



物联网与泛在智能研究中心

# 数据库基础篇

## 第一章 数据库系统概述

主讲：张 浩





# Outline

- 数据、数据库
- 数据库管理系统及其结构
- 数据库系统
- 数据抽象与数据模型
- 数据库系统的发展





# Outline

- 数据、数据库
- 数据库管理系统及其结构
- 数据库系统
- 数据抽象与数据模型
- 数据库系统的发展





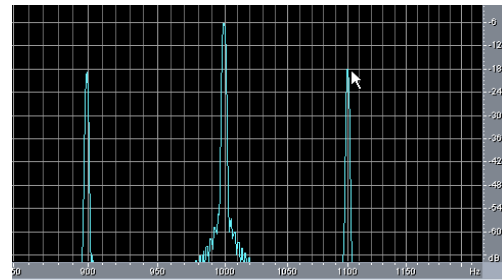
# 什么是数据?

- 数据是描述现实世界中各种具体事物或抽象概念的可存储编码，是信息的载体

- 人的姓名、城市温度、学生成绩
- 数据与其语义是不可分的

- 数据的类型

- 数字、字符串、日期
- 逻辑值、文本、图形
- 图像、声音
- .....



<i>Name</i>	<i>Age</i>	<i>Phone</i>	<i>Address</i>
Smith	20	8798677	London
Jones	21	8899966	Paris
Black	23	2348765	London
Clark	20	5623894	Athens
Adams	25	3568209	Paris





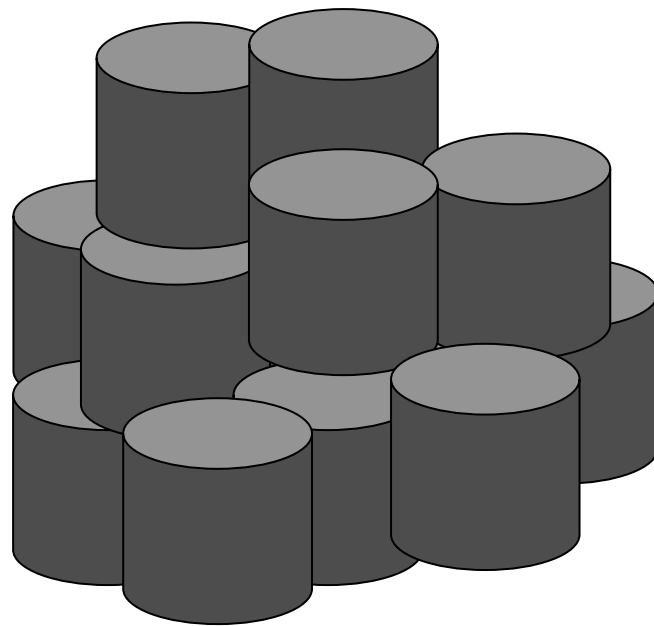
# 什么是数据库?

- 数据库

- 一个互相关联的数据的集合
- 是长期储存在计算机内、有组织的、可共享的

- 数据库的特征

- 数据按一定的数据模型组织、描述和储存
- 可为各种用户共享
- 冗余度较小
- 数据独立性较高
- 易扩展





# 什么是数据库?

- 数据库的类型

- 简单结构数据库：如关系数据库
- 复杂结构数据库：如图数据库
- 半结构化数据：如XML数据库
- 非结构化数据：如多媒体数据库





# 数据库目的

■ 把现实世界  
映射到计算  
机世界

■ 支持各种实  
际应用





# 数据库模式与数据库实例



Grade\_Report

姓名	课程编号	成绩	学期	年度
常红	计 14	86	一	1994
常红	计 15	93	一	1994
都薇	数 1	89	二	1995
都薇	数 13	90	一	1995
都薇	计 14	85	一	1995

Prof-Course

教师名	课程编号
刘德祥	数 01
刘德祥	数 13
陈庆奎	计 15
李金宝	计 17
孙文隽	计 14







- 数据、数据库
- 数据库管理系统及其结构
- 数据库系统
- 数据抽象与数据模型
- 数据库系统的发展





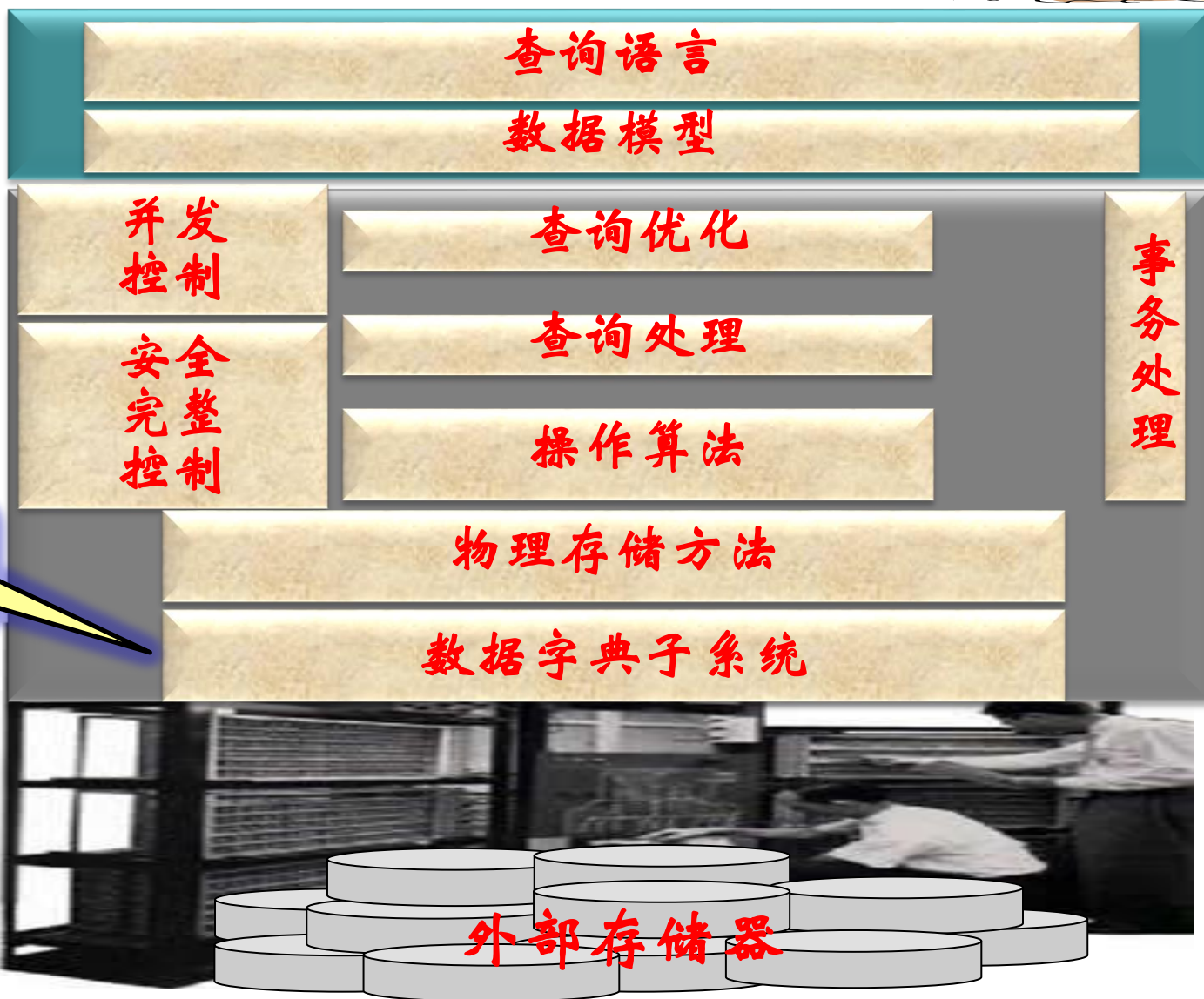
# 什么是数据库管理系统?

- 数据和数据库的概念，下一个问题就是如何科学地组织和存储数据，如何高效地获取和维护数据。完成这个任务的是一个系统软件——数据库管理系统
- 数据库管理系统 (DataBase Management System, DBMS)
  - 管理数据库的软件系统，包括存储管理、安全性管理、完整性管理等。使用户能方便快速地建立、维护、检索、存取和处理数据库中的信息
- 数据库管理系统的系统结构(如下)





# 数据库管理系统结构





# 数据库管理系统的功能与特点

- 有效地支持数据抽象与数据定义
  - 具有坚实的数据模型基础
  - 支持数据的不同视图
- 有效地组织存储数据
  - 提供丰富的数据存取方法
- 有效地处理数据查询与更新
  - 提供有效的数据操纵语言、算法
  - 优化地处理数据查询
- 支持数据独立性
  - 数据库系统在某一层次模式上的改变不会使它的上一层模式也发生改变的能力。





# 数据库管理系统的功能与特点

- 控制数据冗余

- 综合考虑所有用户的数据库视图，把它们集成为一个逻辑模式，数据项只存储1次或少数几次

- 支持数据共享/并发控制

- 允许多个用户或多个应用程序同时访问数据库中的相同数据，即允许数据共享
- 为了支持数据共享，数据库管理系统具有并发控制机制

- 限制非授权的存取

- 为了保证数据库的安全，防止对数据库的非法存取，数据库系统具有一个安全与授权子系统





# 数据库管理系统的功能与特点

- 提供多种用户界面
  - 图形界面，查询语言界面，程序设计语言界面
- 表示数据之间的复杂联系
  - 数据间联系的定义机制
  - 通过数据间联系查询数据的机制
- 支持完整性约束
  - 数据库应用对数据语义一般都有一定的限制，称为完整性约束
- 有效的数据恢复与事务处理
  - 在系统硬件或软件发生故障时，能够保证数据库的正确性





- 数据
- 数据库
- 数据库管理系统及其结构
- **数据库系统**
- 数据抽象与数据模型
- 数据库系统的发展





# 什么是数据库系统?

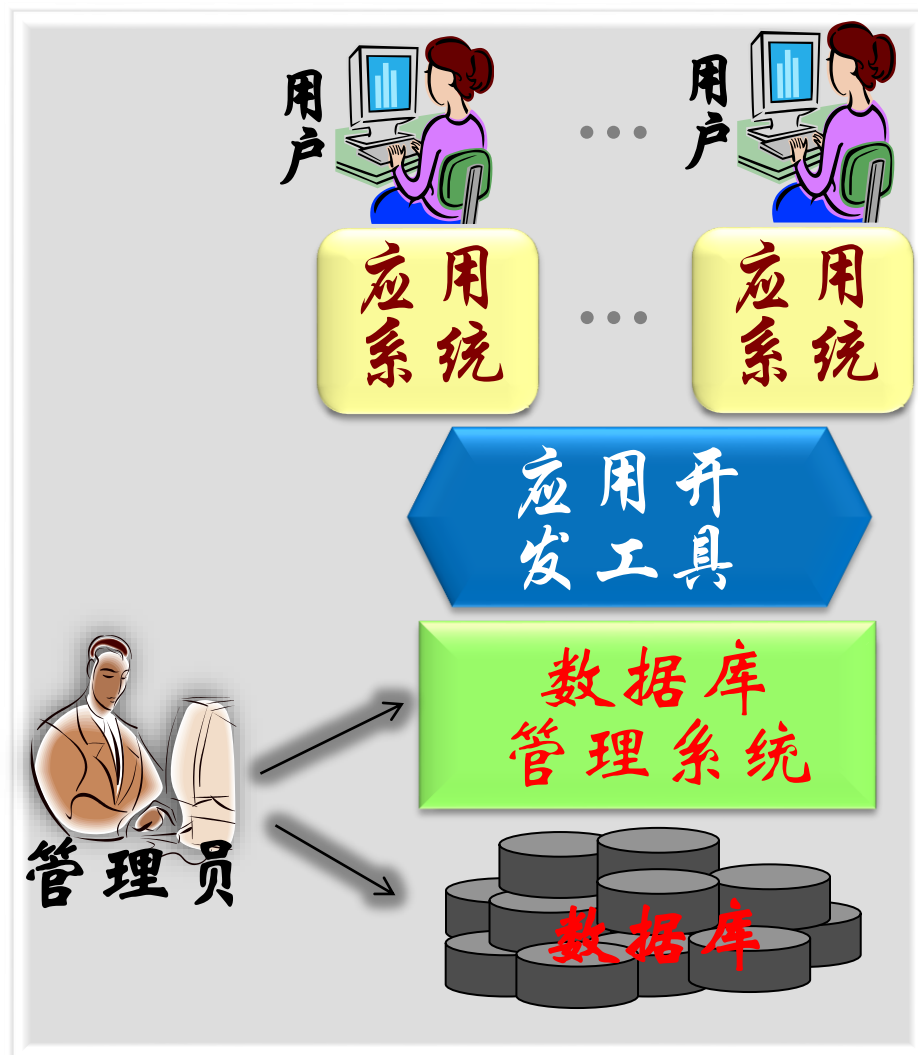
- 数据库系统

- 观点一:

- 数据库
    - 数据库管理系统

- 观点二:

- 数据库
    - 数据库管理系统
    - 数据库管理员
    - 应用开发工具
    - 应用系统
    - 数据库系统用户
      - 数据库管理员
      - 数据库设计员
      - 应用程序员
      - 最终用户







# 数据库系统 VS 文件系统

- 文件处理系统(file-processing system)
  - 传统操作系统所支持的、
  - 数据信息永久存储在多个不同的文件中

财务处

学号	姓名	系别	补贴
----	----	----	----

宿管科

学号	姓名	性别	系别	住址
----	----	----	----	----

教务处

学号	姓名	系别	学分	学位
----	----	----	----	----

学生处

学号	姓名	性别	系别	年龄	学位	出身
----	----	----	----	----	----	----

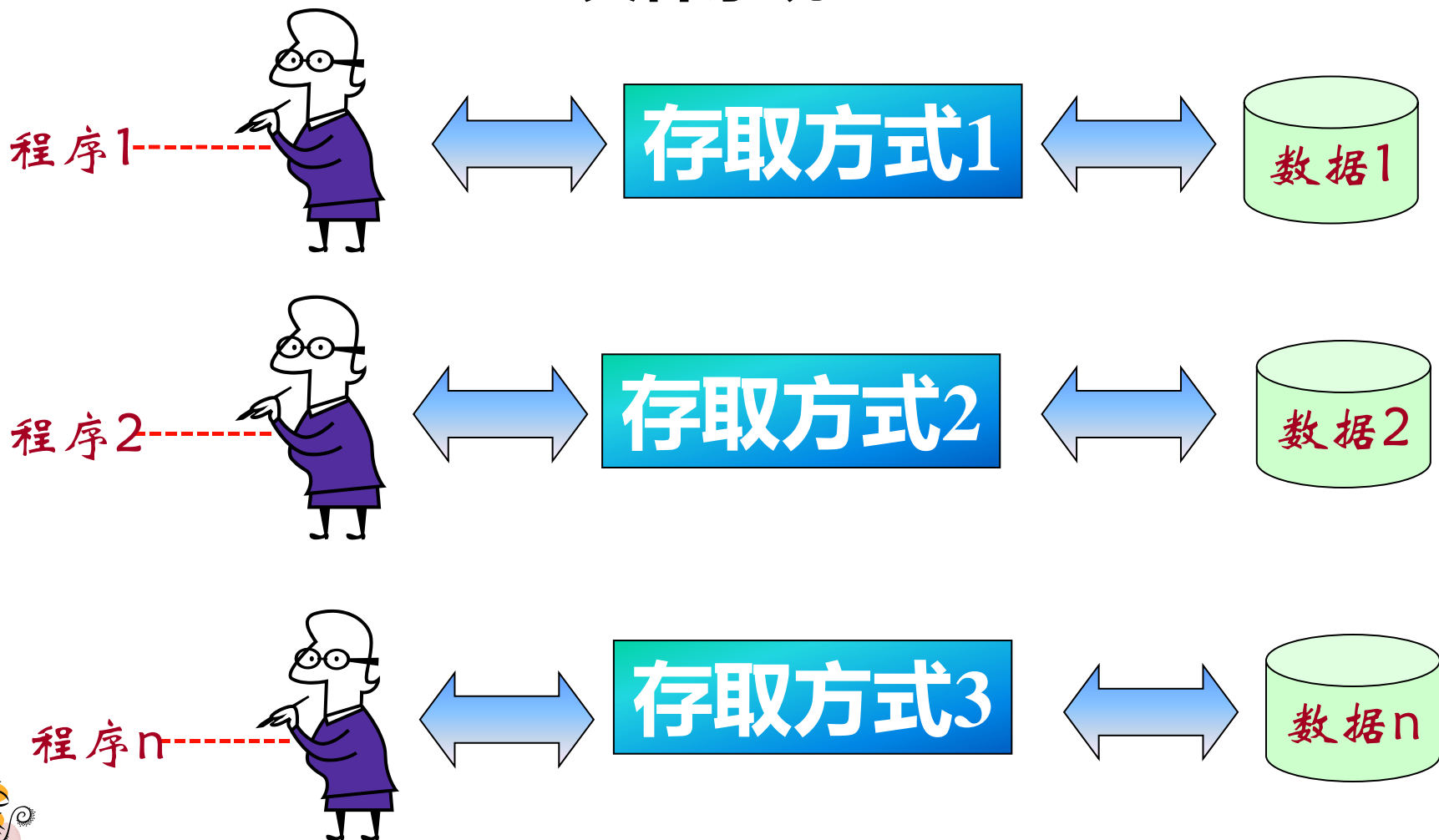
不同部门拥有各自独立的数据文件





# 数据库系统 VS 文件系统

## 文件系统





# 数据库系统 VS 文件系统

- 文件系统

- 数据冗余和不一致

- 数据访问困难

- 数据孤立

- 完整性问题

- 原子性问题

- 并发访问异常

- 安全性问题

- 并非数据库系统的所有用户都可以访问所有数据

- 由于文件系统中应用程序总是即席地加入到文件处理系统中来，这样的安全性约束难以实现



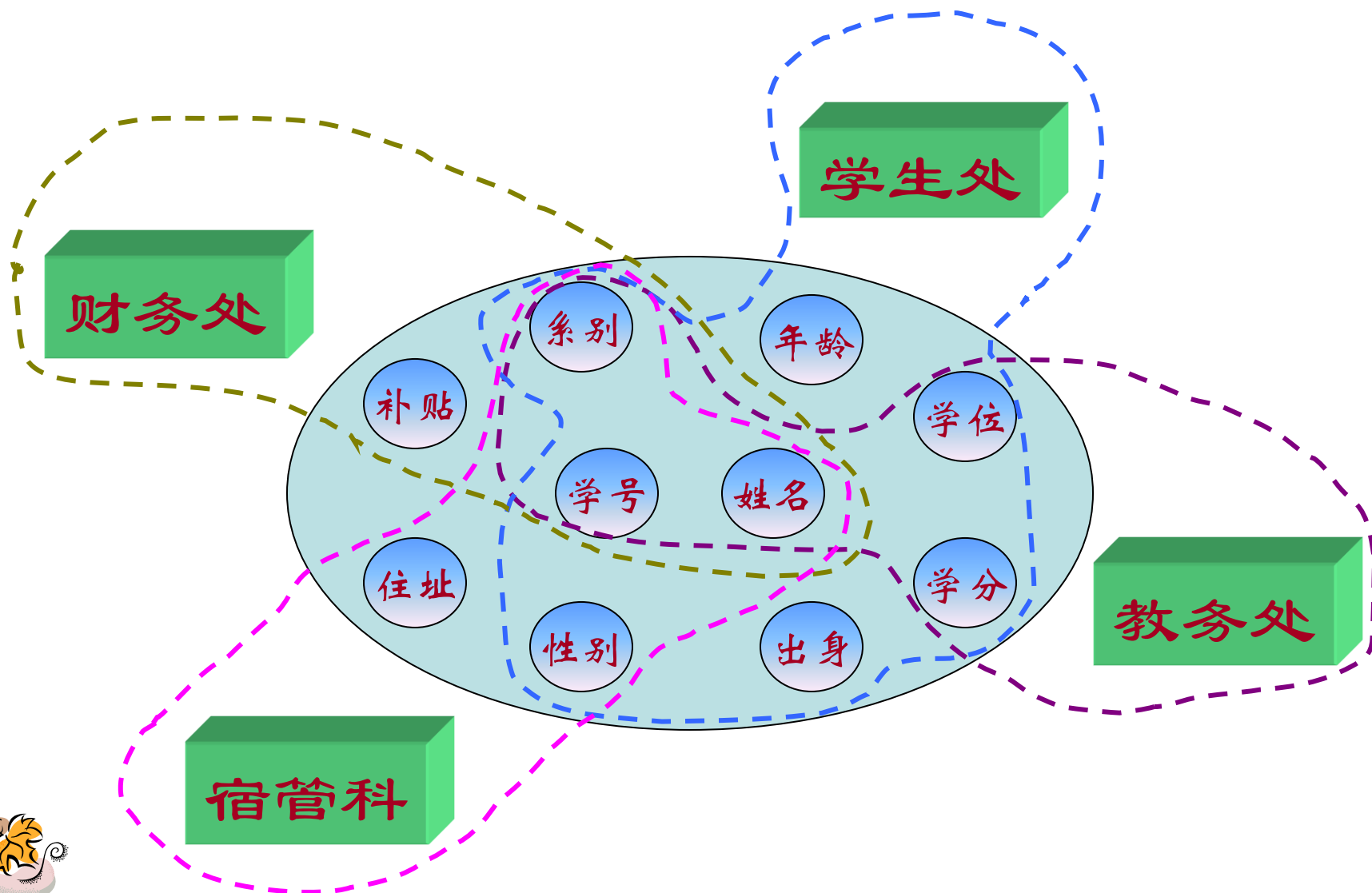
# 数据库系统 VS 文件系统

- 数据库系统(DB+DBMS+……)
  - 按照某种数据模型，将全部门的各种数据组织成一个结构化的数据集合中，整个部门的所需数据不是一盘散沙，可表示出数据之间的有机关联。





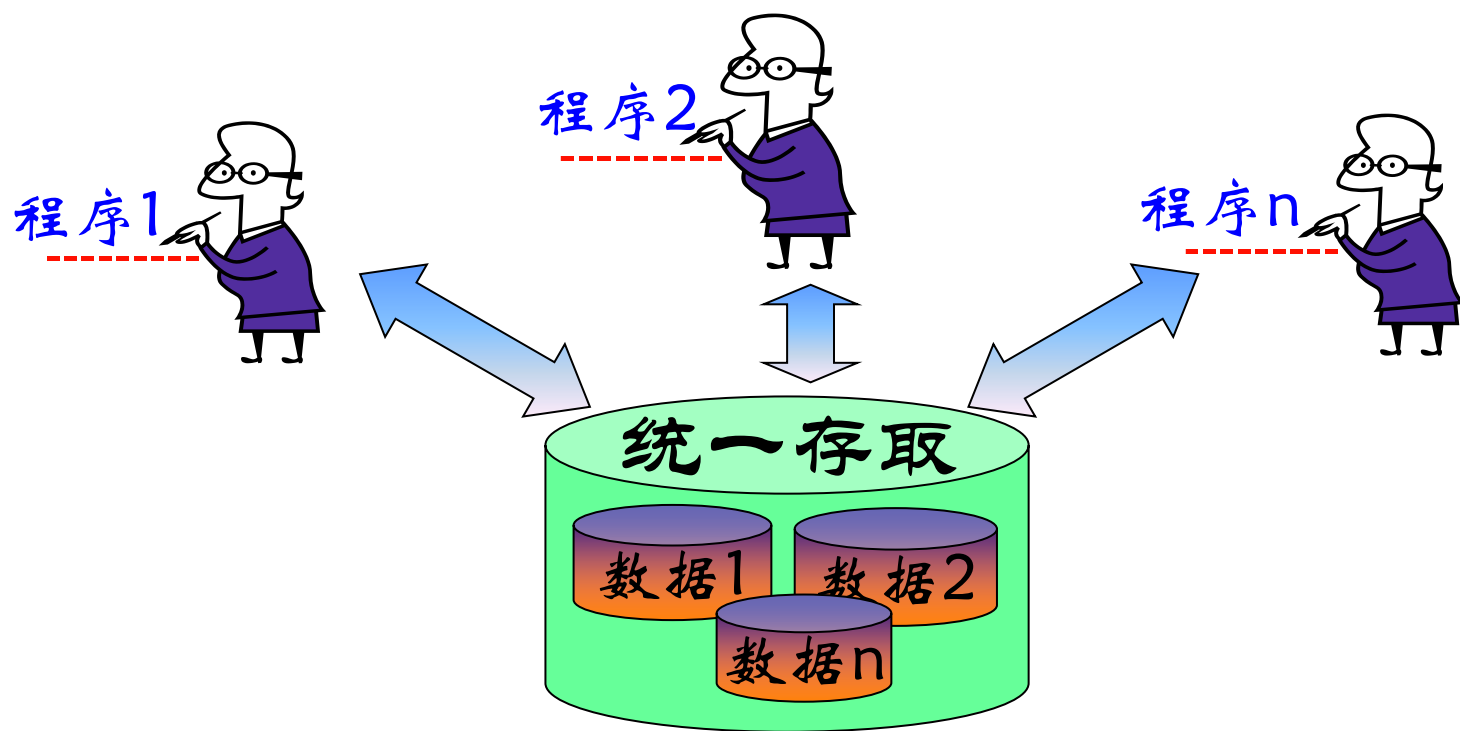
# 数据库系统





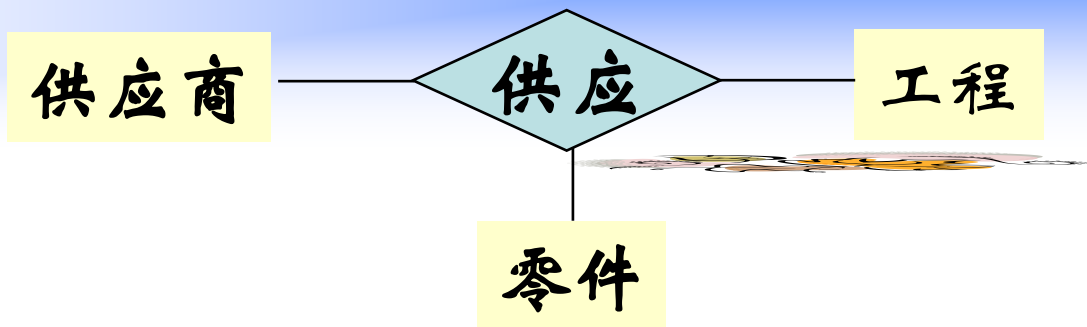
# 数据库系统

信息完整、功能通用（数据字典、元数据）





# 示例



供应商(SNO, SNAME, STATUS, CITY):表示供应商  
属性依次为供应商号, 供应商名, 供应商状态值, 供应商所在城市

零件(PNO, PNAME, COLOR, WEIGHT, CITY):表示零件  
属性依次为零件号, 零件名, 零件颜色, 零件重量, 零件存放的城市

工程(JNO, JNAME, CITY):表示工程

属性依次为工程号, 工程名, 工程所在城市

供应关系(SNO, PNO, JNO, QTY):表示供货关系

属性依次为供应商号, 零件号, 工程号, 供货数量

查询:供应红色零件给北京的工程的供应商名字?



# 示例——基于文件系统

- 分别组织几个文件，存储各类对象的记录

CreateFile(供应商, 零件, 工程, 供应关系)

ScanFile(工程)

找到北京的工程的号码

ScanFile(零件)

找到红色零件的号码

ScanFile(供应关系)

找到对应以上两号码的SNO

ScanFile(供应商)

找到对应以上SNO的供应商姓名

查询: 供应红色零件给北京的工程的供应商名字?







# 示例——基于数据库系统

查询:供应红色零件给北京的工程的供应商名字?

- 数据统一按表结构存放
  - 设表为供应商S, 零件P, 工程J以及供应关系SPJ
- 查询:只需提交查询要求,由系统完成查询过程

SELECT SNAME

FROM S, P, J, SPJ

WHERE SPJ.SNO = S.SNO AND SPJ.PNO = P.PNO

AND SPJ.JNO = J.JNO

AND J.CITY = "BEIJING"

AND P.COLOR = "RED"

声明性的、  
非过程的  
语言





# 数据库系统 VS 文件系统

## 文件系统

- 数据冗余、不一致
- 访问困难、独立性差
- 存在完整性、原子性问题
- 并发访问异常
- 存在安全性问题

## 数据库系统

- 支持数据共享
- 控制数据冗余
- 数据完整一致
- 数据独立性高
- 支持复杂结构
- 三种数据抽象
- 正确安全可靠





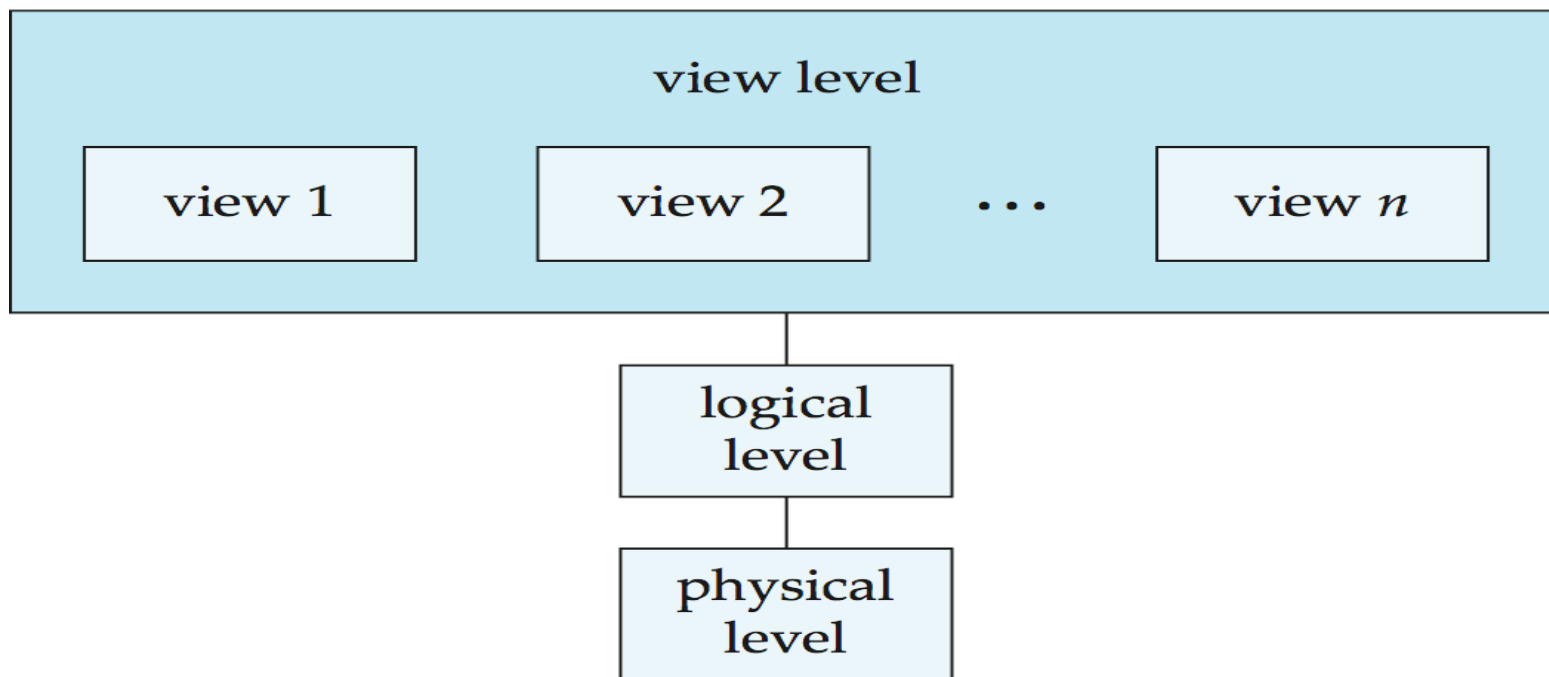
- 数据
- 数据库
- 数据库管理系统及其结构
- 数据库系统
- 数据抽象与数据模型
- 数据库系统的发展





- 数据抽象的作用

- 对于用户系统地隐藏关于数据存储和维护的某些细节
- 屏蔽复杂性，简化用户与系统的交互



数据抽象的三个层次



## • 视图抽象

- 最高层次抽象
- 把现实世界信息按不同用户观点抽象为多个逻辑数据结构，每个逻辑结构称为一个视图，
- 每个视图称之为数据库的一个子模式

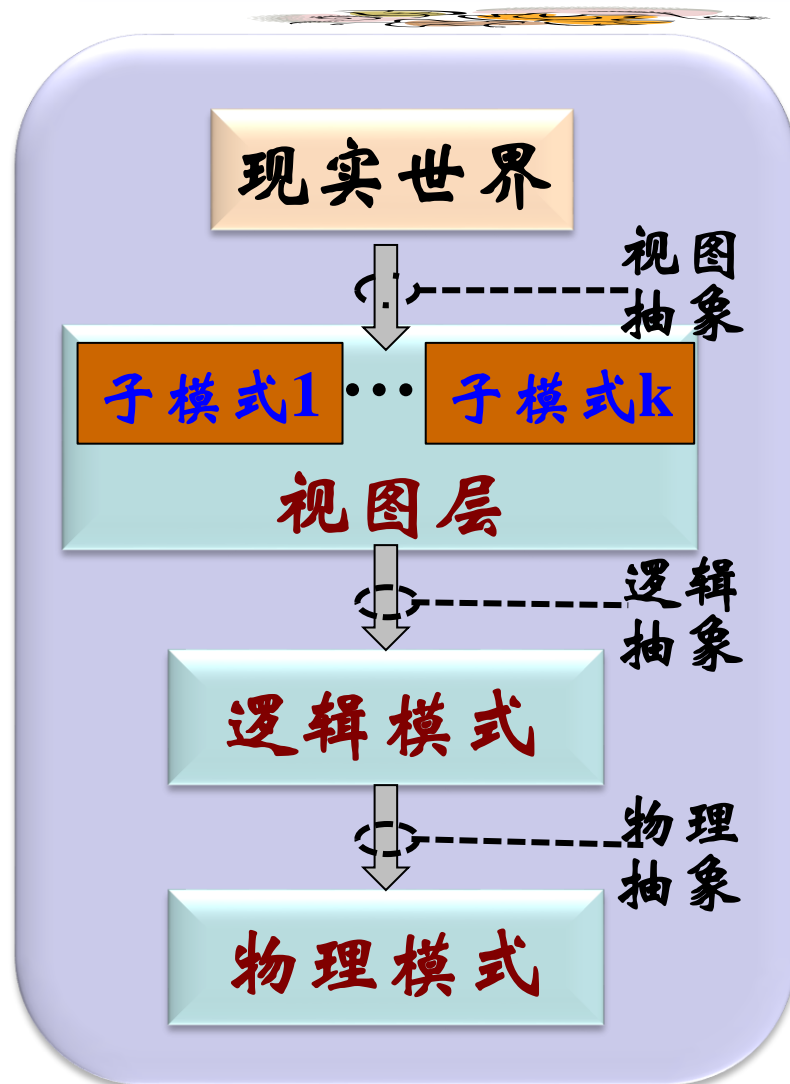
## • 逻辑抽象

- 综合所有视图，把所有用户关心的现实世界抽象为逻辑模式
- 描述数据库存储什么数据及这些数据间存在的关系

## • 物理抽象

- 最低层抽象
- 把逻辑模式抽象成为数据库的物理模式，
- 描述数据实际上是怎样存储的

# 数据抽象





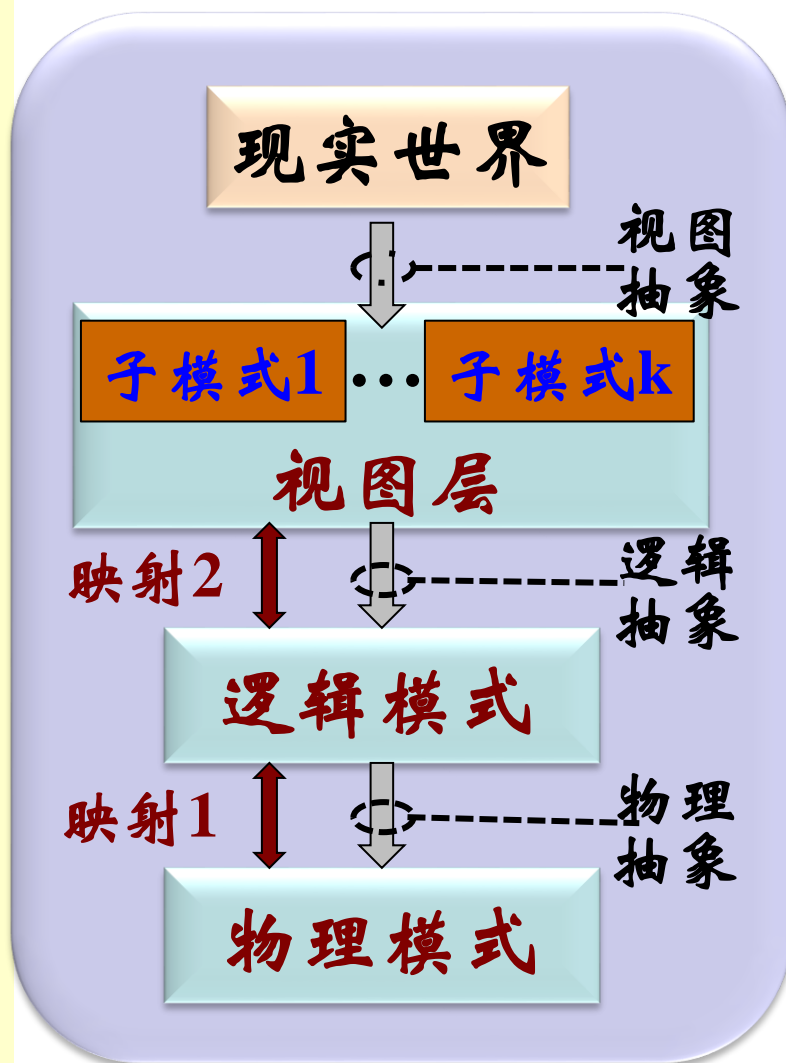
# 数据独立性

- 物理数据独立性

- 由物理模式/逻辑模式映射实现
- 数据库物理模式发生改变时
  - 仅需修改物理模式/逻辑模式映射
  - 数据的逻辑结构不变
  - 应用程序可以不变

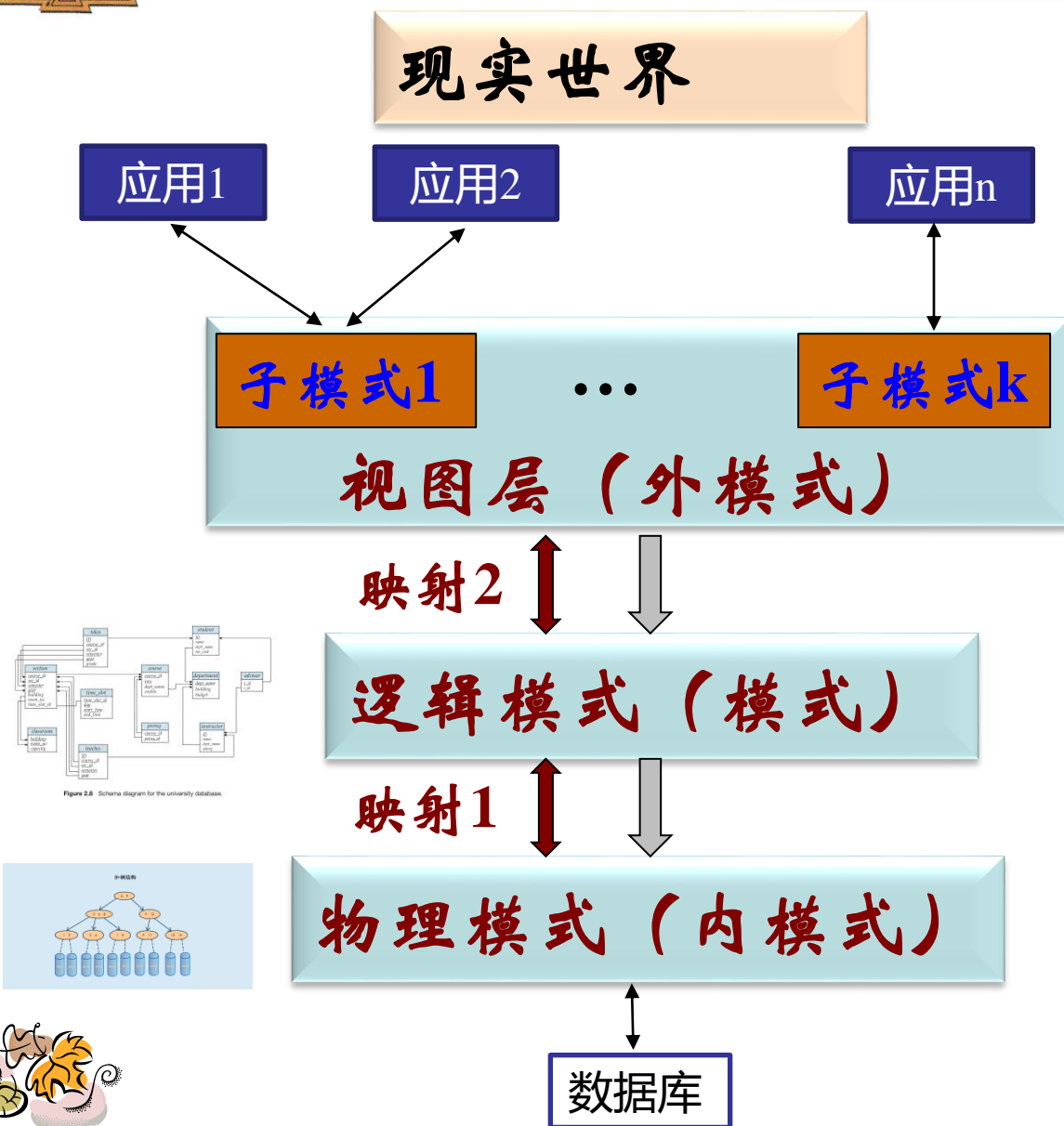
- 逻辑数据独立性

- 由逻辑模式/视图子模式映射实现
- 当逻辑模式发生改变时
  - 仅需修改逻辑模式/视图子模式映射
  - 数据库的视图子模式不变
  - 应用程序可以不变





# 映射与数据独立性



逻辑数据独立性：  
逻辑模式改变，只需要修改映射2，**子模式不变**（应用不变）

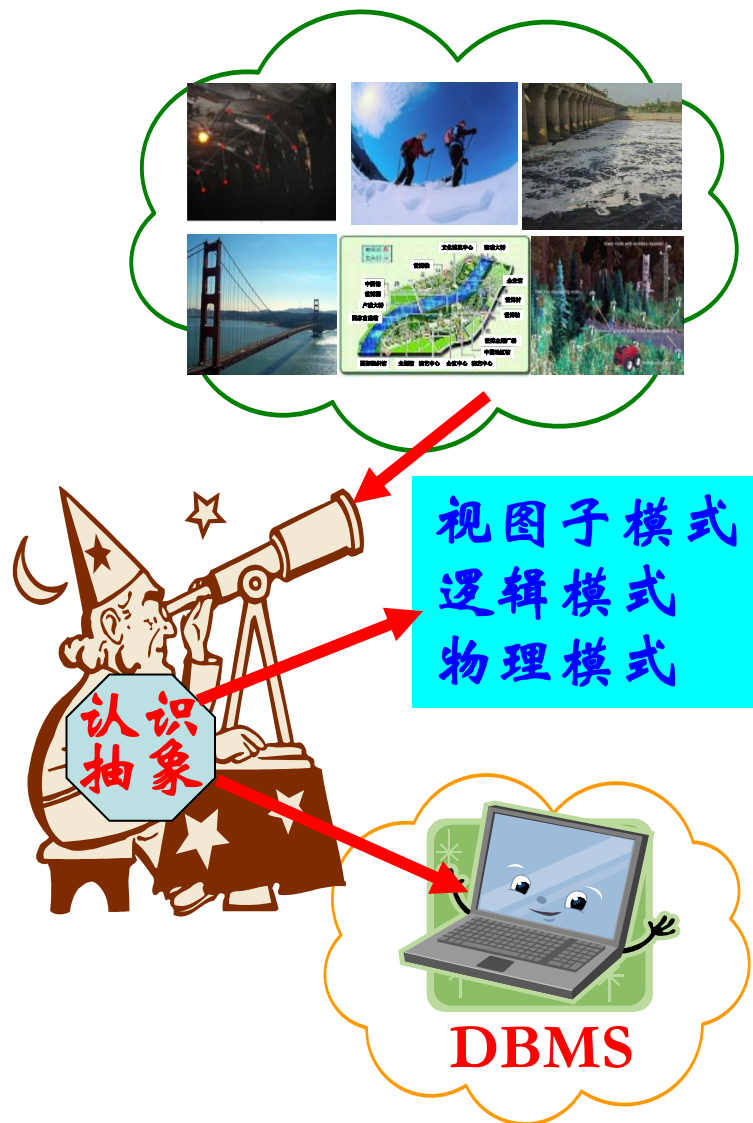
物理数据独立性：  
物理模式改变，只需要修改映射1，**逻辑模式不变**





# 数据模型

- 是实现数据抽象的工具
  - 描述现实世界对象数据
  - 描述现实世界对象的关系
  - 描述数据的完整性约束
  - 描述数据操作集合
- 是数据库系统关键, 决定
  - 数据库结构
  - 数据库的设计方法
  - 数据库管理系统设计实现
  - 数据定义语言(DDL)和操纵语言(DML)
- 数据模型的必要条件
  - 真实地模拟现实世界
  - 容易为用户所理解
  - 便于在计算机上实现







# 数据模型的基本要素

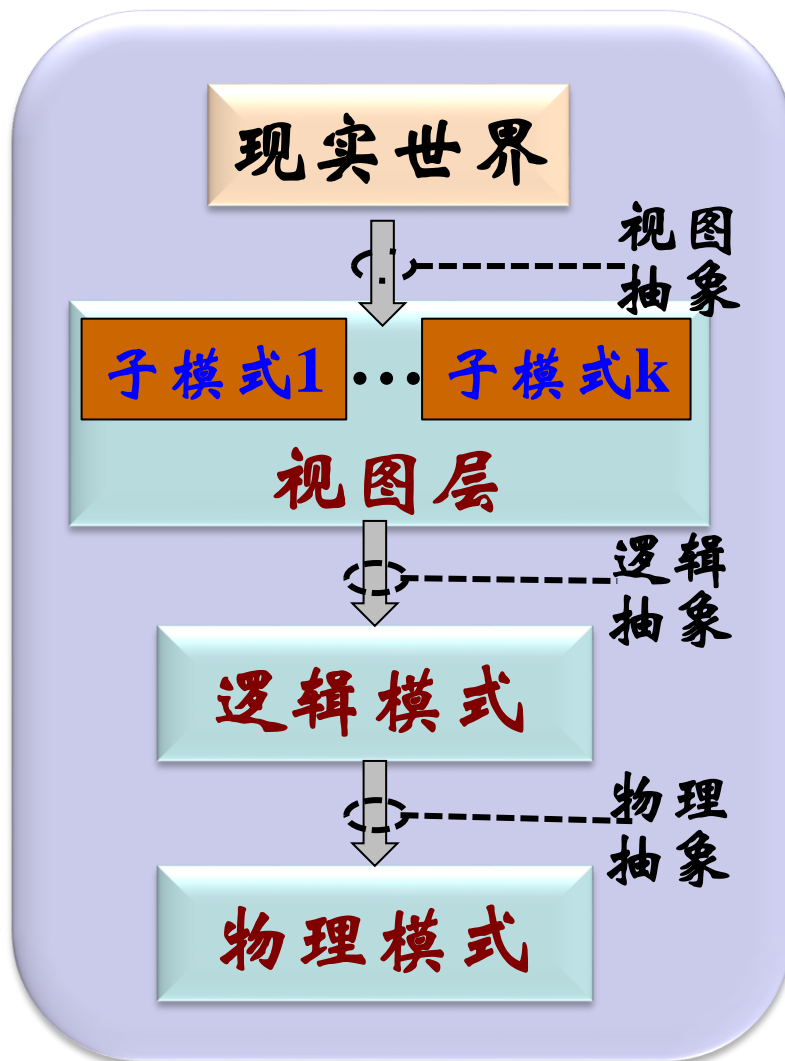
- 数据结构
  - 描述现实世界对象的信息结构
  - 对象的每个属性的数据类型、长度等
  - 描述对象之间联系的信息结构
- 数据操作
  - 数据定义操作
  - 数据查询、更新操作
- 数据的完整性约束
  - 完整性规则的集合
  - 规定了数据必须遵守的语义约束条件





# 常用的数据模型

- 实体关系 (E-R) 数据模型
  - 基于现实世界，用于视图抽象和概念数据库设计
- 面向对象 (O-O) 数据模型
  - 可以在E-R模型中增加了封装、函数和对象标识等概念扩展(22章)
- 关系数据模型
  - 用于逻辑抽象和逻辑模式定义
  - 用表的集合来表示数据和数据间的联系
- 对象关系 (O-R) 数据模型
  - 结合了第2、3中模型特征(22章)
- 层次和网络数据模型
  - 与底层实现联系过于紧密





# 层次数据模型

- 层次数据模型的数据结构
- 层次数据模型的数据操作
- 层次数据模型的完整性约束
- 层次数据模型的优缺点
- 具有代表性的层次数据库系统





# 层次数据模型的数据结构

- 满足下面两个条件的**基本层次联系**的集合为**层次模型**
  - 有且只有一个节点没有父亲节点，这个节点称为根节点
  - 根以外的其它节点有且只有一个父亲节点
- 层次模型中的几个术语
  - 根节点，父亲节点，兄弟节点，叶节点

树状

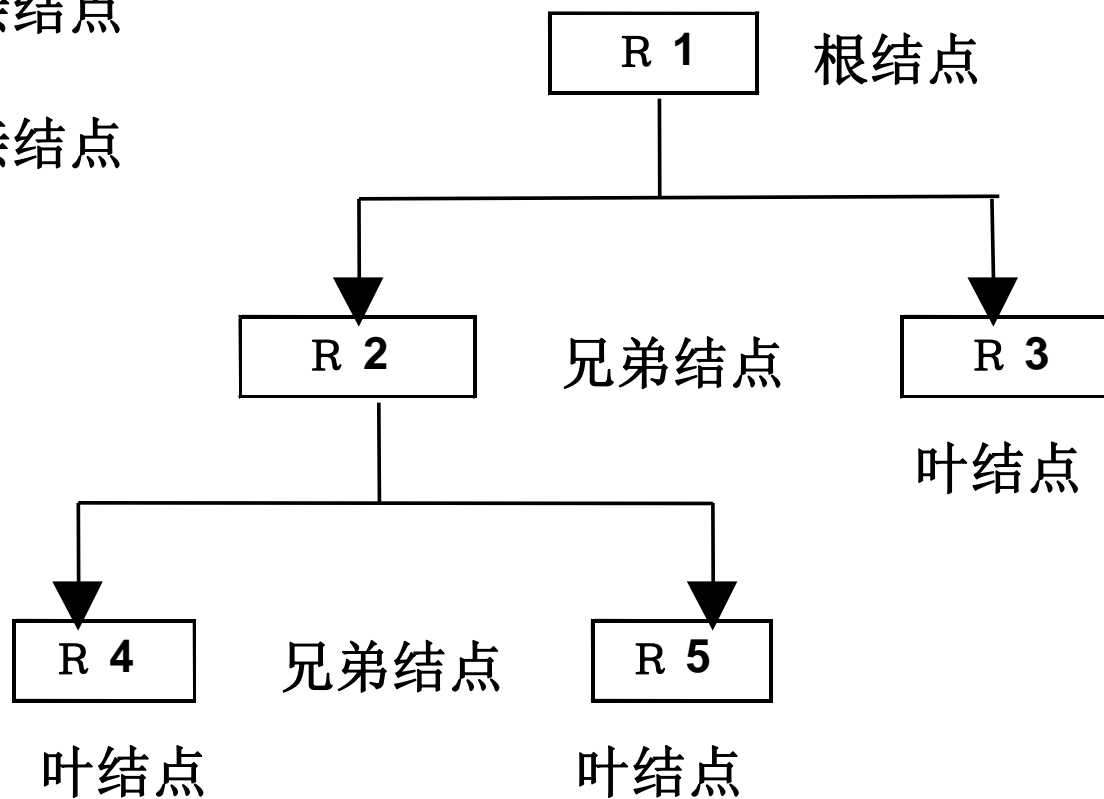




# 层次数据模型的数据结构

R1为R2和R3的双亲结点

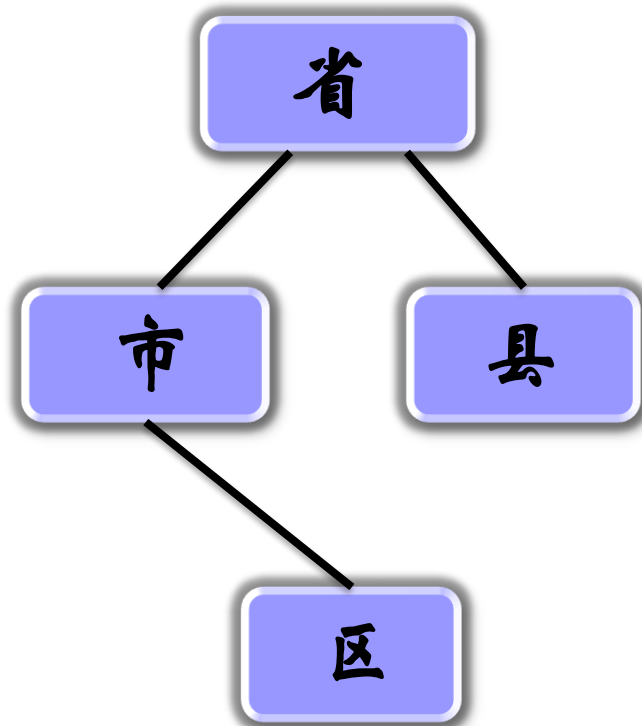
R2为R4和R5的双亲结点





# 层次数据模型的数据结构

- 层次模型的数据结构是满足下列条件的树
  - 每个节点：一个现实世界的对象的数据记录(实体)
    - 每个实体可包含若干个实体属性
    - 实体属性：用字段描述
  - 边表示现对象之间的联系
    - 表示实体之间一对多联系





# 层次数据模型的数据结构

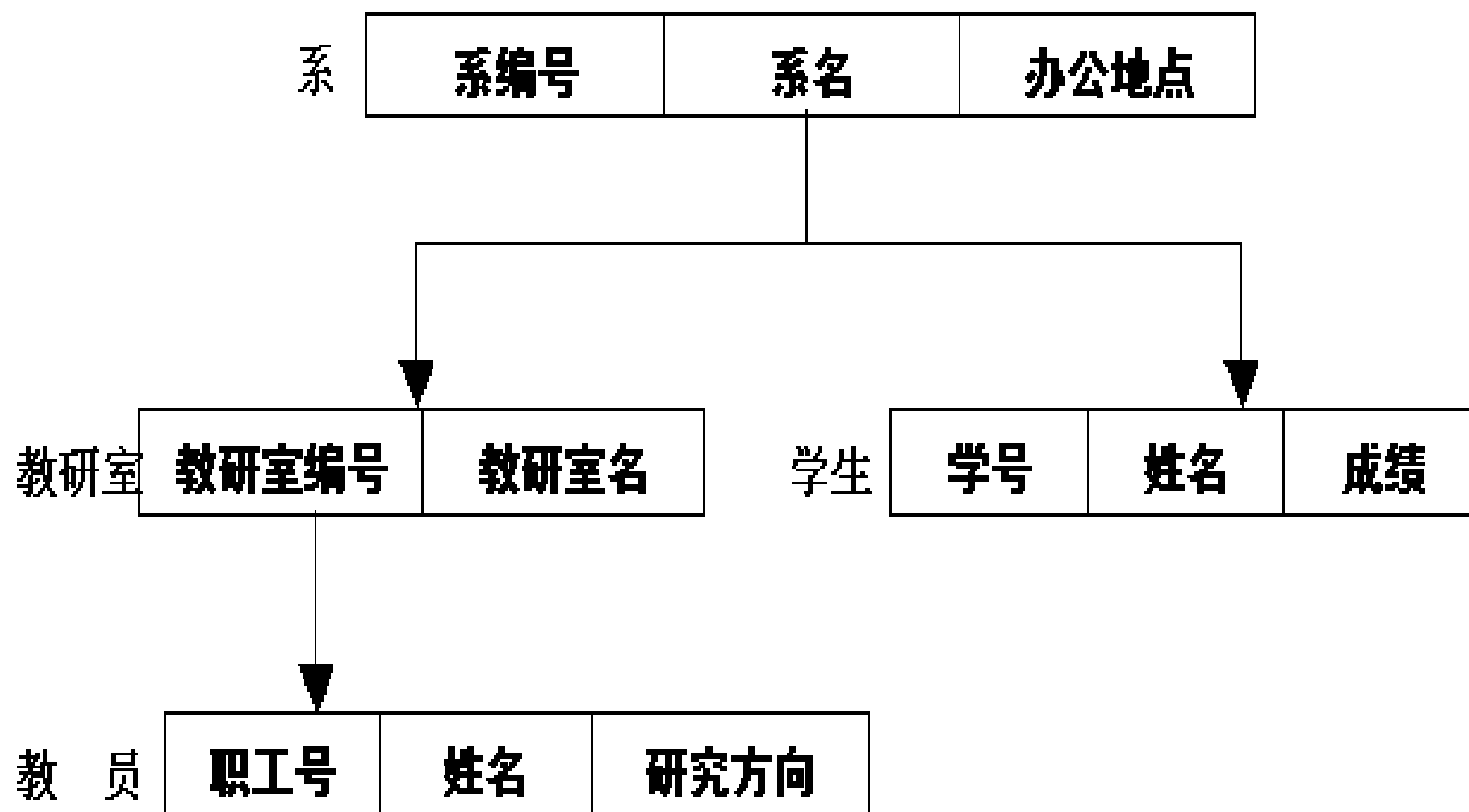
## • 特点

- 由于节点的父亲是唯一的，只能直接处理一对多的实体联系
- 任何记录值只有按其路径查看时，才能显出它的全部意义
- 没有一个子女记录值能够脱离父亲记录值而独立存在





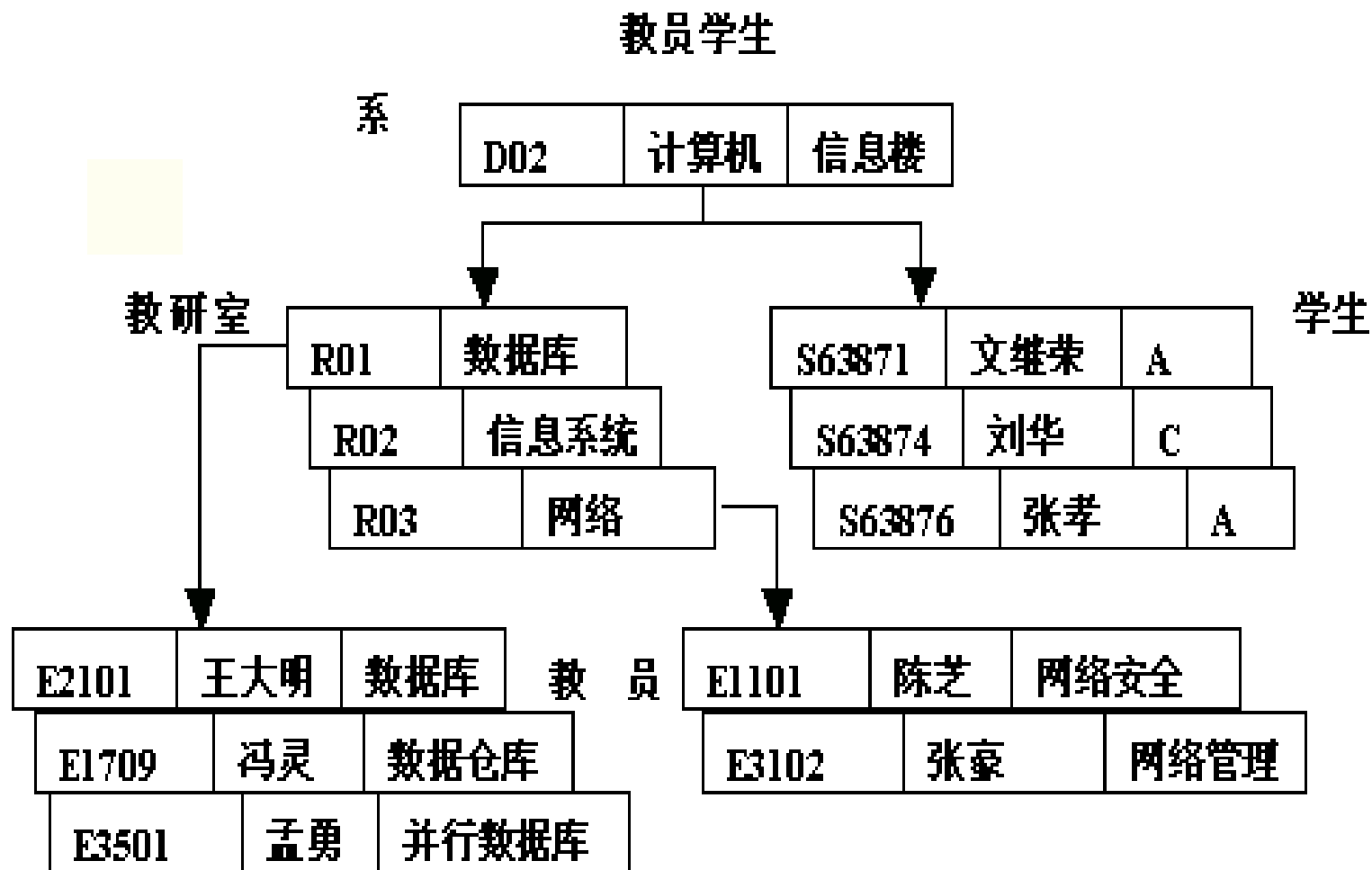
# “教员-教研室-学生”数据模型







# 教员学生数据库的实例





# 层次模型的数据操作

- 树定位
- 路径定义
- 沿路径的记录定位
- 记录查询
- 记录插入
- 记录删除
- 记录更新
- 模式更新

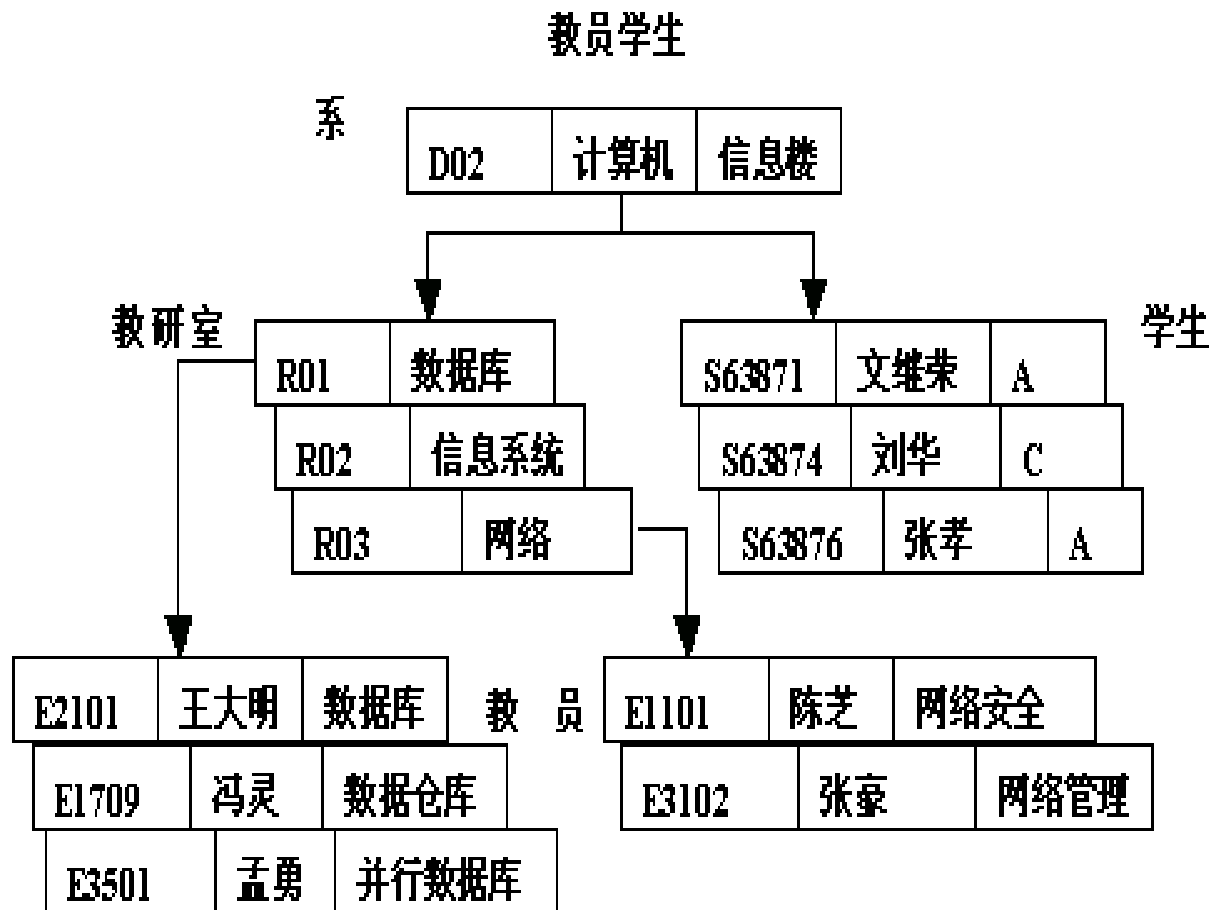


图 1.19 教员学生数据库的一个值



# 层次模型的完整性约束

- 无相应的父亲节点值就不能插入子女节点值
- 如果删除父亲节点值，则相应的子女节点值也被同时删除
- 更新数据时，应更新所有相应记录，以保证数据的一致性





# 层次模型的优缺点

- 优点

- 层次数据模型简单，对具有一对多的层次关系的现实世界描述自然、直观，容易理解
- 层次数据模型提供了良好的完整性支持

- 缺点

- 多对多联系表示不自然
- 对插入和删除操作的限制多
- 查询子女节点必须通过父亲节点
- 面向过程





# 具有代表性的层次数据库系统

- IMS数据库管理系统
  - 第一个大型商用DBMS
  - 1968年推出
  - IBM公司研制





# 网状数据模型

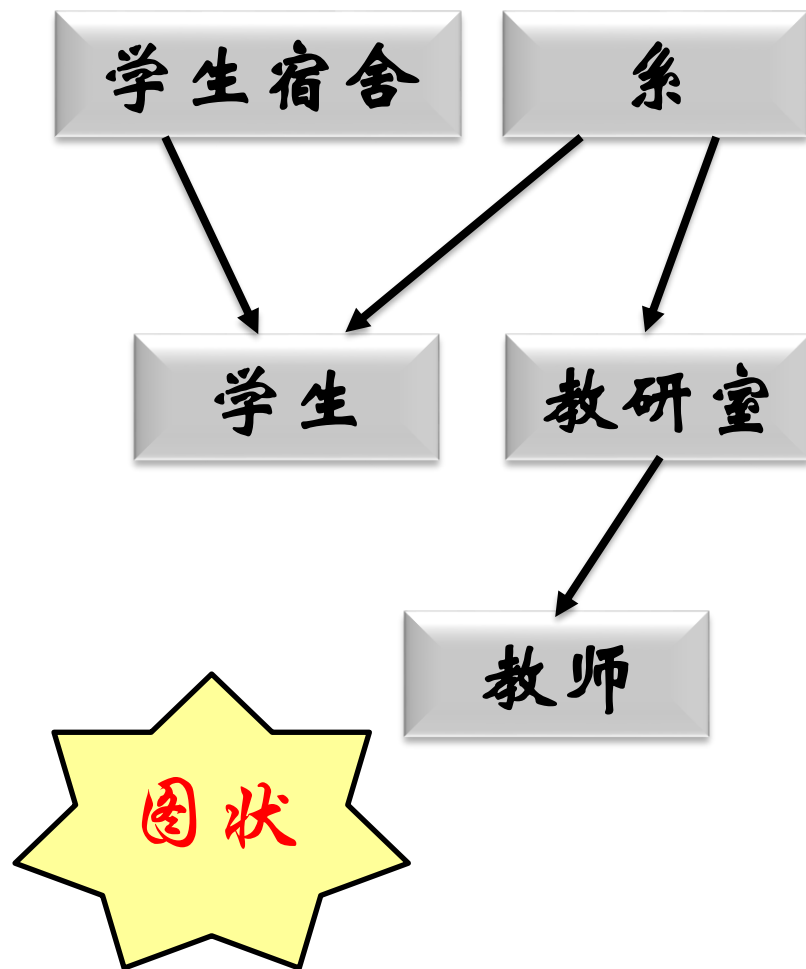
- 网状数据模型的数据结构
- 网状数据模型的数据操作
- 网状数据模型的完整性约束
- 网状数据模型的优缺点
- 具有代表性的网状数据库系统





# 网状数据模型的数据结构

- 网状模型的数据结构是满足下列条件的图
  - 每个节点是一个对象记录
    - 每个实体可包含若干个实体属性
    - 实体属性：用字段描述
  - 边表示对象之间的联系
    - 表示实体之间一对多联系
    - 容易间接表示实体之间的多对多联系
  - 允许多个节点无父亲节点
  - 允许节点有多个父亲节点
  - 允许两个节点之间有多种联系（复合联系）
- 层次模型是网状模型特例





# 网状数据模型的数据操作

- 网络定位
- 路径定位
- 沿路径的记录定位
- 记录查询
- 记录插入
- 记录删除
- 记录更新
- 模式更新







# 网状数据模型的完整性约束



- 完整性约束条件不严格
  - 允许插入尚未确定父亲节点值的子女节点值
  - 允许只删除父亲节点值
  - 一些具体系统提供了一些完整性约束





# 网状数据模型的优缺点

- 优点

- 能够更为直接地描述现实世界，如一个节点可以有多个父亲节点
- 具有良好的性能，查询、更新等不需要访问父亲节点

- 缺点

- 结构比较复杂，不利于最终用户掌握
- DDL、DML语言复杂，用户不容易使用
- 面向过程





# 典型的网状数据库系统

- 最早的网状数据库管理系统IDS(1964)
  - Charles W. Bachman(网状数据库之父)
  - 美国数据系统语言委员会CODASYL下属的数据库任务组DBTG于1971年推出了第一个正式报告——DBTG报告，成为数据库历史上具有里程碑意义的文献
    - 奠定了网状数据库系统的概念、方法和技术
    - 基于IDS的经验所确定的方法称为DBTG方法或CODASYL方法，所描述的网状模型称为DBTG模型或CODASYL模型





- 数据
- 数据库
- 数据库管理系统及其结构
- 数据库系统
- 数据抽象与数据模型
- 数据库系统的发展





# 数据库系统的发展

- 第一代数据库系统
  - 层次和网状数据库系统
- 第二代数据库系统
  - 关系数据库系统
    - 1970年, E.F.Codd提出关系数据模型和理论,  
获得ACM图灵奖
- 第三代数据库系统
  - 面向对象数据模型
    - 数据库技术与其他学科的技术内容互相结合





# 数据库系统的发展

- Internet时代的数据库技术

支持高层决策的数据仓库、OLAP分析

数据挖掘

数字图书馆

电子出版物

电子商务、Web医院、远程教育

基于Ad Hoc无线网的移动数据库

Web上的数据管理与信息检索

数据流管理

.....等。





# Homework

## 阅读教材的第一章





# 小结



- 数据
  - 描述现实世界各种具体事物或抽象概念的信息载体
  - 数据有不同的类型及语义
- 数据库
  - 存储在计算机系统 中的、互相关联的数据集合
  - 数据库模式、数据库实例
- 数据库管理系统
  - 管理数据库的软件系统
  - 支持数据存储、维护、查询、安全、正确
- 数据库系统
  - DB、DBMS、开发工具、应用系统、用户







# 总结



## • 本章重点

- 掌握数据库系统的基本概念，及与文件系统相比较的优势
- 掌握数据抽象中三级模式、两级映像、数据独立性等概念





**Now let's go to  
Next Chapter**

