Database System

北京交通大学软件学院

王方石 教授

E-mail: fshwang@bjtu.edu.cn

Contents

Chapter 1 -Contents

Examples of Database Applications

- Purchases from the supermarket
- Purchases using your credit card
- Booking a holiday at the travel agents
- Using the local library
- Studying at university
- ……

1.1 Four Basic Concepts

1. Data

- the basic object stored in the computer
- the symbol record depicting the objects that exists objectively
- Data type

  digit, text, graph, image, audio, video, etc.
- Example

  student (id, name, sex, dept, DOB)

  course (cid, name, credit)

1.1 Four Basic Concepts

2. Database (DB)

- **A shared collection of logically related data, and a description of this data, designed to meet the information needs of an organization.**
- **including two types of data:**
  - **User data, Logically related data comprises entities, attributes, and relationships of an organization's information. e.g. Grade=90**
  - **Metadata (System catalog , data dictionary) is the 'data about data' and provides description of data, e.g. int grade;**

1.1 Four Basic Concepts

3. Database Management System (DBMS)

- **A software system that enables users to define, create, maintain and control access to this database.**
- **The DBMS is the software that interacts with the users' application program and the database.**
- **Commercial DBMS**：DB2、Oracle、 SQL Server
- **Open Source DBMS**： **MySQL**

Functions Of DBMS

- **DBMS provides the following functions and service:**

(1)Data definition language (DDL).

- **Allows the user to describe the objects in database**
- **Permits users to specify data types, structures and any data constraints on the data to be stored in the DB.**

Structures： array or linked list

data constraints： month 1-12

Functions Of DBMS (cont.)

(2) Data Manipulation Language (DML)

- **Provides basic data operations (Data Retrieval, insert， delete and update) on data held in the database.**
- **Procedural DML: allows user to tell system exactly how to manipulate data.**
- **Non-Procedural DML :allows user to state what data is needed rather than how it is to be retrieved.**

Functions Of DBMS (cont.)

(3) It provides controlled access to database, it may include:

- **A security system, which prevents unauthorized users accessing the DB.**
- **An integrity system, which maintain the consistency of stored data when being stored more than once.**
  - DB integrity refers to the **correctness, validity** and **consistency** of the stored data. **integrity = constraints**
- **A concurrency control system, which allows shared access of the DB.**
- **A recovery control system, which restores the database to a previous consistent state following a hardware or software failure.**

An integrity system

s (sno, sname, age, sex) --Referenced Relation

sc (sno, cno, grade) --Referencing Relation

s

sno	sname	age	sex
S1	LI	17	M
S2	SHI	19	F
S3	LIU	21	F
S4	CHEN	20	M

sc

sno	cno	grade
S1	C1	78
S1	C2	86
S2	C2	77
	C3	76
S3	C3	89
S5	C4	80

No !

A concurrency control system

Two persons are booking the same flight using the ticket reservation system at the same time.

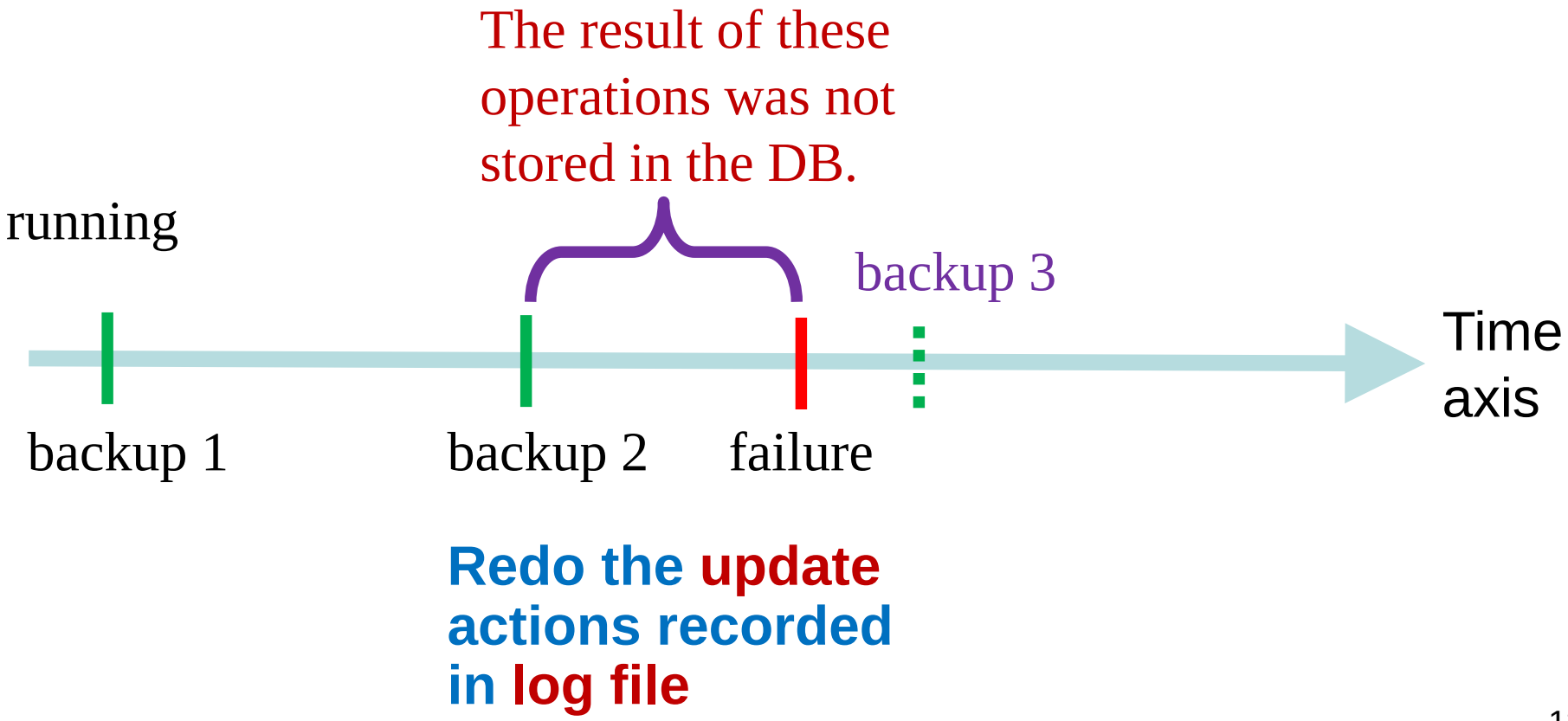
The number of current left seats: X = 100

	T1		T2
(1)	Read (X) from DB;	(3)	Read (X) from DB;
(2)	If X>0, then X = X - 1;	(4)	If X>0, then X = X - 1;
(5)	Write (X) back to DB	(6)	Write (X) back to DB

X =? 99 Correct ? No! X=98

A recovery control system

- Recover DB using periodical Backup and log file
- log file Contains information about all updates to database



Functions Of DBMS (cont.)

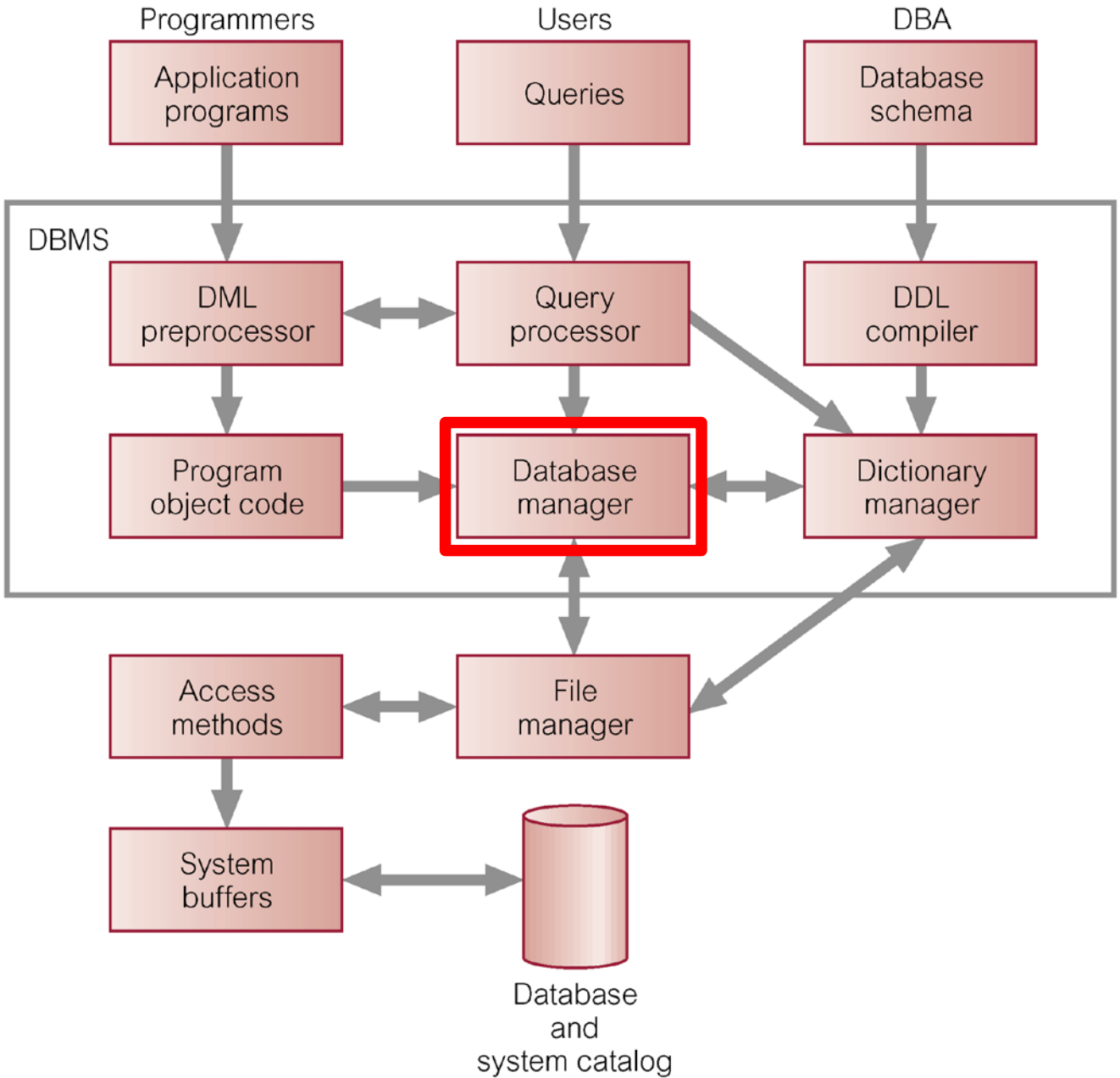
(4) A view mechanism.

- **Reduces complexity by letting users see the data in the way they want to see it.**
- **Customize the appearance of the DB. DOB->age**
- **Provides a level of security. Exclude data that some users should not see.**
- **Provides Services to Promote Data Independence.**

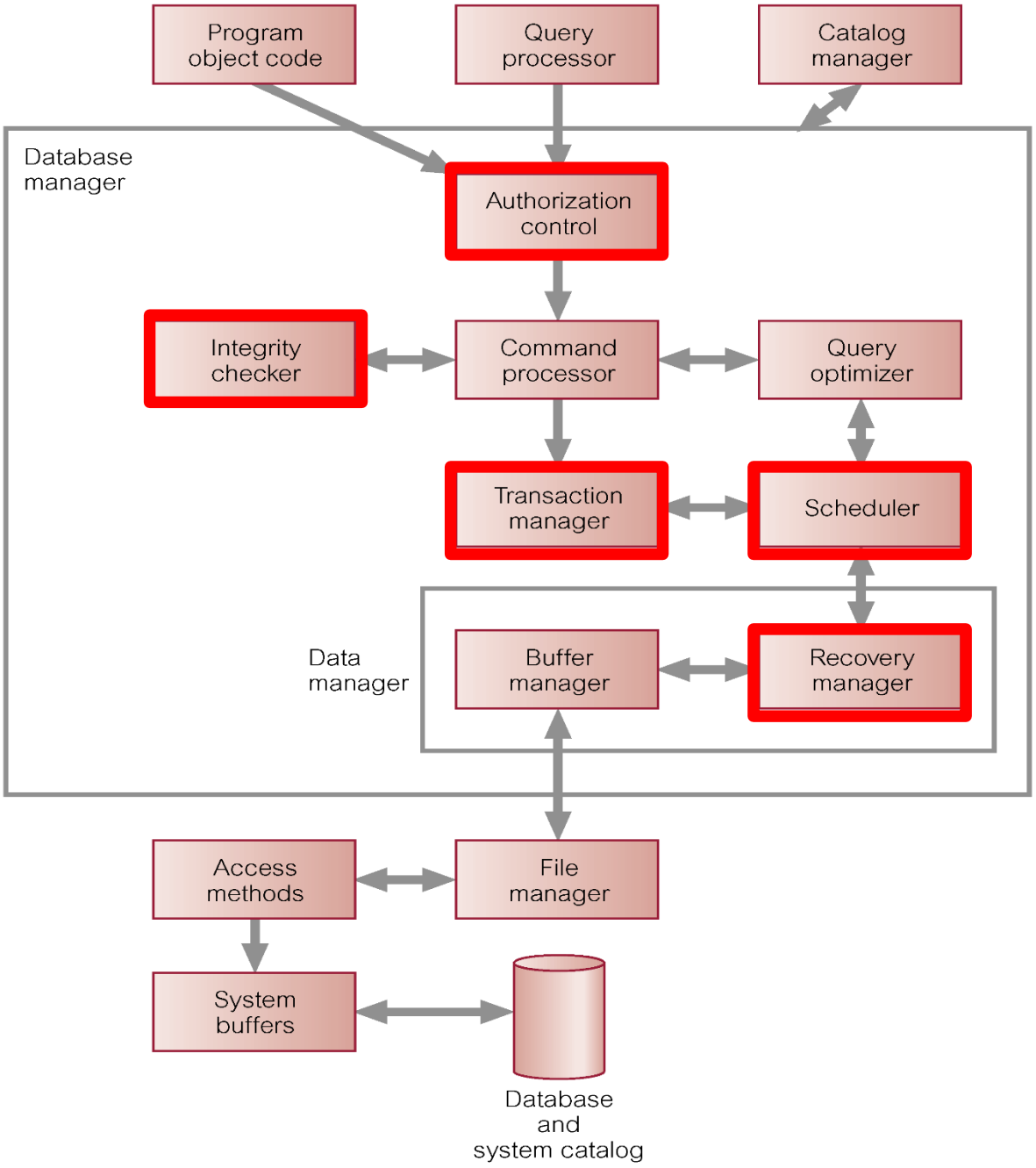
(5) Utility Services.

- **import facilities--Load data,**
- **monitoring facilities--Monitor performance,**
- **statistical analysis programs,**
- **index reorganization facilities**
- **garbage collection and reallocation**

Components of a DBMS



Components of Database Manager (DM)



Advantages & Disadvantages of DBMSs

1. Advantages of DBMSs

- **Control of data redundancy**
- **Data consistency**
- **More information from the same amount of data**
- **Sharing of data**
- **Improved data integrity**
- **Improved security**
- **Integration allows the DBA to define and enforce the necessary standards.**
- **Economy of scale**
- **Balance of conflicting requirements**

Advantages of DBMSs

- **Improved data accessibility and responsiveness**
  (obtain data without programming)
- **Increased productivity**
  (Programmer need not know the physical detail)
- **Improved maintenance through data independence**
  (Even if data are changed ,program need not to be changed)
- **Increased concurrency**
- **Improved backup and recovery services**
  (DBMS backups data automatically, data can be restored using Log files and copies)

2. Disadvantages of DBMSs

- **Complexity**
- **Size (several megabyte)**
- **Cost of DBMS**
- **Additional hardware costs (storage device)**
- **Cost of conversion**

  (e.g. Hierarchical DB to relational DB)
- **Performance**

  (In general, general systems run more slowly than special systems)
- **Higher impact of a failure**

1.1 Four Basic Concepts

4. Database System (DBS)

- **It is a computer system using DB technique**
- **Components of DBS**
  - **Hardware**

    Can range from a PC to a network of computers.
  - **Software**

    operating system, DBMS, network software (if necessary), High-level Programming Language, Compiling system, Application development tools and also the application programs.

Components of DBS (cont.)

- **Data**

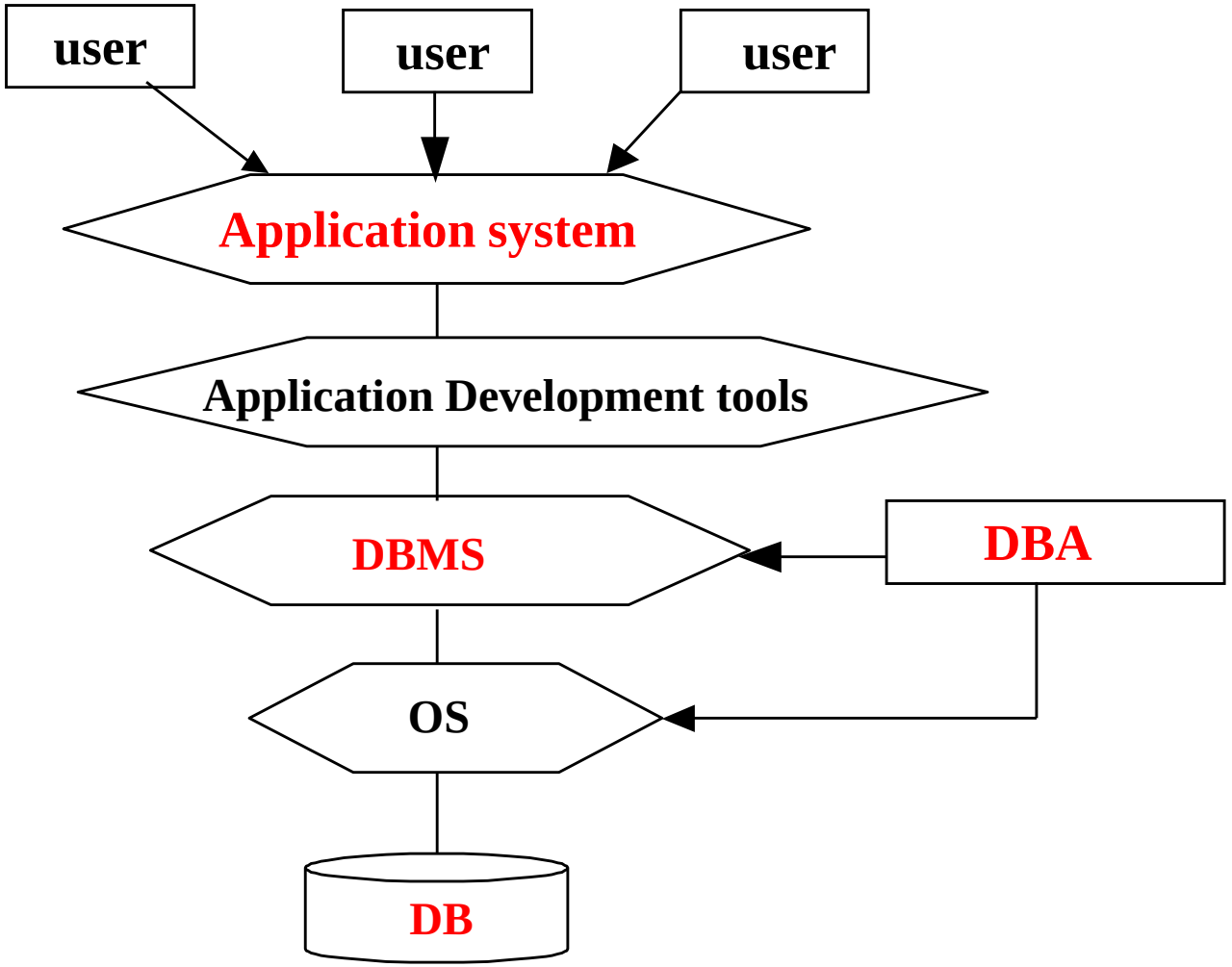
  Used by the organization and a description of this data called the schema.

- **People**
  - **Conventional users**
  - **DBA**

Roles in the Database Environment

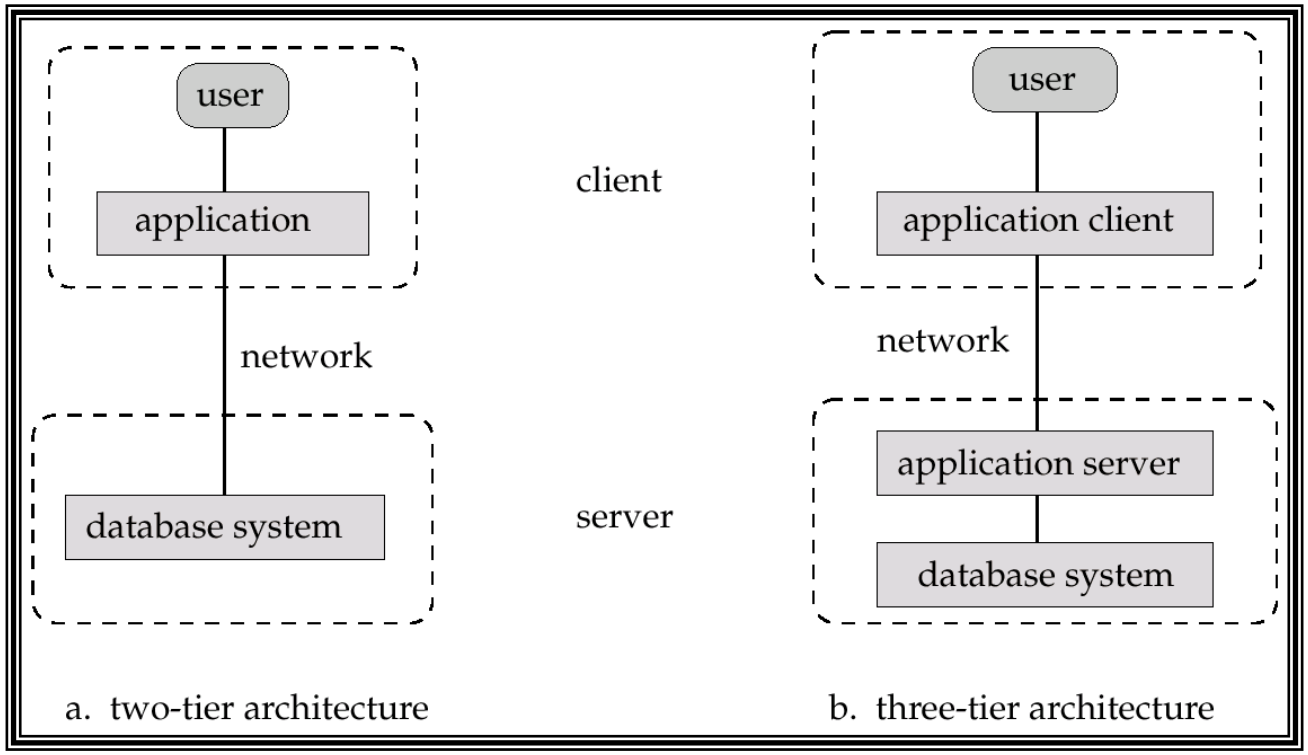
- **Data Administrator (DA)**
- **Database Administrator (DBA)**
- **Database Designers (Logical and Physical)**
- **Application Programmers**
- **End Users (naive and sophisticated)**

DB system: consisting of DB、DBMS、application system, and people.



Application Architectures

Business logics



browser

Business logics

- **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”

History of Database Systems

- **First-generation (middle of 1960s)**
  - **Hierarchical (IBM---IMS) ： Tree Structure**
  - **Network (DBTG---DBTG report, 1965-1971)： Graph Structure**
- **Second generation (1970, E.F. Codd)**
  - **Relational (IBM-System R， UCB- INGRES)**
- **Third generation (1980s-1990s)**
  - **Object-Oriented (OODB)（combine OO with DB, 1980s, fail）**
  - **Object Relational (ORDBS)（ introduce OO to RDB，Postgres）**
- **New Trend (after 1990s)**
  - **Semi-structured DBS (based on XML DM)**

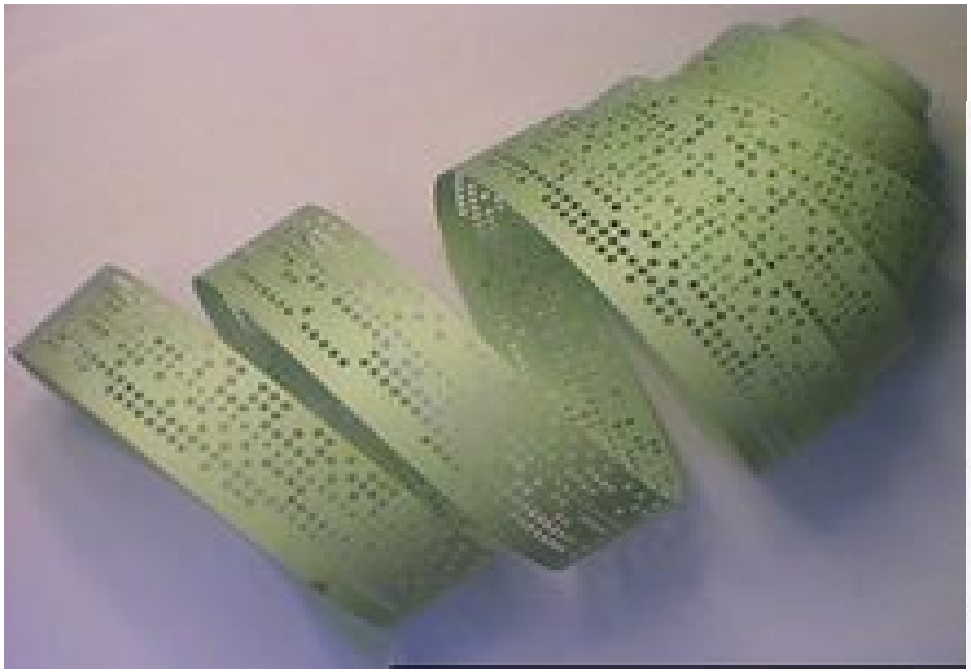
1.2 Development History of Data Management Technology

Data Management is to classify, organize, encode, store, retrieve and maintain data.

1. Manual System (before the mid-1950s)
2. Traditional File-Based System(1950s-1960s)
3. Database System (after the mid-1960s)

1. Manual System (before the mid-1950s)

- **Background**
  - Software: no OS or specified software
  - Hardware: no direct access device ,e.g. disk only punched tape, magnetic tape and card.

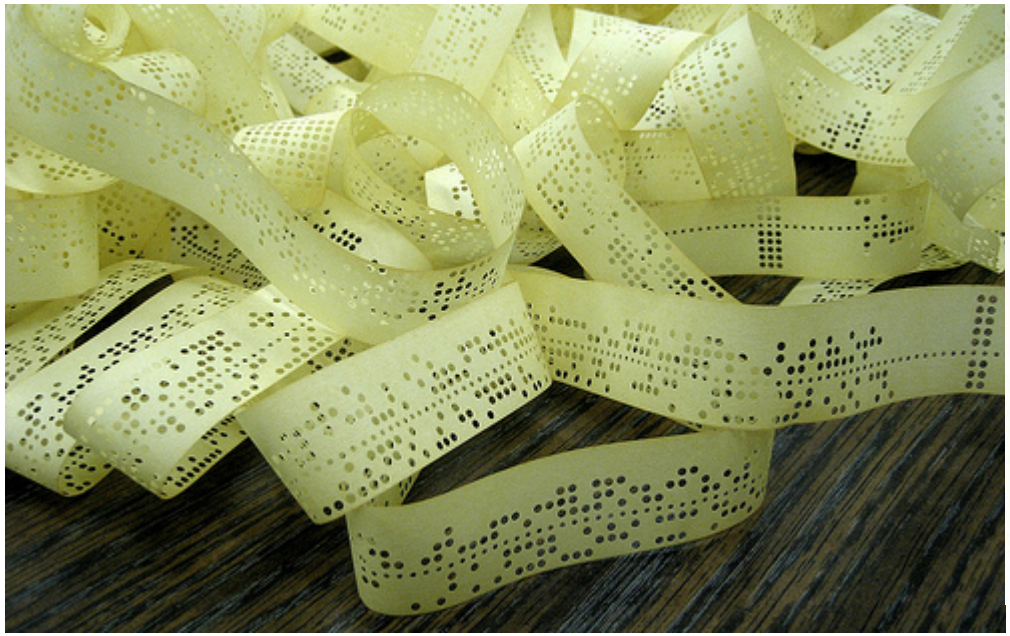


punched tape (老式打孔纸带)

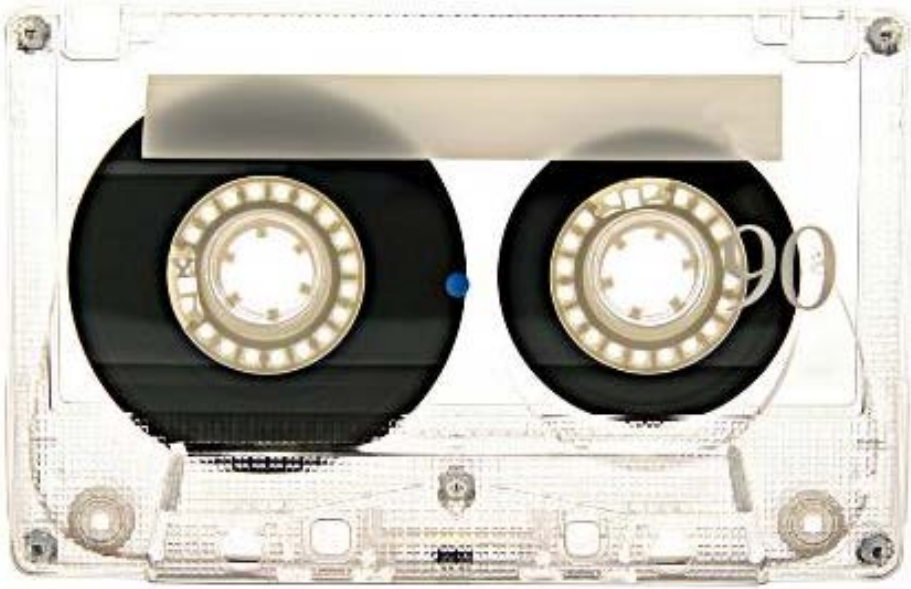
magnetic tape (磁带)

1. Manual System (before the mid-1950s)

- **Features**:
  - Data can not be kept for a long time
  - Program manages data
  - Data can not be shared
  - Data dependence



punched tape (老式打孔纸带)



magnetic tape (磁带)

2. Traditional File-Based System

(from the mid-1950s to the mid-1960s)

◆**A collection of application programs that perform services for the end users (e.g. reports).**

◆**Background**

- Hardware: Direct access device, e.g. **disk**, memory drum
- Software: OS—file system

8、5、3寸软盘

2. Traditional File-Based System

(from the mid-1950s to the mid-1960s)

◆Features (Outperform Manual System)

- ➢ Data can be kept for a long time, I, D,U, S operation
- ➢ File system manages data

memory drum（磁鼓）

Limitations of File-Based Approach

(1) Separation and isolation of data

- **Each program defines and manages its own data.**
- **Users of one program may be unaware of potentially useful data held by other programs.**

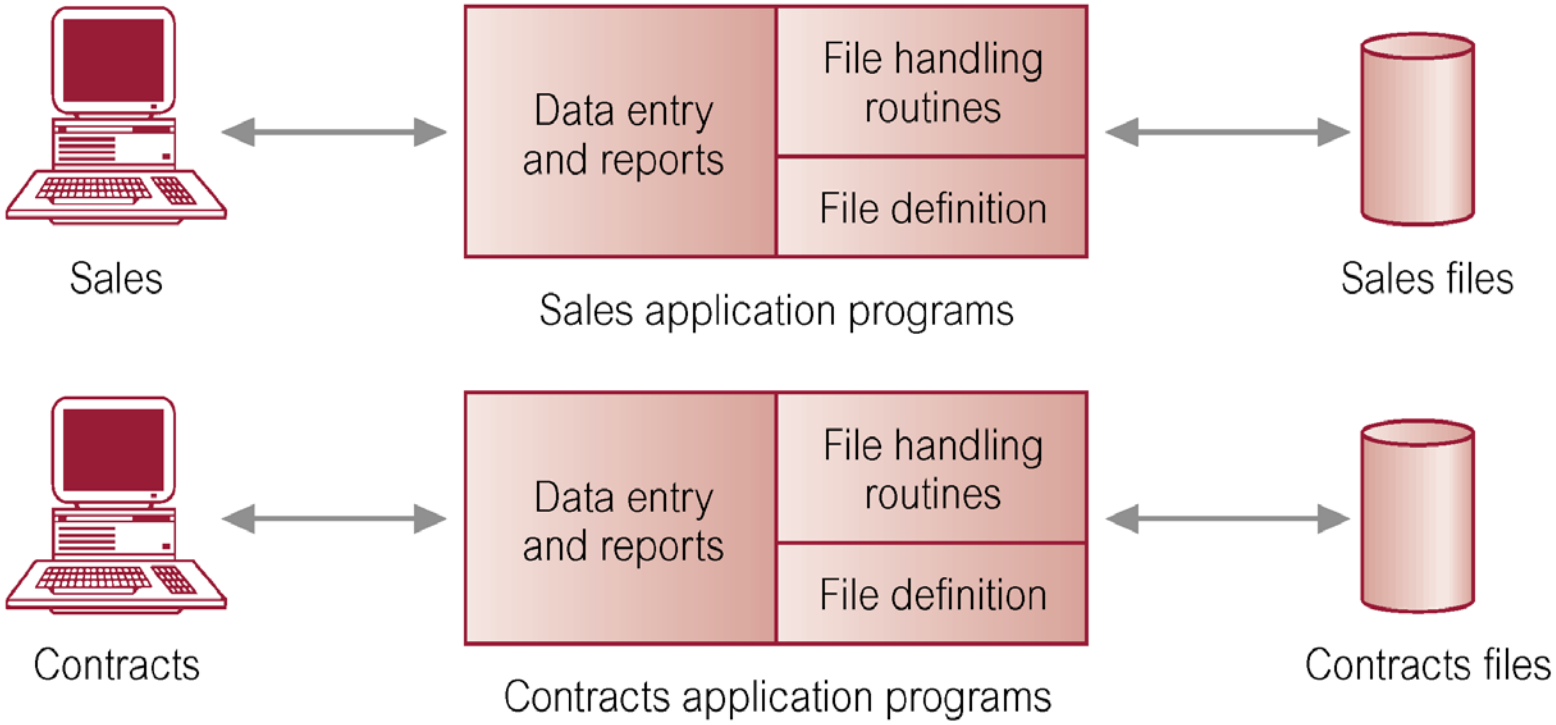
(2) Duplication of data

- **Same data is held by different programs.**
- **Wasted space and potentially different values and/or different formats for the same item.**

e.g. name char(30); or name varchar(25);

Not synchronized very well

File-Based Processing—House Agency



Sales Files

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)

PrivateOwner (ownerNo, fName, lName, address, telNo)

Client (clientNo, fName, lName, address, telNo, prefType, maxRent)

Contracts Files

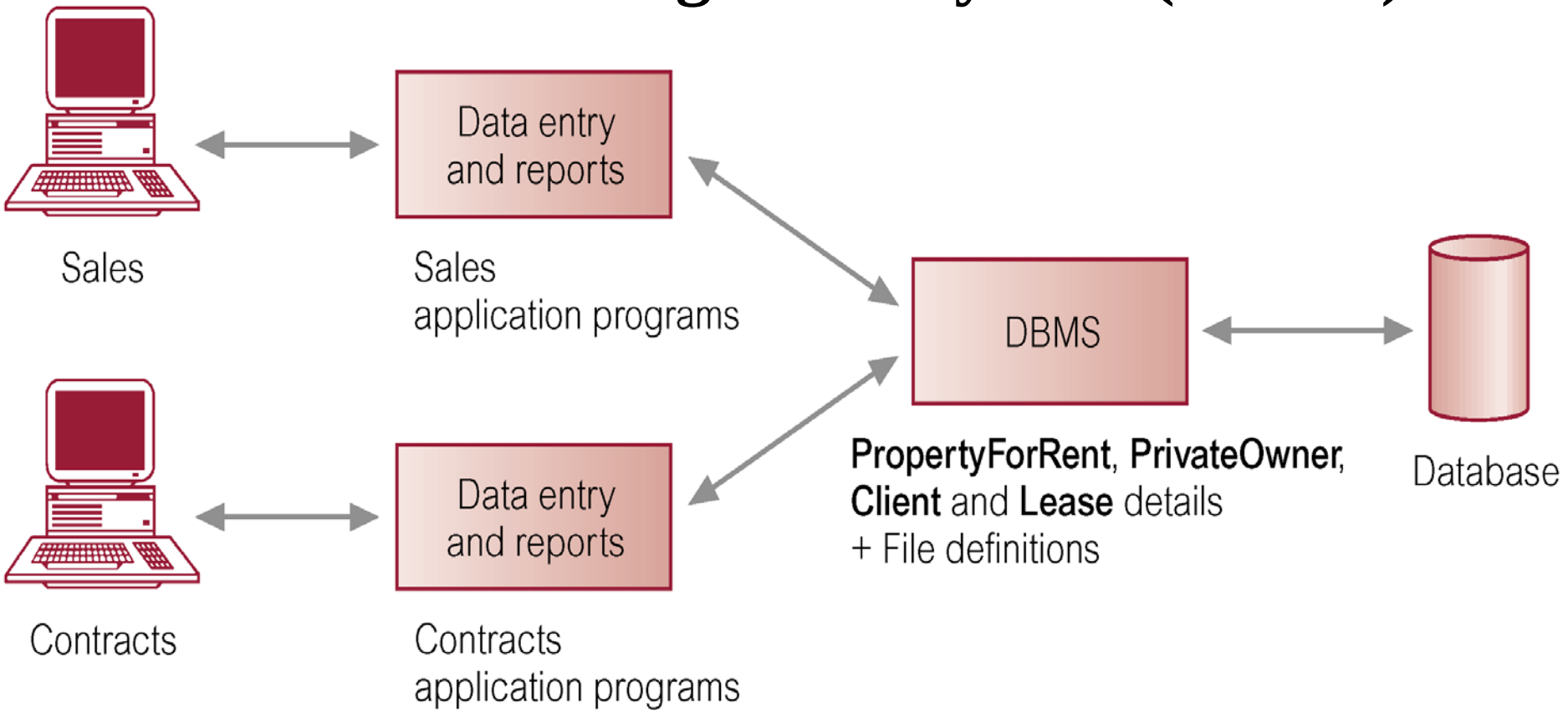
Lease (leaseNo, propertyNo, clientNo, rent, paymentMethod, deposit, paid, rentStart, rentFinish, duration)

PropertyForRent (propertyNo, street, city, postcode, rent)

押金

Client (clientNo, fName, lName, address, telNo)

Database Management System (DBMS)



PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)

PrivateOwner (ownerNo, fName, lName, address, telNo)

Client (clientNo, fName, lName, address, telNo, prefType, maxRent)

Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentFinish)

Limitations of File-Based Approach

(3) Data dependence

- File structure is defined inside the application code, known as program-data dependence (programs depend on data).

(4) Incompatible file formats

- Programs are written in different languages, and so cannot easily access each other's files.

(5) Fixed Queries/Proliferation (翻新) of application programs

- Programs are written to satisfy particular functions.
- Any new requirement needs a new program.
- Provide no security or integrity or recovery
- Access to the files was restricted to one user at a time

All the above limitations of the file-based approach can be attributed to two factors:

(1) The definition of the data is embedded in the application programs, rather than being stored separately and independently;

(2) There is no control over the access and manipulation of data beyond that imposed by the application programs.

To be more effective, a new approach was required. What emerged were the DB and the DBMS.

3. Database System (after the late of 1960s)

DBS is much better than the file-based approach

- Features of DBS
  - Data Structuring
  - High degree of data sharing, low degree of data redundancy
  - High degree of Data independence
  - Data can be managed and controlled unifiedly by DBMS

3. Database System (after the late of 1960s)

(1) Data Structuring

- Data is for the entire enterprise, not for only one application
- Both Internal and external data structuring, relationship among data
- The length of the data record is variable.
- The smallest access unit is **data item**.
- Data is described by **data model**, not in the application program.

3. Database System (after the late of 1960s)

(2) High degree of data sharing, low degree of data redundancy

- Reduce data redundancy, save the storage space
- Avoid the incompatibility （float & int） and inconsistency (stored more than once) of the data.

3. Database System (after the late of 1960s)

(3) High degree of Data independence

- Logical Data independence
- Physical Data independence

(4) Data can be managed and controlled unifiedly by DBMS.

- **security protection (priviledge)**
- **integrity check (correctness, validity and consistency)**
- **concurrency control**
- **database recovery**

1.3 Data Model (DM)

1.3.1 Definition

◆ **Model is a simulation of 'real world' objects and event, and their associations.**

e.g. Aeromodelling(航模) ⟶ plane

◆ **Data Model**

- **a kind of model**
- **an abstract of data feature in the real world**
- **【definition】An integrated collection of concepts for describing and manipulating data, relationships between data, and constraints on the data in an organization.**
- **Data Model is a simulation of Real world.**

Why need Data Model

- **Computer can not directly handle objects in real world.**
- **Objects in real world need to be transformed to data which can be handled by computer.**
- **DM is the tool used to abstract ,represent and handle the data and information in real world.**

Requirements for Data Model

- **can truly simulate the real world**
- **can represent data in an understandable way**
- **can be easily implemented on computers.**

DM is the core/kernel and foundation of DBS.

1.3.2 Three components of DM

Data Model comprises:

- **a structural part（Data Structure）;**
- **a manipulative part（Data Manipulation）;**
  - **query**
  - **modify (insert, delete, update)**
- **a set of integrity rules.**
  - **It is a set of rules (constraints) which ensure that the data is accurate.**
  - **Integrity=correctness + validity+ consistency**

1.3.3 Two Types of Data Model

◆Conceptual Model (called Information Model)

It builds models for data and information according to the view of users. It is used to design the database.

◆Structural Data Model (called Data Model)

➢Logical data model

- **It builds models for data and information according to the view of computer systems.**
- **It is used to implement DBMS.**

➢Physical data model

- **PM is the lowest layer of abstract of data.**
- **It describes the ways of data representation and data access in computer system.**
- **It is the task of DBMS to implement PM.**

Two types of Data Model

3 worlds

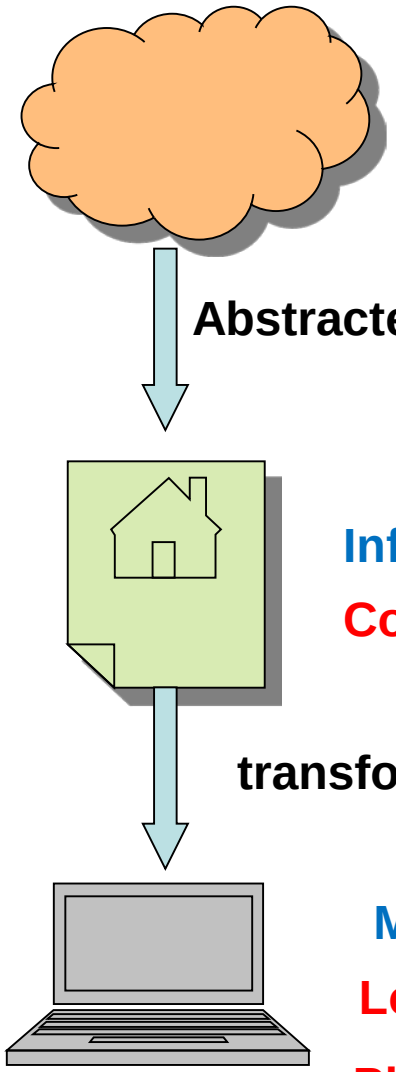
- **Real World**
- **Information World**
- **Machine World**

Data Model

- **Information World** — **Conceptual Model**
- **Machine World**
  - **Logical Model**
  - **Physical Model**

Two hierarchies of transformation from objects in **Real World** to data in machine world.

Two types of Data Model

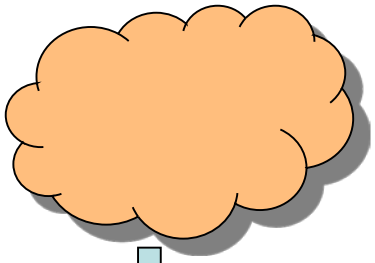


CM is also called Information Model. It builds models for data and information according to the view of users. It represents the logical (or community) view that is DBMS—— independent. It is used to design DB.

LM builds models for data and information according to the view of computer systems. It is used to implement DBMS. There are 3 types, network model, Hierarchical model and relational model.

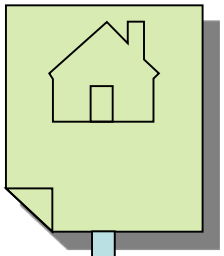
PM is the lowest layer of abstract of data. It describes the ways of data representation and data access in computer system. It is the task of DBMS to implement PM.

Two types of Data Model



Real World

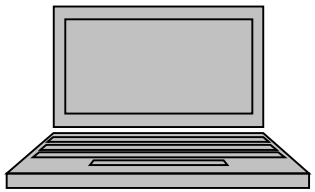
Abstracted by human brain



Information World

Conceptual Model

transform



Machine World

Logical Model and

Physical Model

概念模型，也称信息模型，它是按用户的观点来对数据和信息建模，主要用于数据库设计。

逻辑模型，主要包括网状模型、层次模型、关系模型等，它是按计算机系统的观点来对数据和信息建模，主要用于**DBMS**的实现。

物理模型是对数据最底层的抽象，它描述数据在系统内部的表示方式和存取方法，是面向计算机系统的。物理模型的具体实现是**DBMS**的任务.

Conceptual Model

◆Purpose of Conceptual Model

- ➢ It is used to build models in the information world.
- ➢ It is the middle layer between real world and machine world.
- ➢ It is a strong tool to design DB.
- ➢ It is a language for designers and users to communicate.

◆ Basic requirement for Conceptual Model

- ➢ Strong semantic expression ability
- ➢ Simple, clear, understandable

Basic concepts in information world

◆Entity

An entity is a distinct object (a person, place, thing, concept, or event) in the organization that is to be represented in DB.

◆Attribute

An attribute is a property that describes some aspect of the object that we wish to record.

e.g. Student (sno, sname, age, sex)

◆Relationship

A relationship is an association between entities.

e.g. Student & course M:N

Basic concepts in information world

- **Key（码、键）**

  A key is a set of attributes to identify one entity.

  e.g. StudID, name,（StudID, name), …

- **Entity type （实体类型）**

  A group of objects with the same properties, which are identified by the enterprise as having an independent existence. e.g. Student, Teacher

- **Entity Set（实体集）**

  The set of the Entity with the same type.

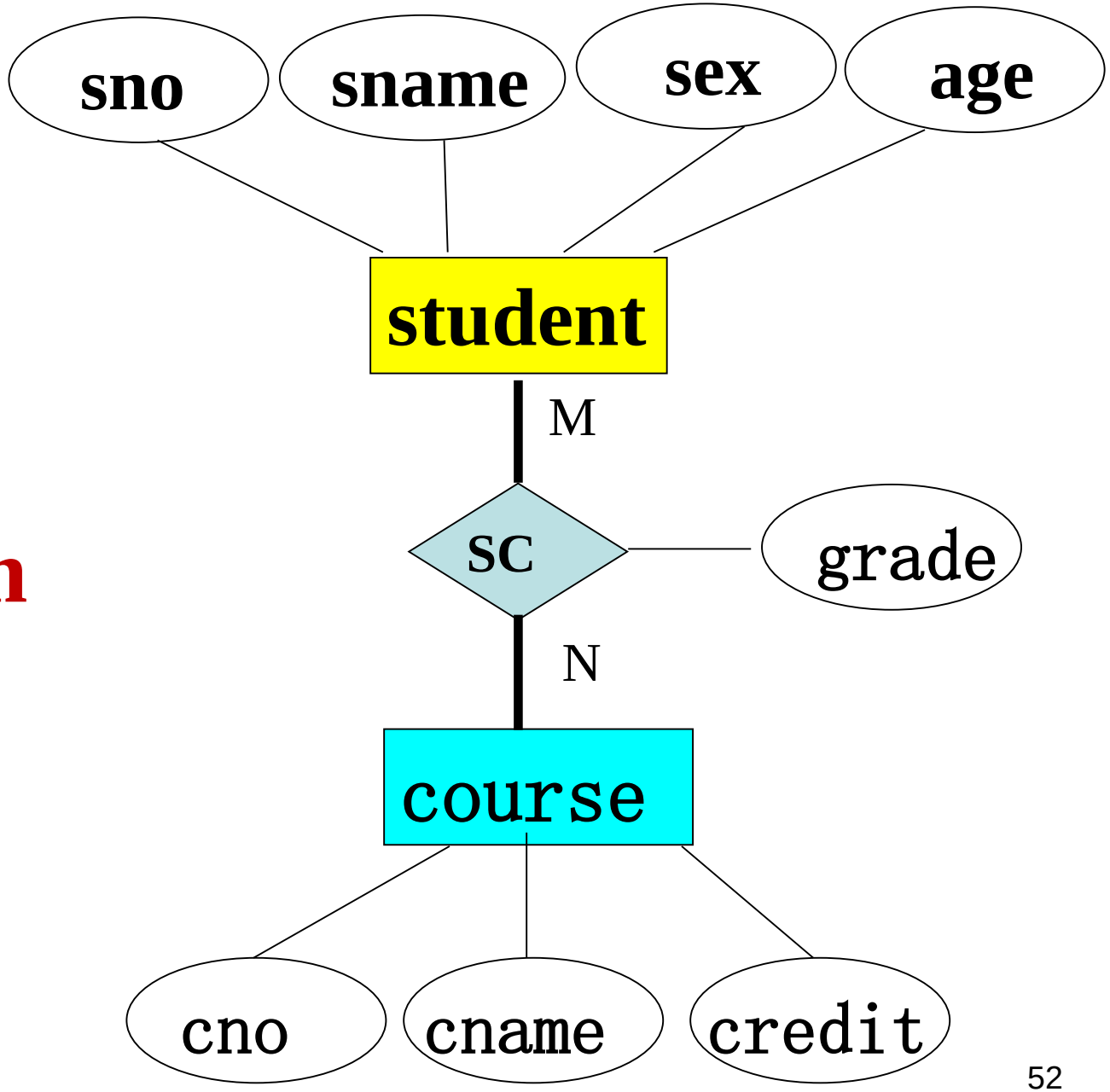
Conceptual Modelling （概念建模）

- **Conceptual modelling, or conceptual database design, is process of developing a model of information, which is independent of implementation details, such as the target DBMS, application programs, programming languages, or any other physical considerations.**
- **Its result is a conceptual data model.**
- **Conceptual model is independent of DBMS and all implementation details (DBMS-independent).**
- **Entity-Relationship Approach，called E-R model**

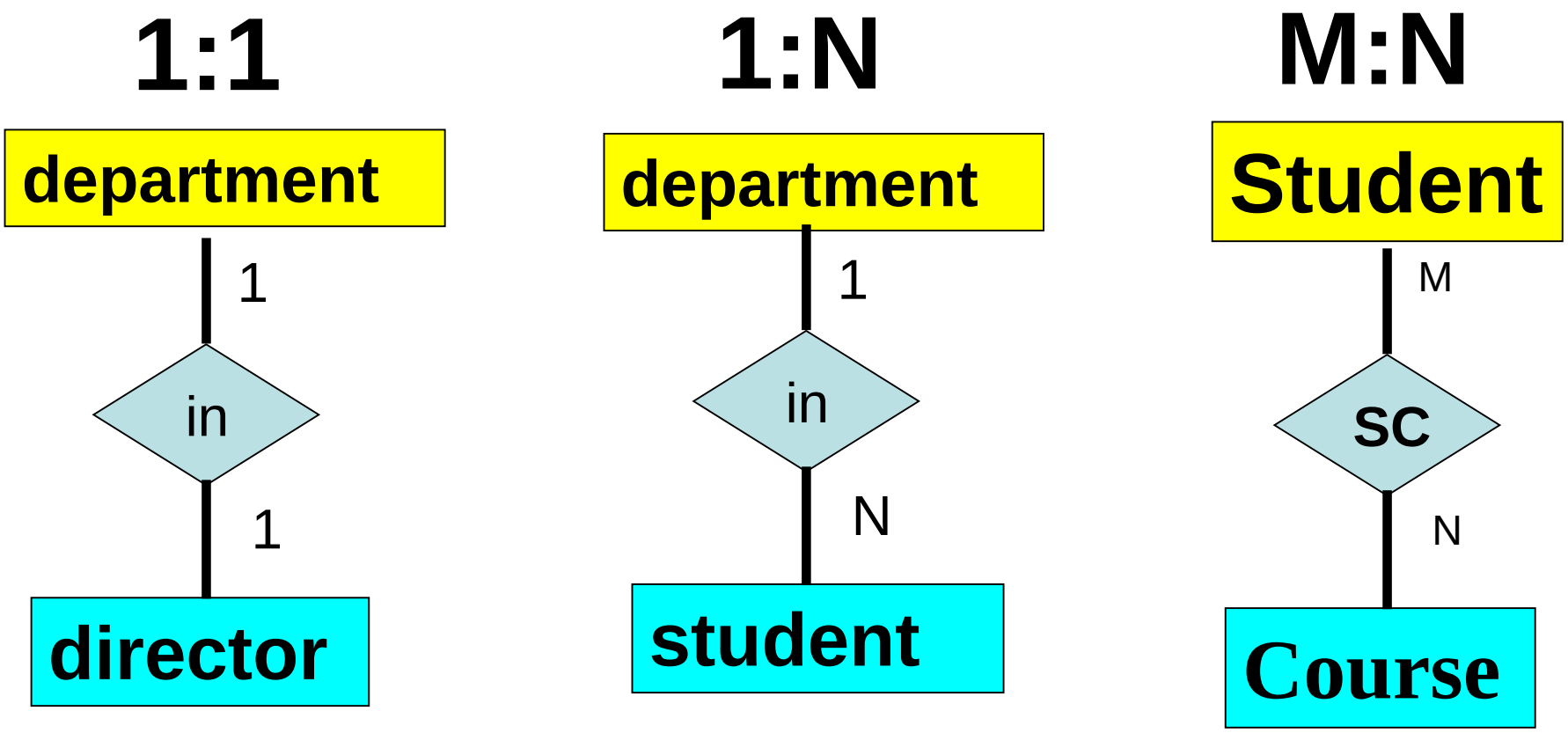
A simple example for Conceptual Model

ER Model

ER Diagram



Relationship Types



Commonly Used Logical Data Model

Logical data model assumes that we know the underlying data model of the target DBMS, which maybe is:

- **Hierarchical model---Tree**
- **Network model---Graph**
- **Relational model—2D tables**
- **Object-oriented data model**
- **Object Relational data model**
- **Semi-structured data model**

Example for Logical & Physical Model

Logical Model

student	name	num	Birth_year	addr	sex	dept
course	cnum	cname	credit			
sc	snum	cnum	grade			

Physical Model

```
CREATE TABLE student
 (    name     char(10),
      snum     int PRIMARY KEY ,
      birth_year      int,
      addr     char(15));
```

```
CREATE TABLE course
 ( cnum        int PRIMARY KEY ,
   cname       char(10),
   credit      int
 )
```

```
CREATE TABLE sc
 (    snum     int,
      cnum     int,
      grade    int,
      PRIMARY KEY (snum,cnum),
      FOREIGN KEY(snum)REFERENCES student(snum)  on delete cascade,
      FOREIGN KEY(cnum)REFERENCES course(cnum) on delete cascade,
      CHECK ((grade IS NULL) OR (grade BETWEEN 0 AND 100)));
```

Chapter 1 -Contents

E.g.

read the student information from keyboard and write them into a file named *stu_list*, and then read data from the file and output them on screen

Student

num	name	age	addr
S1	LI	18	Beijing
S2	WANG	19	Shang Hai
S3	LIU	21	Shan Dong
S4	CHEN	20	Si Chuan

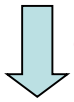
Main body of C program

```
Student information (Data Structure)
Struct student_type
  { char name [10];  int num, age;  char addr [15];} stud[SIZE];
 Main()
 {  for (i=0; i<SIZE; i++)
       scanf ( "%s%d%d%s", stud[i].name, &stud[i].num,
                    &stud[i].age, stud[i].addr);


    ……..

    for (i=0; i<SIZE; i++)
    { fread (& stud[i], sizeof ( struct student_type ), 1, fp );

      printf ("%-10s %4d %4d %-15s \n",
                 stud[i].name, stud[i].num, stud[i].age, stud[i].addr);
    }
 }
```

Change the data structure

Student

num	name	age	addr
S1	LI	18	Beijing
S2	WANG	19	Shang Hai
S3	LIU	21	Shan Dong
S4	CHEN	20	Si Chuan



change

Student

num	name	birth_Year	addr
S1	LI	1991	Beijing
S2	WANG	1990	Shang Hai
S3	LIU	1988	Shan Dong
S4	CHEN	1989	Si Chuan

Main body of C program

Student information (Data Structure)

Changed to birth_year

```
Struct student_type
  { char name [10];  int num, age;  char addr [15];} stud[SIZE];
 Main()
 {  for (i=0; i<SIZE; i++)
       scanf ( "%s%d%d%s", stud[i].name, &stud[i].num,
                  &stud[i].age, stud[i].addr);
```

Changed to birth_year

```
    ……..

    for (i=0; i<SIZE; i++)
    { fread (& stud[i], sizeof ( struct student_type ), 1, fp );

      printf ("%-10s %4d %4d %-15s \n",
                stud[i].name, stud[i].num, stud[i].age, stud[i].addr);
    }
}
```

DateTime.Year - stud[i].birth_year

1.4 Architecture of Database System

Schema

- The overall structure of the database is called the schema.
- It describes the types of DB, not the concrete values.
- It is relatively stable, whereas value is varying.

Instance

- The data in DB at any particular point in time is called a database instance.—the concrete value of a schema
- One schema can have many instances.
- Instance varies along with the modified data / flying time.

Example for schema and instance

Schema

Student (sno, sname, age, sex, Dept)

Instance

Freshmen in 2016

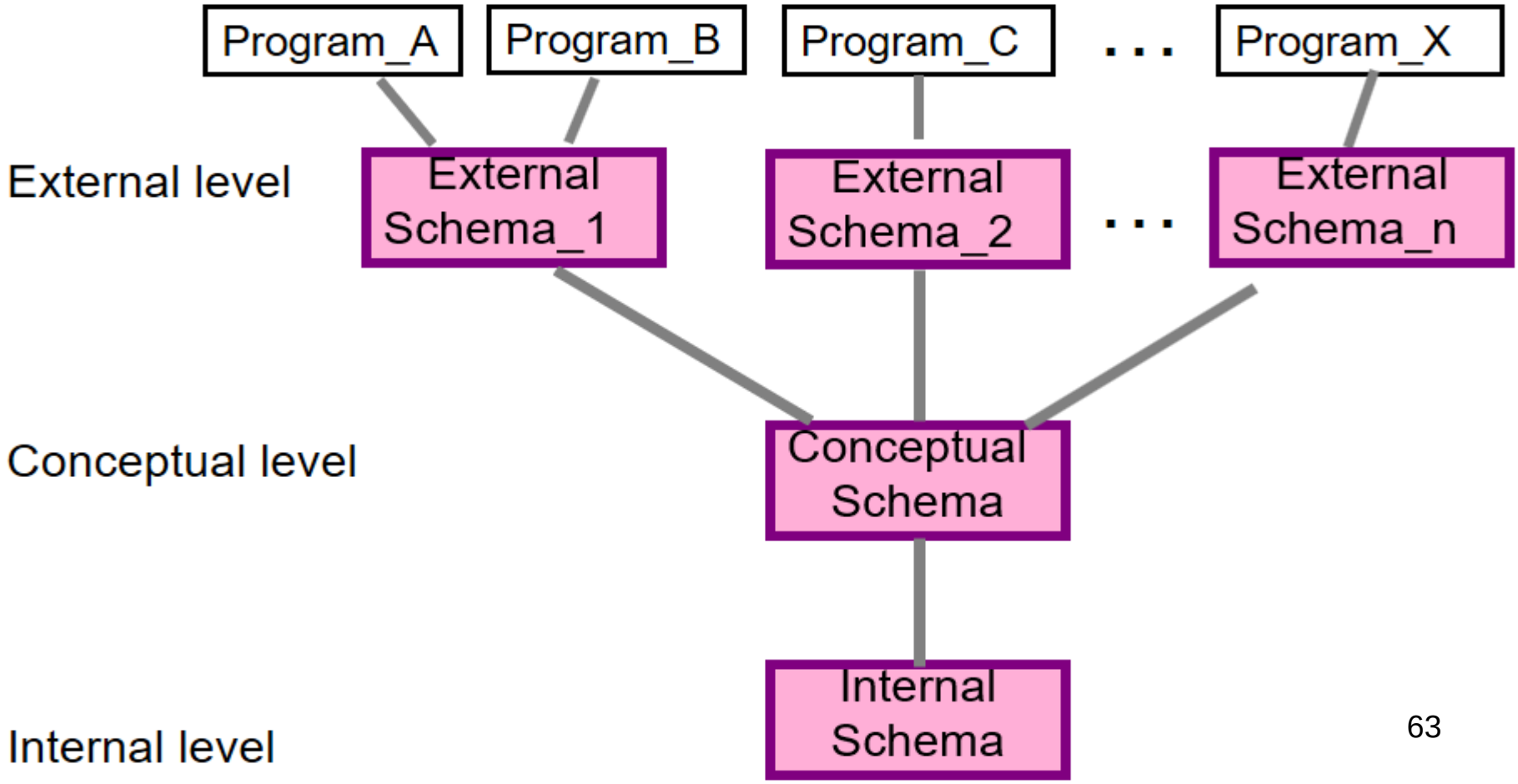
Sno	sname	age	sex	dept
1630001	李一山	19	男	软件
1630002	张立	20	男	软件
.....	.....	.....	.....	.....
1630120	赵萱	19	女	软件

Freshmen in 2017

Sno	sname	age	sex	dept
1730001	包拯	18	男	软件
1730002	王丽	19	女	软件
.....	.....	.....	.....	.....
1730120	张强	18	男	软件

ANSI-SPARC Three-Level Architecture

- **External Schema**
- **Conceptual Schema**
- **Internal Schema**



Conceptual Schema

Conceptual Schema（Logical schema, or Schema.）

- **It represents the global logical structure of the entire database, it is common data view of all users.**
- **Describes what data is stored in database and relationships among the data.**
- **It is independent of any storage considerations.**

The Conceptual Schema includes:

- **all entities, their attributes, and their relationships;**
- **the constraints on the data;**
- **semantic information about the data;**
- **security and integrity information.**

Conceptual Schema

Notes：

- Generally, there is only one schema in one DB.
- Schema is the center of the DB architecture.
- Schema does not consider the physical storage details and hardware.
- Schema has no concern for the application program, development tools and programming language.

External Schema

External Schema（User Schema or Subschema）

- **It is the Local logical structure of the database. It is users' view of the database.**
- **Describes that part of database that is relevant to a particular user.**
- **Some views might include derived or calculated data. E.g. age=current_year – birth_year**

Notes：

- Subschema is the subset of conceptual schema.
- One conceptual schema can have many subschemas.
- One subschema can be used by several application system, but one application program can use only one subschema.

Internal Schema

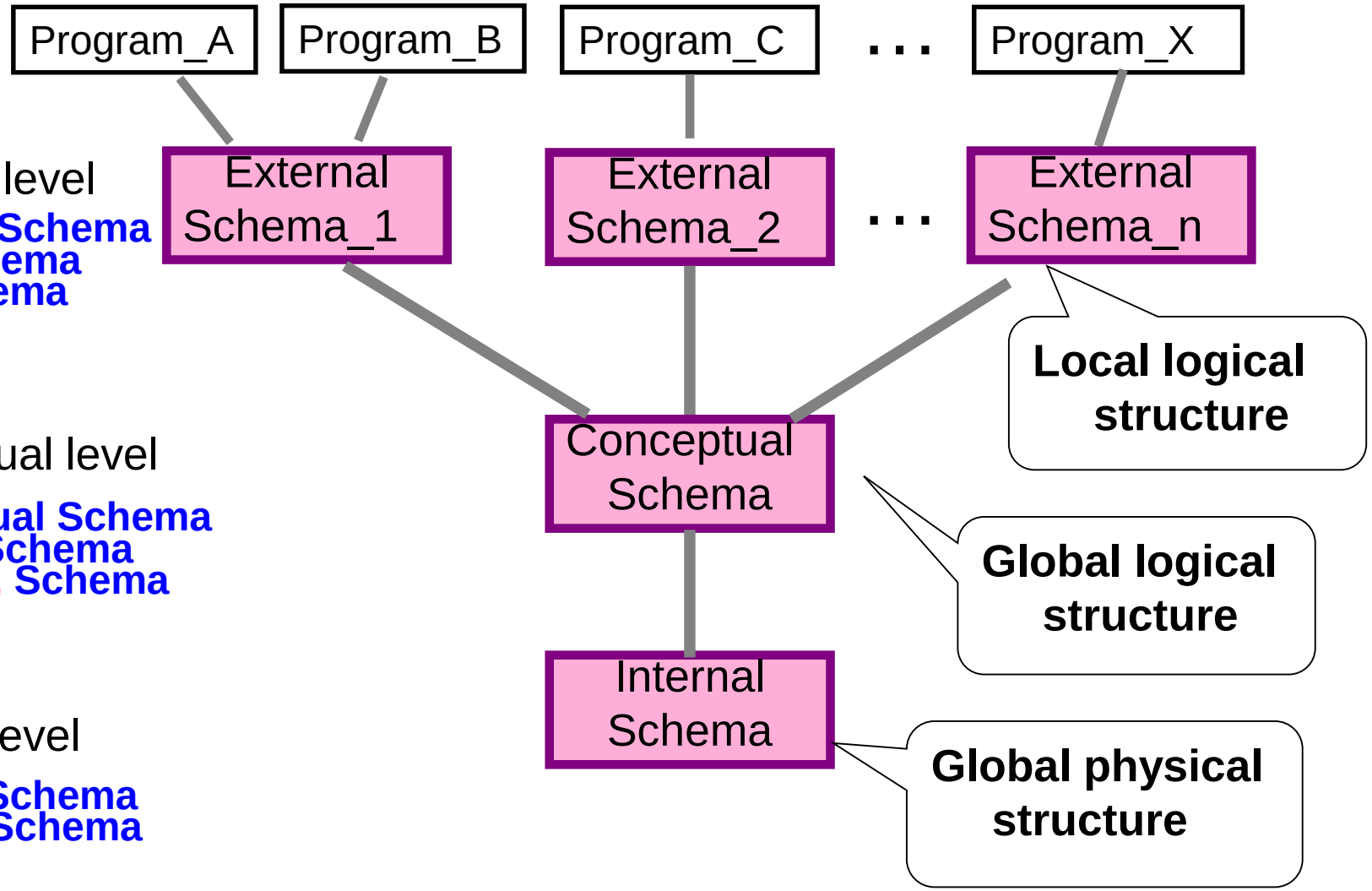
Internal Schema （Storage schema）

- **It is physical representation of the DB on the computer. It is the global physical structure of the DB.**
- **Describes how the data is stored in the database.**
- **One DB has only one Internal Schema.**

The internal level includes:

- **storage space allocation for data and indexes;**
- **record descriptions for storage (with stored sizes for data items)**
- **data compression and data encryption techniques.**

ANSI-SPARC Three-Level Architecture



Example: Differences between Three Levels of ANSI-SPARC Architecture

External Schema

view_1（dorm manager）

snum	sname	age	addr

view_2（academic staff）

snum	sname	cname	grade

Conceptual Schema

student

snum	sname	Birth_year	addr	sex

course

cnum	cname	credit

sc

snum	cnum	grade

Internal schema including index structure and access method

```
struct student
  { int snum;
    char sname [10];
    int Birth_year;
    char addr [15];
    char sex[1];
    struct student *next;
  };  index num;
```

```
struct course
  {int cnum;
    char cname [10];
    int credit;
    struct course *next;
  };  index cnum;
```

```
struct sc
  { int snum;
    int cnum;
    int grade;
    struct sc *next;
  };
 index snum; index cnum;
```

Objectives of Three-Level Architecture

View in DB is separated from the physical detail of the DB, its objectives are:

- **DBA should be able to change conceptual structure of database without affecting all users.**
  (e.g. addition/removal of entities).
- **DBA should be able to change database storage structures without affecting the users' views.**
  (e.g. change of index method).

In one word, upper levels are unaffected by changes to lower levels. It is called data independence.

1.5 Data Independence

◆Definition of data independence

- Independence between program and data
- Not independence between data and data

◆The definition of Database

DB is a shared collection of logically related data.

◆ Data Independence is to ensure that the change of lower-level schema should not affect the upper-level schema, then do not need to change the program.

◆Two kinds of Data Independence

- ●Logical Data Independence
- ●Physical Data Independence

◆ **Data Independence is guaranteed by two-level of mapping provided by DBMS.**

- ● **external/conceptual mapping**
- ● **conceptual/ internal mapping**

Logical Data Independence

- **Logical Data Independence**
  - **Refers to immunity of external schemas to changes in conceptual schema.**

  i.e. Conceptual schema changes (e.g. addition/removal of entities or attributes) should not require changes to external schema or rewrites of application programs.
- **Logical Data Independence is guaranteed by the mapping between external schema and conceptual schema（external/conceptual mapping）.**
- **The definition of external/conceptual mapping is described in the corresponding subschema.**

Physical Data Independence

- **Physical Data Independence**
  - **Refers to immunity of conceptual schema to changes in the internal schema.**

  i.e. Internal schema changes (e.g. using different file organizations, storage structures/devices) should not require change to conceptual or external schemas.
- **Physical Data Independence is guaranteed by the mapping between conceptual schema and internal schema （conceptual/ internal mapping）.**
- **The definition of conceptual/internal mapping is unique and is described in the conceptual schema.**

Example for Physical Data Independence

Internal Schema，Different index methods

Hashing -> B^+tree

The physical detail related to storage structure is encapsulated into DBMS. Changing IS will not affect CS and the programs do not need to be modified.

Select * from Student

Student

snum	name	sex	age
S1	LI	M	17
S2	WANG	F	19
S3	LIU	F	21
S2	CHEN	M	20

Example for Logical Data Independence

Problem：How to solve the problem in C program?

Student

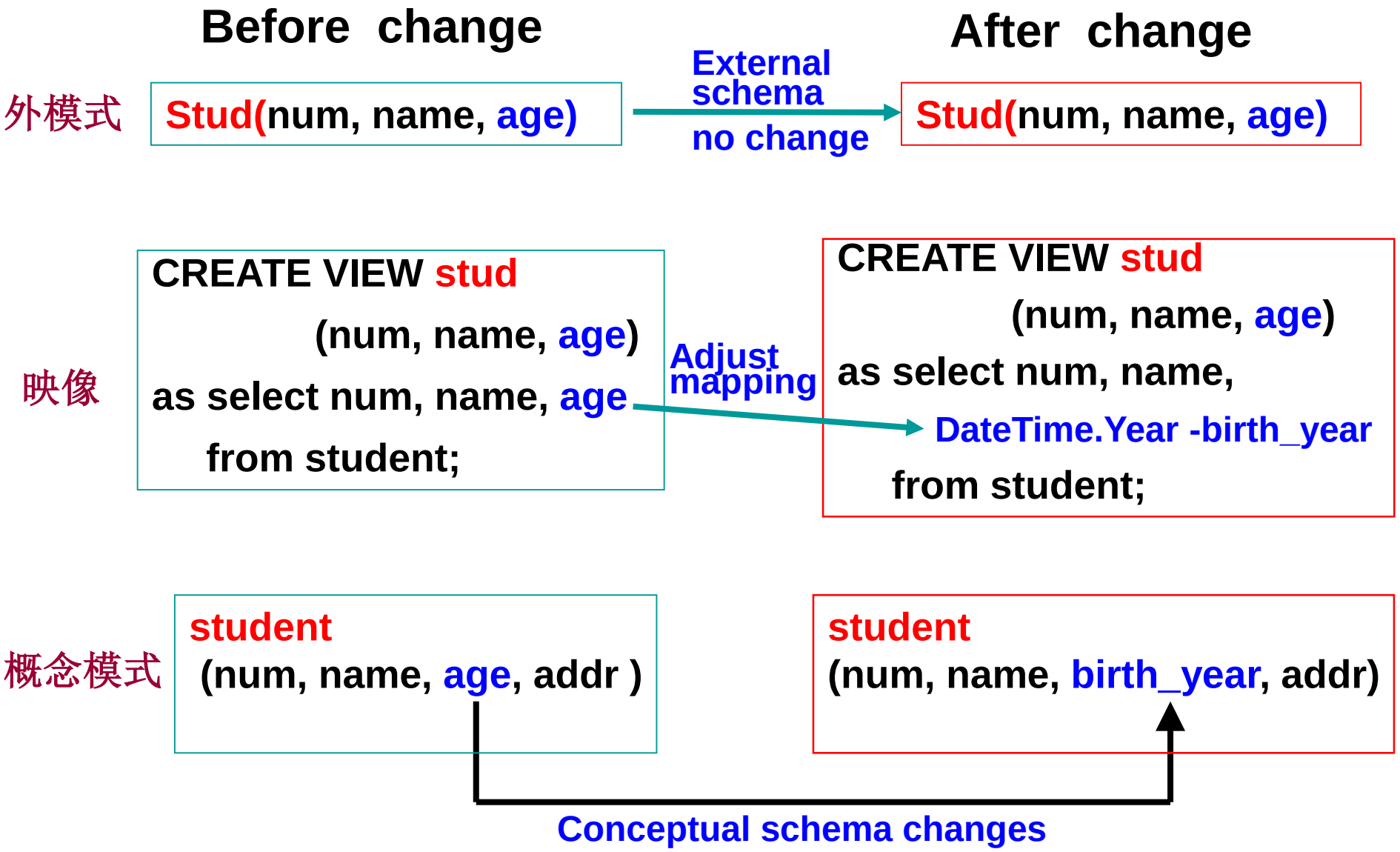
num	name	age	addr
S1	LI	18	Beijing
S2	WANG	19	Shang Hai
S3	LIU	21	Shan Dong
S4	CHEN	20	Si Chuan

change

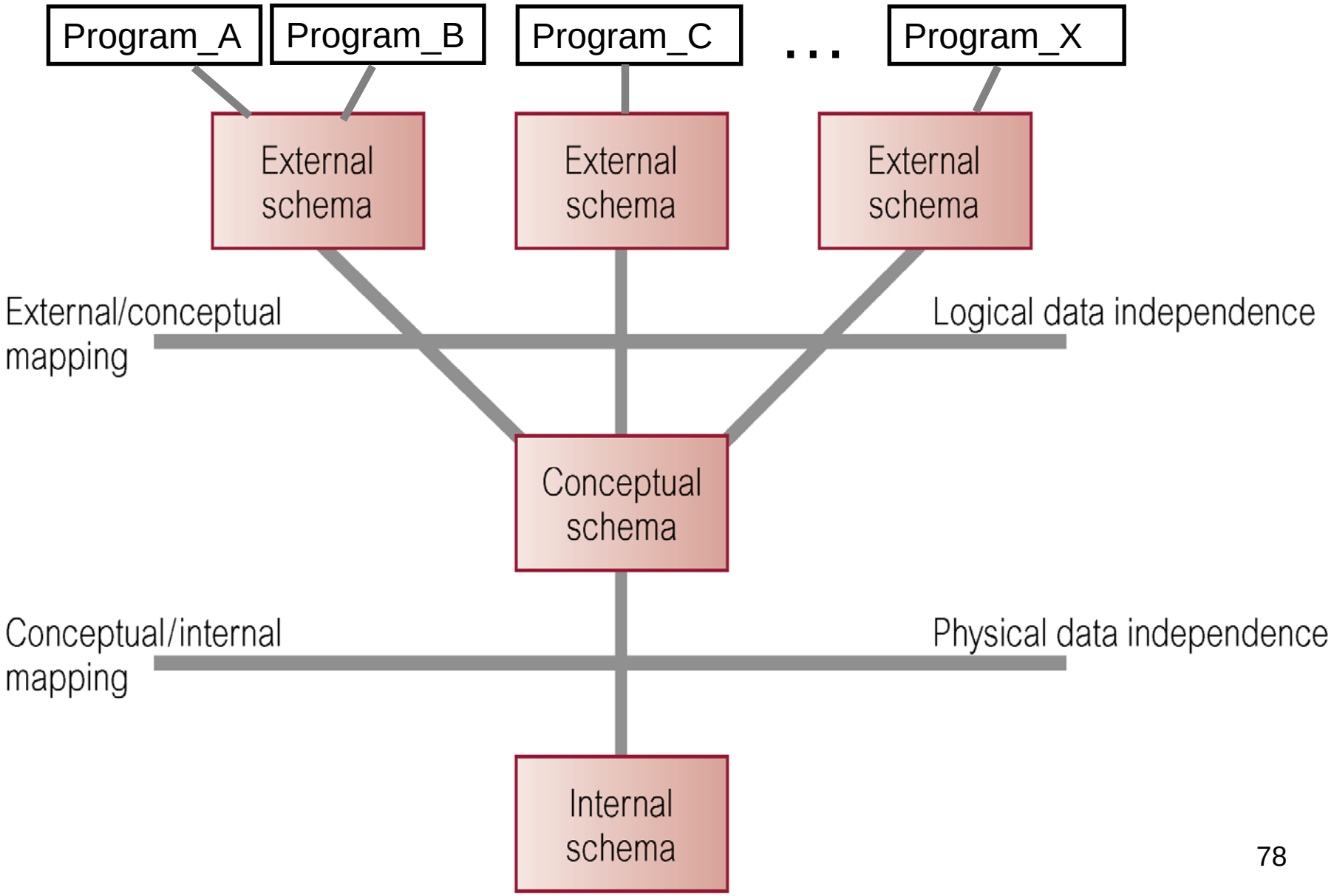
Student

num	name	birth_Year	addr
S1	LI	1991	Beijing
S2	WANG	1990	Shang Hai
S3	LIU	1988	Shan Dong
S4	CHEN	1989	Si Chuan

Problem：program depends on data


Before change
After change
外模式
Stud(num, name, age)
External schema
no change
Stud(num, name, age)
映像
CREATE VIEW stud
(num, name, age)
as select num, name, age
from student;
Adjust mapping
CREATE VIEW stud
(num, name, age)
as select num, name,
DateTime.Year -birth_year
from student;
概念模式
student
(num, name, age, addr)
student
(num, name, birth_year, addr)
Conceptual schema changes

Data Independence and the ANSI-SPARC Three-Level Architecture



Data Independence and the ANSI-SPARC Three-Level Architecture

(1) **Internal Schema** describes the Global Physical Structure of DB.

(2) **Conceptual Schema** describes the Global Logical Structure of DB.

(3) **External Schema** describes the Local logical structure of the database.

(4) The **external/conceptual** mapping provides the Logical Data Independence.

(5) The **conceptual / internal** mapping provides the Physical Data Independence.

Difference between Conceptual Schema and Conceptual Model

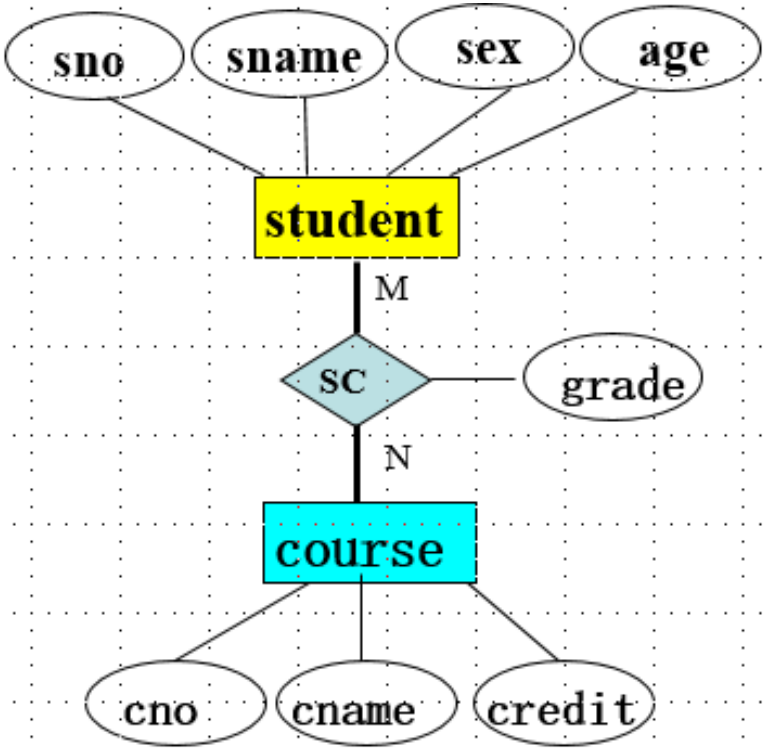
Conceptual Schema describes the whole/global logical structure of DB using a lower-level DDL. It is just logical model. It belongs to Machine World.

Conceptual Model describes the whole/global logical structure of DB using a higher-level method which is easy for users to understand. It is a model independent to computer system, which is abstracted by human brain. It belongs to Information World and is not related to a specific DBMS. It has clear semantic meaning. It is a communication tool between system designers and users.

Their common lies in that they describe the same objects. Difference is that their description methods are different.

Example of Conceptual Schema and Conceptual Model

Conceptual Model



Conceptual Schema

student	snum	sname	Birth_year	addr	sex

course	cnum	cname	credit
sc	snum	cnum	grade

describe the same objects by different tools in different world (abstract level)

The position of Conceptual Model in schemas of DB