data and the relationships among those data
  - Includes a DML and DDL

  - Most commercial relational database systems employ the SQL language

  - More details refer to Chapter 2.

Review

Review

- Basic Concepts:
  - Data, Database, DBMS
  - Purpose of DBMS: Solve the problem of
    - Data redundancy and inconsistency (数据冗余和不一致)
    - Difficulty in accessing data
    - Data isolation (数据孤立)
    - Integrity problems (完整性问题)
    - Atomicity of updates (更新操作的原子性)
    - Concurrent access by multiple users (多用户并发访问)
    - Security problems (安全性问题)

Review

- View of Data
  - Data Abstraction
    - architecture for a database system
      - Physical level
      - Logical level
      - View level
  - Schema (スキーマ)
    - Physical schema (内部スキーマ)
    - Logical schema　(概念スキーマ）
    - Subschema　(外部スキーマ）
  - Instance (インスタンス）
  - Physical Data Independence (物理的データ独立性)

Review

- View of data
  - Data Model
    - a collection of conceptual tools
  - Classification of Data Model
    - *Conceptual Data Model*
    - *Logical Data Model*
    - *Physical Data Model*

- Database Languages
  - DDL & DML
  - SQL

- Relational database
  - based on the relational model

Thanks