

# 数据库系统概论

An Introduction to Database System

中国人民大学信息学院

School of Information,  
Renmin University of China

2014

# 教材及参考书 (1)

## 教材

- 王珊,萨师煊.数据库系统概论 (第5版) .北京:高等教育出版社, 2014.9
- **A First Course in Database Systems**  
**Jeffrey.D.Ullman, Jennifer Widom**  
**Dept. Of Computer Science Stanford University**



# 教材及参考书 (2)

## 参考书

- C.J. Date, An Introduction to Database System (Ed.8), Addison-Wesley,2003

## 上机软件

- KingBase ES

系统可以从北京人大金仓信息技术股份有限公司的网站<http://www.kingbase.com.cn/>上免费下载



# 学习方式

## 听课

(启发式、讨论式)

## 读书

(预习、复习)

## 报告

(综合练习)



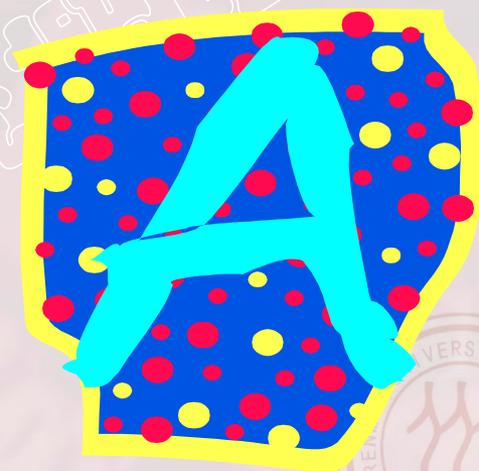
# 考试成绩

## ❖ 平时成绩

(书面作业、上机练习、综合练习)

## ❖ 期中考试

## ❖ 期末考试



# 内容安排 (1)

## 基础篇

- 第1章 绪论
- 第2章 关系数据库
- 第3章 关系数据库标准语言SQL
- 第4章 数据库安全性
- 第5章 数据库完整性

## 设计与应用开发篇

- 第6章 关系数据理论
- 第7章 数据库设计
- 第8章 数据库编程



# 内容安排 (2)

## 系统篇

- 第9章 关系查询处理和查询优化
- 第10章 数据库恢复技术
- 第11章 并发控制
- \* 第12章 数据库管理系统

第1至第11章是本科专业的基本教程  
(书中有\*号的部分除外)

第12至第16章是新技术篇 (本科生、研究生选读)



# 数据库系统概论

## 第一章 绪论



# 第一章 绪论

## 1.1 数据库系统概述

## 1.2 数据模型

## 1.3 数据库系统的结构

## 1.4 数据库系统的组成

## 1.5 小结



# 数据库的地位

- ❖ 数据库技术产生于六十年代末，是数据管理的有效技术，是计算机科学的重要分支。
- ❖ 数据库技术是信息系统的核心和基础，它的出现极大地促进了计算机应用向各行各业的渗透。
- ❖ 数据库已经成为每个人生活中不可缺少的部分。



# 1.1 数据库系统概述

## 1.1.1 数据库的4个基本概念

## 1.1.2 数据管理技术的产生和发展

## 1.1.3 数据库系统的特点



# 1.1.1 数据库的4个基本概念

- ❖ 数据 (**Data**)
- ❖ 数据库 (**Database**)
- ❖ 数据库管理系统 (**DBMS**)
- ❖ 数据库系统 (**DBS**)



# 1. 数据

❖ 数据（**Data**）是数据库中存储的基本对象

❖ 数据的定义

■ 描述事物的符号记录

❖ 数据的种类

■ 数字、文字、图形、图像、音频、视频、学生的档案记录等



# 数据举例

❖ 数据的含义称为数据的语义，数据与其语义是不可分的。

■ 例如 **93**是一个数据

语义1：学生某门课的成绩

语义2：某人的体重

语义3：计算机系**2013**级学生人数

语义4：请同学给出.....



# 数据举例

## ❖ 学生档案中的学生记录

(李明, 男, **199505**, 江苏南京市, 计算机系, **2013**)

■ 语义: 学生姓名、性别、出生年月、出生地、所在院系、入学时间

■ 解释: 李明是个大学生, **1995年5月**出生, 江苏南京市人, **2013年**考入计算机系



## 2. 数据库

### ❖ 数据库的定义

- 数据库（**Database**，简称**DB**）是长期储存在计算机内、有组织的、可共享的大量数据的集合。

### ❖ 数据库的基本特征

- 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



# 3. 数据库管理系统

## ❖ 什么是数据库管理系统

- 位于用户与操作系统之间的一层数据管理软件
- 是基础软件，是一个大型复杂的软件系统

## ❖ 数据库管理系统的用途

- 科学地组织和存储数据、高效地获取和维护数据



# 数据库在计算机系统的位置



# 数据库管理系统的主要功能

## ■ 数据定义功能

- 提供数据定义语言（**DDL**）
- 定义数据库中的数据对象

## ■ 数据组织、存储和管理

- 分类组织、存储和管理各种数据
- 确定组织数据的文件结构和存取方式
- 实现数据之间的联系
- 提供多种存取方法提高存取效率



# 数据库管理系统的主要功能

## ■ 数据操纵功能

- 提供数据操纵语言（DML）
- 实现对数据库的基本操作（查询、插入、删除和修改）

## ■ 数据库的事务管理和运行管理

- 数据库在建立、运行和维护时由数据库管理系统统一管理和控制
- 保证数据的安全性、完整性、多用户对数据的并发使用
- 发生故障后的系统恢复



# 数据库管理系统的主要功能

## ■ 数据库的建立和维护功能

- 数据库初始数据的装载和转换
- 数据库转储、恢复功能
- 数据库的重组
- 性能监视、分析等

## ■ 其它功能

- 数据库管理系统与网络中其它软件系统的通信
- 数据库管理系统系统之间的数据转换
- 异构数据库之间的互访和互操作



# 4.数据库系统

❖ 数据库系统（Database System，简称DBS）

❖ 数据库系统的构成

■ 数据库

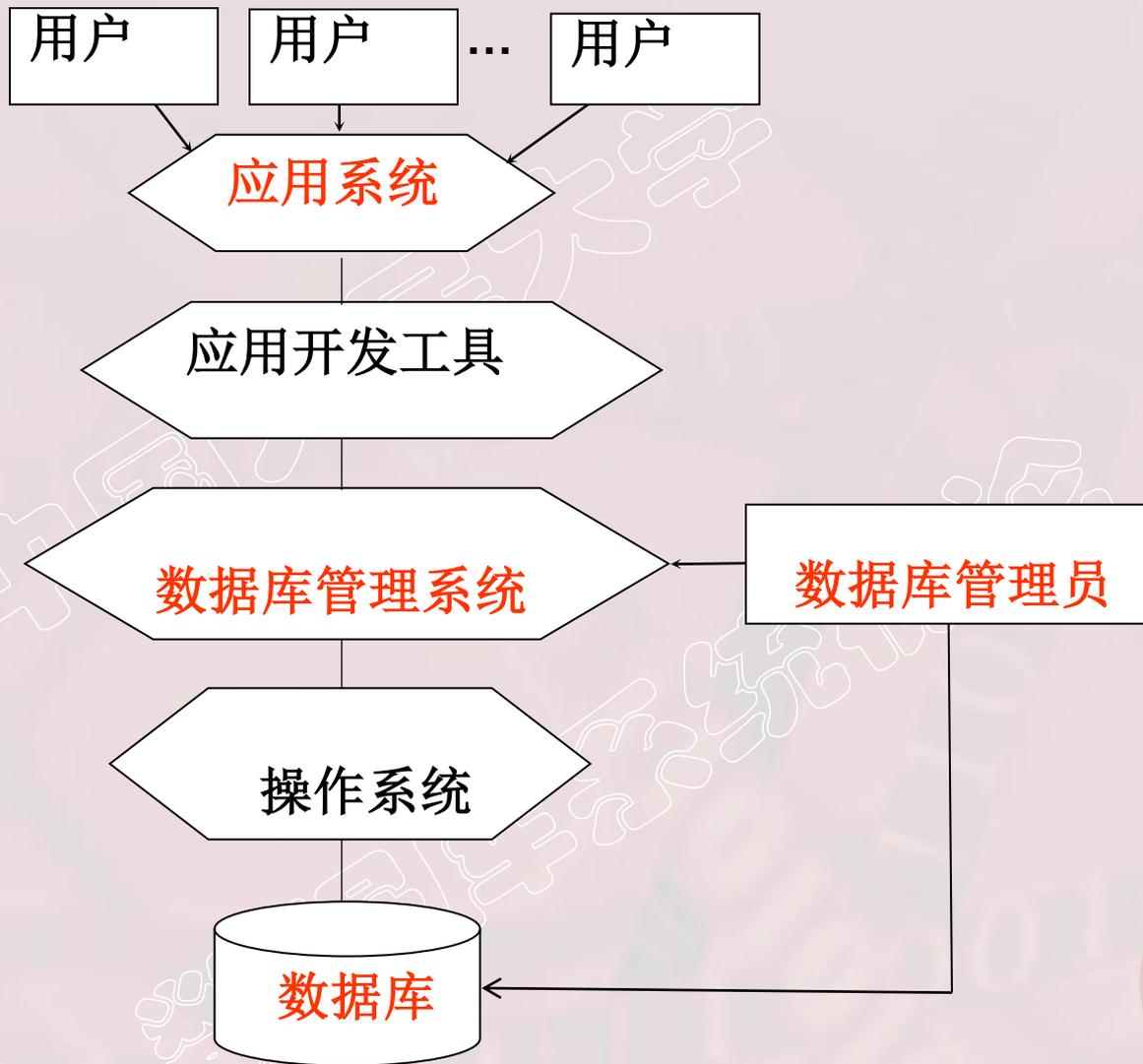
■ 数据库管理系统（及其应用开发工具）

■ 应用程序

■ 数据库管理员



# 数据库系统



# 1.1 数据库系统概述

1.1.1 四个基本概念

1.1.2 数据管理技术的产生和发展

1.1.3 数据库系统的特点



# 数据管理技术的产生和发展

## ❖ 什么是数据管理

- 对数据进行分类、组织、编码、存储、检索和维护
- 数据处理的中心问题

## ❖ 数据管理技术的发展过程

- 人工管理阶段（20世纪50年代中之前）
- 文件系统阶段（20世纪50年代末--60年代中）
- 数据库系统阶段（20世纪60年代末--现在）



# 数据管理技术的产生和发展（续）

## ❖ 数据管理技术的发展动力

- 应用需求的推动
- 计算机硬件的发展
- 计算机软件的发展



# 1. 人工管理阶段

## ❖ 时期

- 20世纪50年代中之前

## ❖ 产生的背景

- 应用背景      科学计算
- 硬件背景      无直接存取存储设备
- 软件背景      没有操作系统
- 处理方式      批处理



# 人工管理阶段（续）

## ❖ 特点

- 数据的管理者：用户（程序员），数据不保存
- 数据面向的对象：某一应用程序
- 数据的共享程度：无共享、冗余度极大
- 数据的独立性：不独立，完全依赖于程序
- 数据的结构化：无结构
- 数据控制能力：应用程序自己控制



# 应用程序与数据的对应关系（人工管理阶段）



人工管理阶段 应用程序与数据之间的对应关系



## 2. 文件系统阶段

### ❖ 时期

- 20世纪50年代末--60年代中

### ❖ 产生的背景

- 应用背景      科学计算、数据管理
- 硬件背景      磁盘、磁鼓
- 软件背景      有文件系统
- 处理方式      联机实时处理、批处理



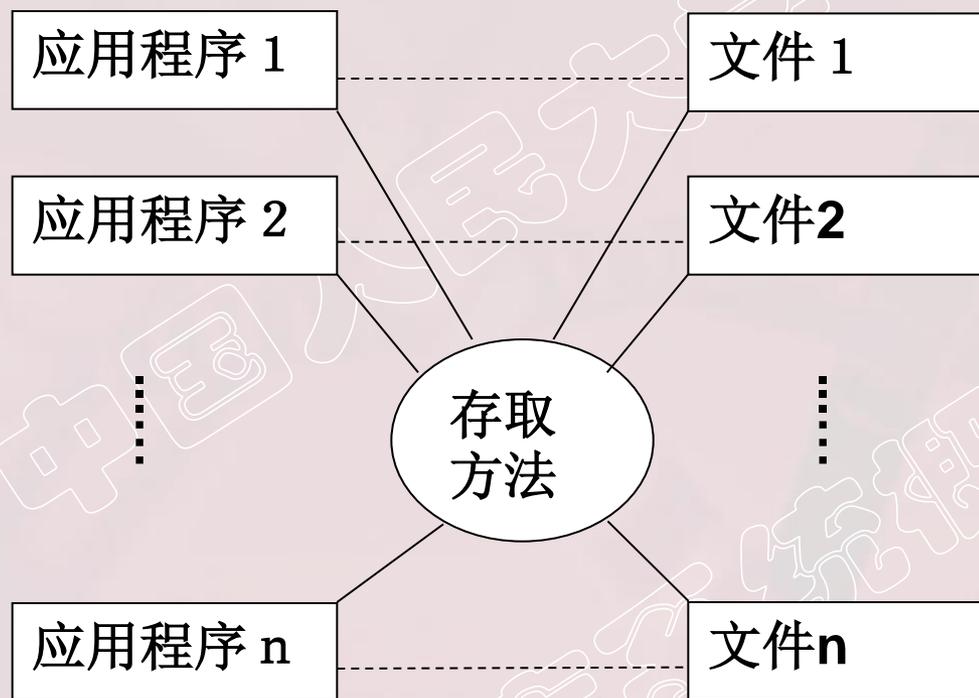
# 文件系统阶段（续）

## ❖ 特点

- 数据的管理者：文件系统，数据可长期保存
- 数据面向的对象：某一应用
- 数据的共享程度：共享性差、冗余度大
- 数据的结构化：记录内有结构，整体无结构
- 数据的独立性：独立性差
- 数据控制能力：应用程序自己控制



# 应用程序与数据的对应关系（文件系统阶段）



文件系统阶段 应用程序与数据之间的对应关系



# 3. 数据库系统阶段

## ❖ 时期

- 20世纪60年代末以来

## ❖ 产生的背景

- 应用背景      大规模数据管理
- 硬件背景      大容量磁盘、磁盘阵列
- 软件背景      有数据库管理系统
- 处理方式      联机实时处理,分布处理,批处理

## ❖ 文件系统到数据库系统，标志着数据管理技术的飞跃



# 1.1 数据库系统概述

1.1.1 四个基本概念

1.1.2 数据管理技术的产生和发展

1.1.3 数据库系统的特点



# 一个例子

❖ 学生的信息包括学号、姓名、性别、年龄、专业和奖励

## ■ 用文件系统实现学籍管理

### ● 数据存储

- 定长记录 存储在“学生基本信息”文件中
- 变长记录 存放在另一个“奖励”文件
- “学生基本信息”表中的位置和长度描述“奖励”文件中记录的开始位置和长度

### ● 查询数据

- 编写应用程序，实现数据的录入和查找

■ 缺点：程序员必须关注记录结构和不同文件中记录之间的联系，工作量大，编程复杂，开发速度慢



# 一个例子（续）

学号	姓名	性别	年龄	专业	位置	长度
20100001	史玉明	女	20	计算机	0	30
20100100	李明虎	男	21	机械	30	15
20100234	张翔	男	21	化工	45	0
.....	.....	.....	.....	.....	.....	.....

“学生基本信息”文件的结构和内容

奖励
2011校奖学金，2012国家奖学金
2012校优秀学生

“奖励”文件的结构和内容



# 一个例子（续）

## ■ 数据库系统管理

### ● 存储数据

#### ➢ 建立两张表:

**STUDENT**表-存放学生的基本信息,

**AWARD**表-存放学生的奖励情况

#### ➢ 使用两条插入命令 完成学生基本信息和奖励情况的数据录入功能

### ● 查询功能

可以用一条查询语句实现



# 1.1.3 数据库系统的特点

- ❖ 数据结构化
- ❖ 数据的共享性高，冗余度低且易扩充
- ❖ 数据独立性高
- ❖ 数据由数据库管理系统统一管理和控制



# 数据结构化

❖ 数据的整体结构化是数据库的主要特征之一

❖ 整体结构化

- 不再仅仅针对某一个应用，而是面向全组织
- 不仅数据内部结构化，整体是结构化的，数据之间具有联系
- 数据记录可以变长
- 数据的最小存取单位是数据项

数据的用数据模型描述，无需应用程序定义



# 数据的共享性高，冗余度低且易扩充

- ❖ 数据面向整个系统，可以被多个用户、多个应用共享使用。
- ❖ 数据共享的好处
  - 减少数据冗余，节约存储空间
  - 避免数据之间的不相容性与不一致性
  - 使系统易于扩充



# 数据独立性高

## ❖ 物理独立性

- 指用户的应用程序与数据库中数据的物理存储是相互独立的。当数据的物理存储改变了，应用程序不用改变。

## ❖ 逻辑独立性

- 指用户的应用程序与数据库的逻辑结构是相互独立的。数据的逻辑结构改变了，应用程序不用改变。

## ❖ 数据独立性由数据库管理系统的二级映像功能来保证。



# 数据由数据管理系统统一管理和控制

## ❖ 数据库管理系统提供的数据库控制功能

### (1) 数据的安全性 (**Security**) 保护

保护数据以防止不合法的使用造成的数据的泄密和破坏。

### (2) 数据的完整性 (**Integrity**) 检查

保证数据的正确性、有效性和相容性。

### (3) 并发 (**Concurrency**) 控制

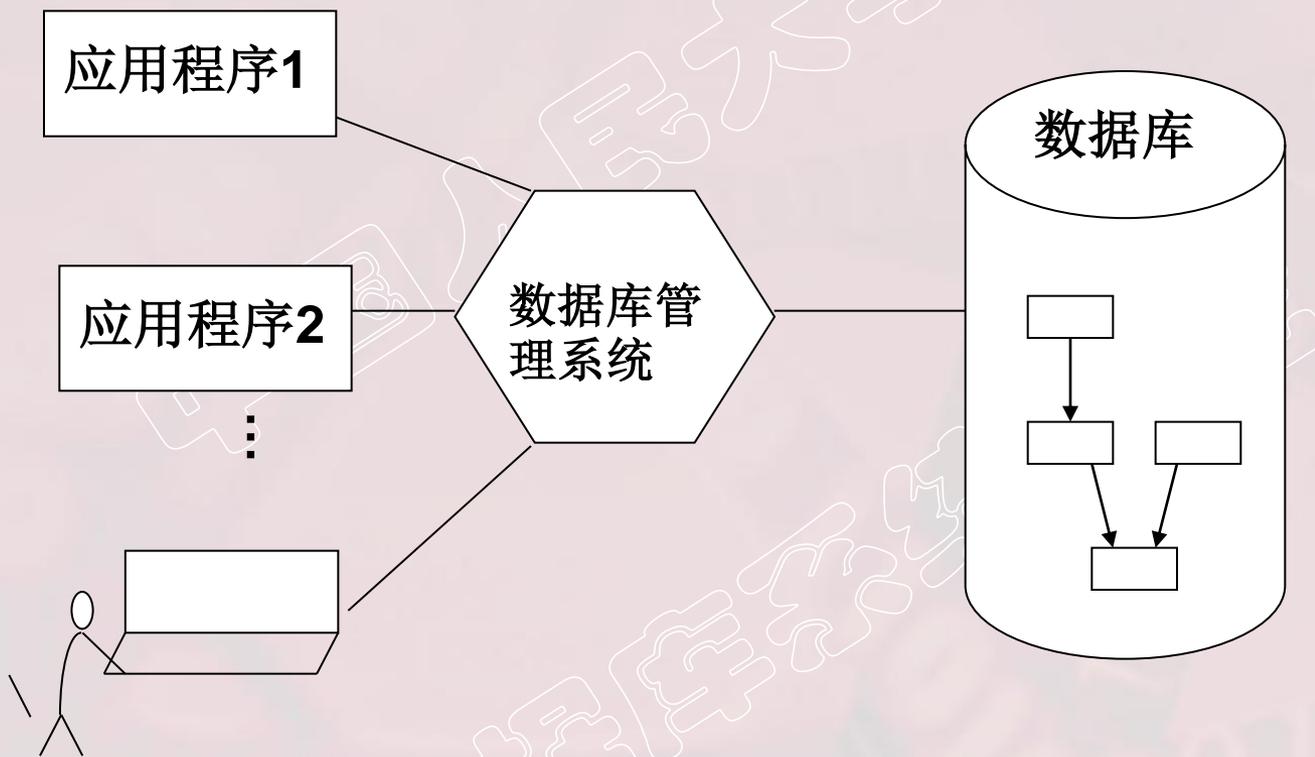
对多用户的并发操作加以控制和协调，防止相互干扰而得到错误的结果。

### (4) 数据库恢复 (**Recovery**)

将数据库从错误状态恢复到某一已知的正确状态。



# 应用程序与数据的对应关系（数据库系统阶段）



数据库系统阶段 应用程序与数据之间的对应关系



# 数据库概念小结

- ❖ 数据库是长期存储在计算机内有组织的大量的共享的数据集合。
- ❖ 可以供各种用户共享，具有最小冗余度和较高的数据独立性。
- ❖ 数据库管理系统在数据库建立、运用和维护时对数据库进行统一控制，以保证数据的完整性、安全性，并在多用户同时使用数据库时进行并发控制，在发生故障后对数据库进行恢复。



# 第一章 绪论

1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统的结构

1.4 数据库系统的组成

1.5 小结



## 1.2 数据模型

- ❖ 数据模型是对现实世界数据特征的抽象。
- ❖ 通俗地讲数据模型就是现实世界的模拟。
- ❖ 数据模型应满足三方面要求
  - 能比较真实地模拟现实世界
  - 容易为人所理解
  - 便于在计算机上实现
- ❖ 数据模型是数据库系统的核心和基础



# 1.2 数据模型

## 1.2.1 两类数据模型

## 1.2.2 概念模型

## 1.2.3 数据模型的组成要素

## 1.2.4 常用的数据模型

## 1.2.5 层次模型

## 1.2.6 网状模型

## 1.2.7 关系模型



# 1.2.1 两类数据模型

❖ 数据模型分为两类（两个不同的层次）

(1) **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，用于数据库设计。

(2) **逻辑模型和物理模型**

■ 逻辑模型主要包括网状模型、层次模型、关系模型、面向对象数据模型、对象关系数据模型、半结构化数据模型等。按计算机系统的观点对数据建模，用于**DBMS**实现。

■ 物理模型是对数据最底层的抽象，描述数据在系统内部的表示方式和存取方法，在磁盘或磁带上的存储方式和存取方法。



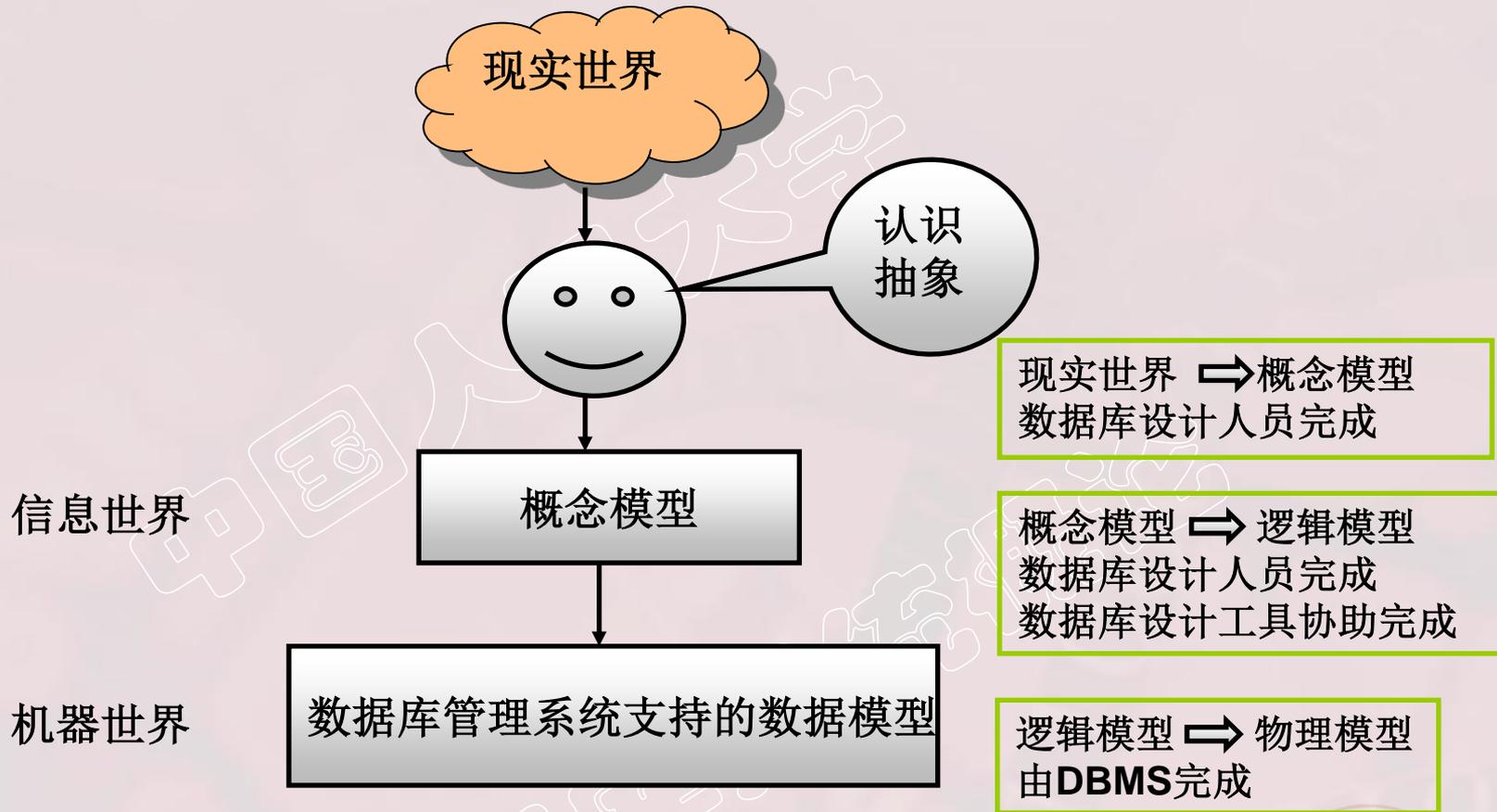
# 两类数据模型（续）

## ❖ 客观对象的抽象过程---两步抽象

- 现实世界中的客观对象抽象为概念模型；
  - 将现实世界抽象为信息世界
- 把概念模型转换为某一数据库管理系统支持的数据模型。
  - 将信息世界转换为机器世界



# 两类数据模型（续）



现实世界中客观对象的抽象过程



# 1.2 数据模型

1.2.1 两大类数据模型

1.2.2 概念模型

1.2.3 数据模型的组成要素

1.2.4 最常用的数据模型

1.2.5 层次模型

1.2.6 网状模型

1.2.7 关系模型



## 1.2.2 概念模型

### ❖ 概念模型的用途

- 概念模型用于信息世界的建模
- 是现实世界到机器世界的一个中间层次
- 是数据库设计的有力工具
- 数据库设计人员和用户之间进行交流的语言

### ❖ 对概念模型的基本要求

- 较强的语义表达能力
- 简单、清晰、易于用户理解



## 1.2.2 概念模型

1. 信息世界中的基本概念
2. 概念模型的一种表示方法：实体-联系方法



# 1. 信息世界中的基本概念

## (1) 实体 (Entity)

客观存在并可相互区别的事物称为实体。

可以是具体的人、事、物或抽象的概念。

## (2) 属性 (Attribute)

实体所具有的某一特性称为属性。

一个实体可以由若干个属性来刻画。

## (3) 码 (Key)

唯一标识实体的属性集称为码。



# 信息世界中的基本概念（续）

## （4）实体型（Entity Type）

用实体名及其属性名集合来抽象和刻画同类实体称为  
实体型

## （5）实体集（Entity Set）

同一类型实体的集合称为实体集



# 信息世界中的基本概念（续）

## （6）联系（Relationship）

- 现实世界中事物内部以及事物之间的联系在信息世界中反映为实体（型）内部的联系和实体（型）之间的联系。
- 实体内部的联系通常是指组成实体的各属性之间的联系
- 实体之间的联系通常是指不同实体集之间的联系
- 实体之间的联系有一对一、一对多和多对多等多种类型



# 实体-联系方法

## ❖ 实体-联系方法（Entity-Relationship Approach）

- 用E-R图来描述现实世界的概念模型
- E-R方法也称为E-R模型



# 1.2 数据模型

1.2.1 两大类数据模型

1.2.2 概念模型

1.2.3 数据模型的组成要素

1.2.4 最常用的数据模型

1.2.5 层次模型

1.2.6 网状模型

1.2.7 关系模型



## 1.2.3 数据模型的组成要素

- ❖ 数据结构
- ❖ 数据操作
- ❖ 数据的完整性约束条件



# 1. 数据结构

## ❖ 数据模型的数据结构

- 描述数据库的组成对象，以及对象之间的联系

## ❖ 描述的内容

1. 与对象的类型、内容、性质有关
2. 与数据之间联系有关

## ❖ 数据结构是对系统静态特性的描述



## 2. 数据操作

### ❖ 数据操作

- 对数据库中各种对象（型）的实例（值）允许执行的操作的集合，包括操作及有关的操作规则

### ❖ 数据操作的类型

- 查询
- 更新（包括插入、删除、修改）



# 数据操作（续）

## ❖ 数据模型对操作的定义

- 操作的确切含义
- 操作符号
- 操作规则（如优先级）
- 实现操作的语言

## ❖ 数据操作是对系统动态特性的描述



# 3. 数据的完整性约束条件

## ❖ 数据的完整性约束条件

- 一组完整性规则的集合
- 完整性规则：给定的数据模型中数据及其联系所具有的制约和依存规则
- 用以限定符合数据模型的数据库状态以及状态的变化，以保证数据的正确、有效和相容



# 数据的完整性约束条件（续）

## ❖ 数据模型对完整性约束条件的定义

- 反映和规定必须遵守的**基本的通用的**完整性约束条件。
- 提供定义完整性约束条件的机制，以反映**具体应用所**涉及的数据必须遵守的特定的语义约束条件。



# 1.2 数据模型

1.2.1 两大类数据模型

1.2.2 数据模型的组成要素

1.2.3 概念模型

1.2.4 常用的数据模型

1.2.5 层次模型

1.2.6 网状模型

1.2.7 关系模型



## 1.2.4 常用的数据模型

- ❖ 层次模型 (Hierarchical Model)
- ❖ 网状模型 (Network Model)
- ❖ 关系模型 (Relational Model)
- ❖ 面向对象数据模型 (Object Oriented Data Model)
- ❖ 对象关系数据模型 (Object Relational Data Model)
- ❖ 半结构化数据模型 (Semistructure Data Model)



# 1.2 数据模型

1.2.1 两大类数据模型

1.2.2 数据模型的组成要素

1.2.3 概念模型

1.2.4 常用的数据模型

1.2.5 层次模型

1.2.6 网状模型

1.2.7 关系模型



## 1.2.5 层次模型

- ❖ 层次模型是数据库系统中最早出现的数据模型
- ❖ 层次数据库系统的典型代表是IBM公司的IMS  
(Information Management System) 数据库  
管理系统
- ❖ 层次模型用树形结构来表示各类实体以及实体间的联系



# 1. 层次模型的数据结构

## ❖ 层次模型

满足下面两个条件的基本层次联系的集合为层次模型

1. 有且只有一个结点没有双亲结点，这个结点称为根结点
2. 根以外的其它结点有且只有一个双亲结点

## ❖ 层次模型中的几个术语

- 根结点，双亲结点，兄弟结点，叶结点



# 层次模型的数据结构（续）

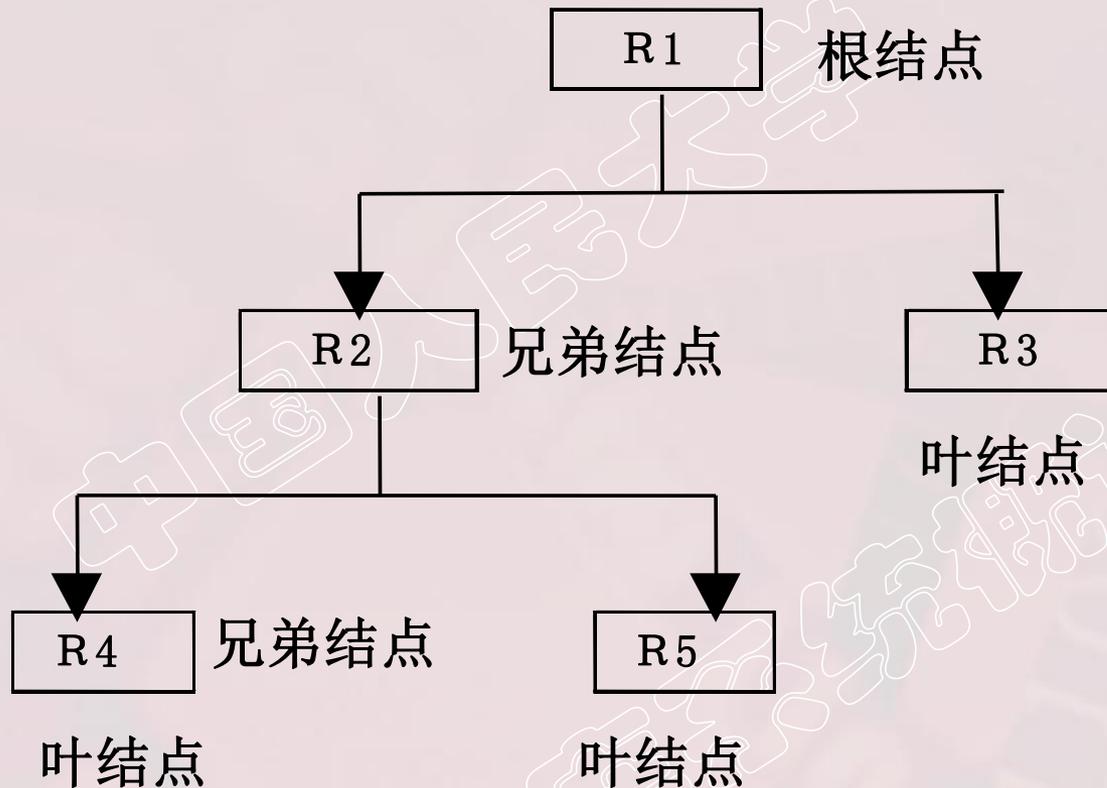


图1.9 一个层次模型的示例



# 层次模型的数据结构（续）

## ❖ 层次模型的特点：

- 结点的双亲是唯一的
- 只能直接处理一对多的实体联系
- 每个记录类型可以定义一个排序字段，也称为码字段
- 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双亲记录值而独立存在



# 层次模型的数据结构（续）

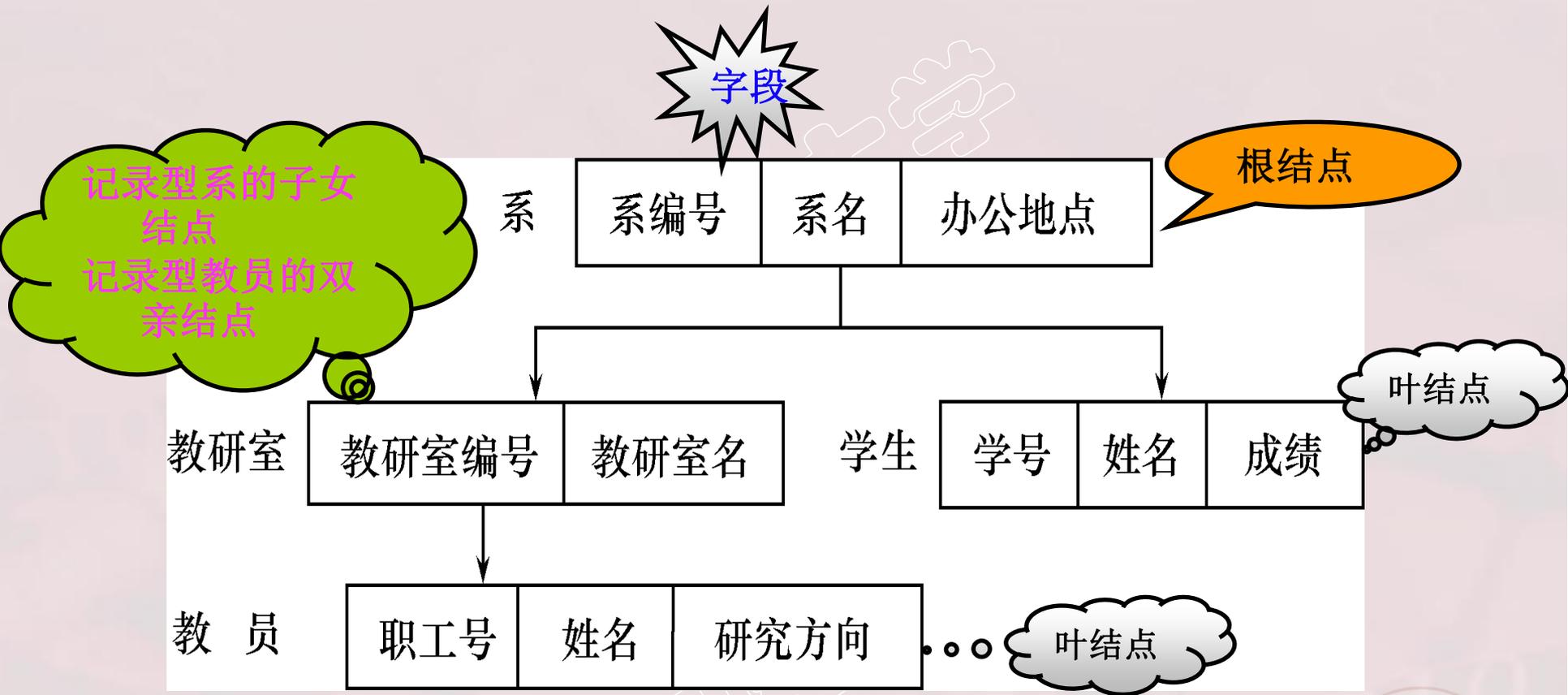


图1.10 教员学生层次数据库模型



# 层次模型的数据结构（续）

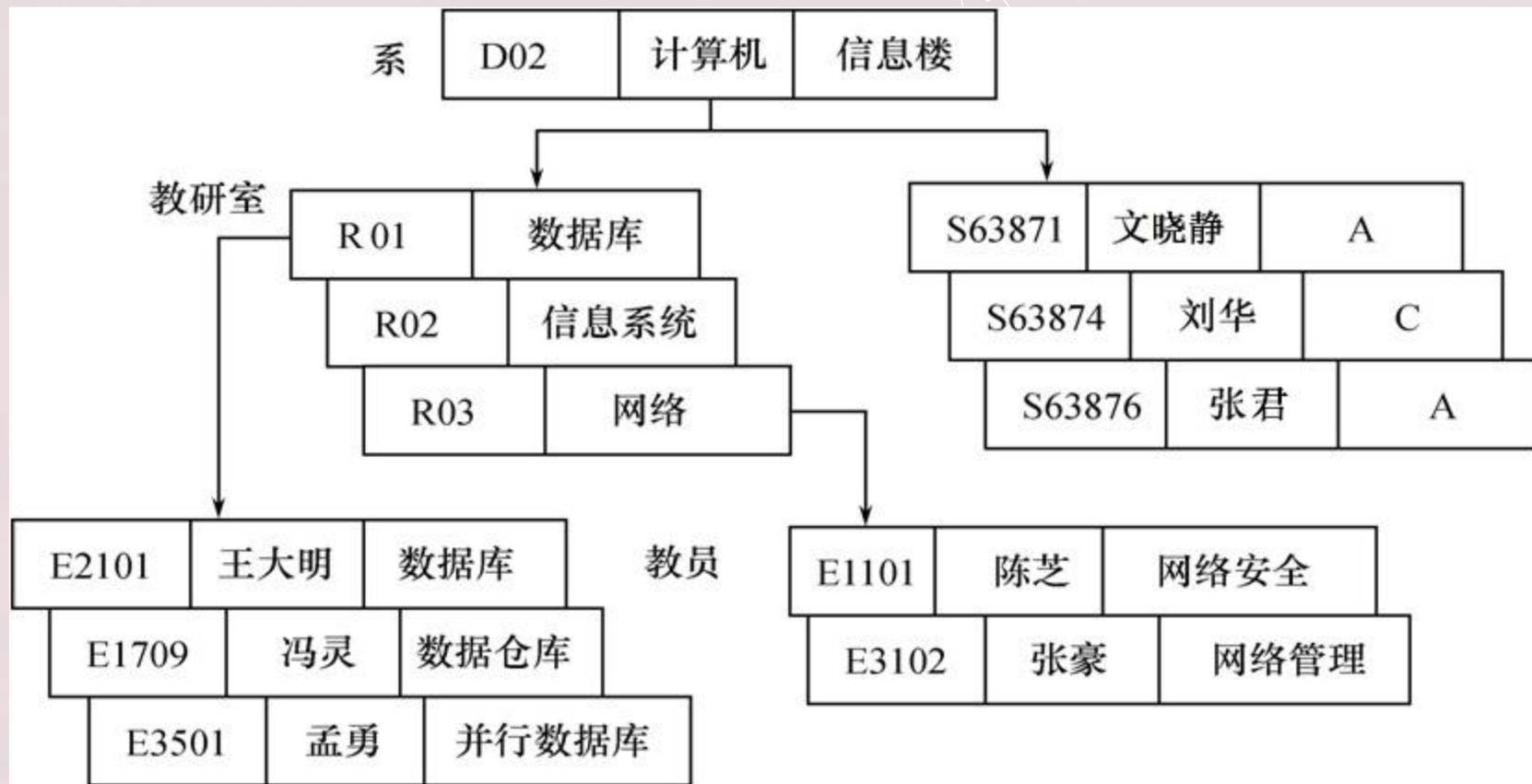


图1.11 教员学生层次数据库的一个值

## 2. 层次模型的数据操纵与完整性约束

### ❖ 层次模型的数据操纵

- 查询
- 插入
- 删除
- 更新



# 层次模型的完整性约束条件（续）

## ❖ 层次模型的完整性约束条件

- 无相应的双亲结点值就不能插入子女结点值
- 如果删除双亲结点值，则相应的子女结点值也被同时删除
- 更新操作时，应更新所有相应记录，以保证数据的一致性



# 3.层次模型的优缺点

## ❖ 优点

- 层次模型的数据结构比较简单清晰
- 查询效率高，性能优于关系模型，不低于网状模型
- 层次数据模型提供了良好的完整性支持

## ❖ 缺点

- 结点之间的多对多联系表示不自然
- 对插入和删除操作的限制多，应用程序的编写比较复杂
- 查询子女结点必须通过双亲结点
- 层次命令趋于程序化



# 1.2 数据模型

1.2.1 两大类数据模型

1.2.2 数据模型的组成要素

1.2.3 概念模型

1.2.4 最常用的数据模型

1.2.5 层次模型

1.2.6 网状模型

1.2.7 关系模型



## 1.2.6 网状模型

- ❖ 网状数据库系统采用**网状模型**作为数据的组织方式
- ❖ 典型代表是**DBTG**系统：
  - 亦称**CODASYL**系统
  - 20世纪70年代由**DBTG**提出的一个系统方案
- ❖ 实际系统
  - **Cullinet Software**公司的 **IDMS**
  - **Univac**公司的 **DMS1100**
  - **Honeywell**公司的**IDS/2**
  - **HP**公司的**IMAGE**



# 1. 网状模型的数据结构

## ❖ 网状模型

满足下面两个条件的基本层次联系的集合：

1. 允许一个以上的结点无双亲；
2. 一个结点可以有多个的双亲。



# 网状模型的数据结构（续）

## ❖ 表示方法（与层次数据模型相同）

**实体型：**用记录类型描述

每个结点表示一个记录类型（实体）

**属性：**用字段描述

每个记录类型可包含若干个字段

**联系：**用结点之间的连线表示记录类型（实体）之间的一对多的父子联系



# 网状模型的数据结构（续）

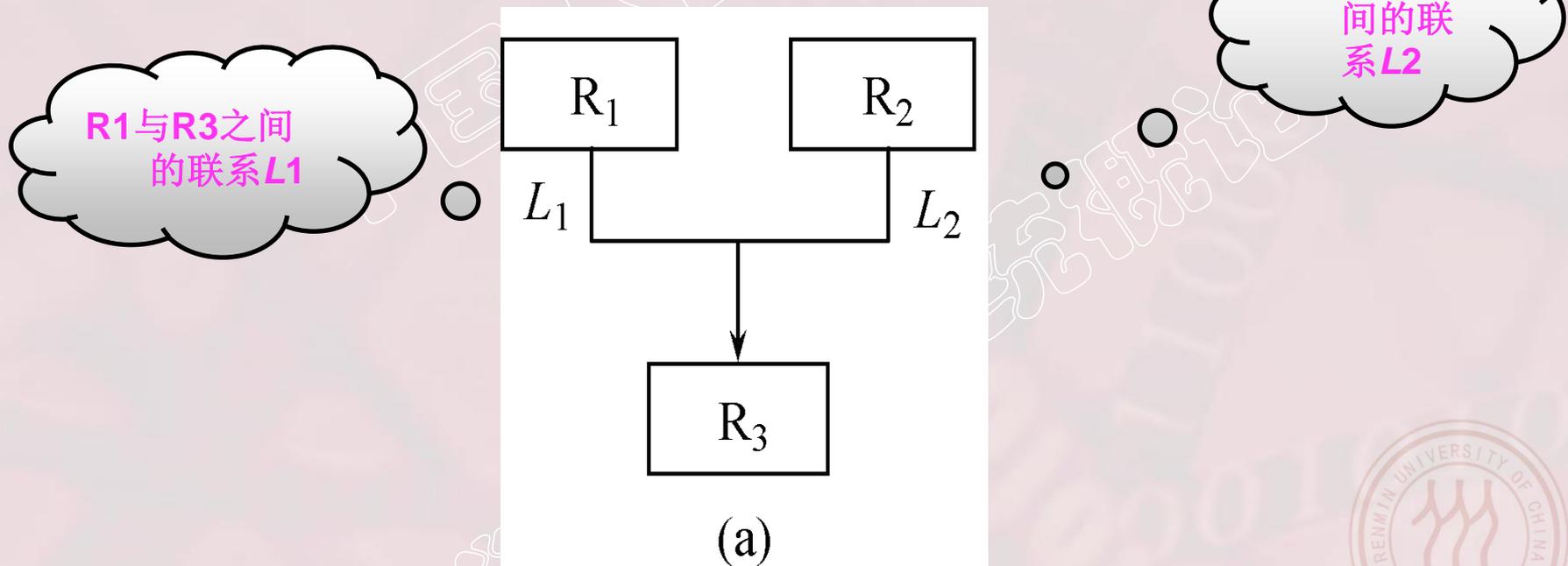
## ❖ 网状模型与层次模型的区别

- 网状模型允许多个结点没有双亲结点
- 网状模型允许结点有多个双亲结点
- 网状模型允许两个结点之间有多种联系（复合联系）
- 网状模型可以更直接地描述现实世界
- 层次模型实际上是网状模型的一个特例



# 网状模型的数据结构（续）

- ❖ 网状模型中子女结点与双亲结点的联系可以不唯一，要为每个联系命名，并指出与该联系有关的双亲记录和子女记录



网状模型的例子



# 网状模型的数据结构（续）

## 多对多联系在网状模型中的表示

- 用网状模型**间接**表示多对多联系

- 方法：

将多对多联系**直接**分解成一对多联系



# 网状模型的数据结构（续）

例如：一个学生可以选修若干门课程，某一课程可以被多个学生选修，学生与课程之间是多对多联系

- 引进一个学生选课的联结记录，由**3**个数据项组成
  - 学号
  - 课程号
  - 成绩
  - 表示某个学生选修某一门课程及其成绩



# 网状模型的数据结构（续）

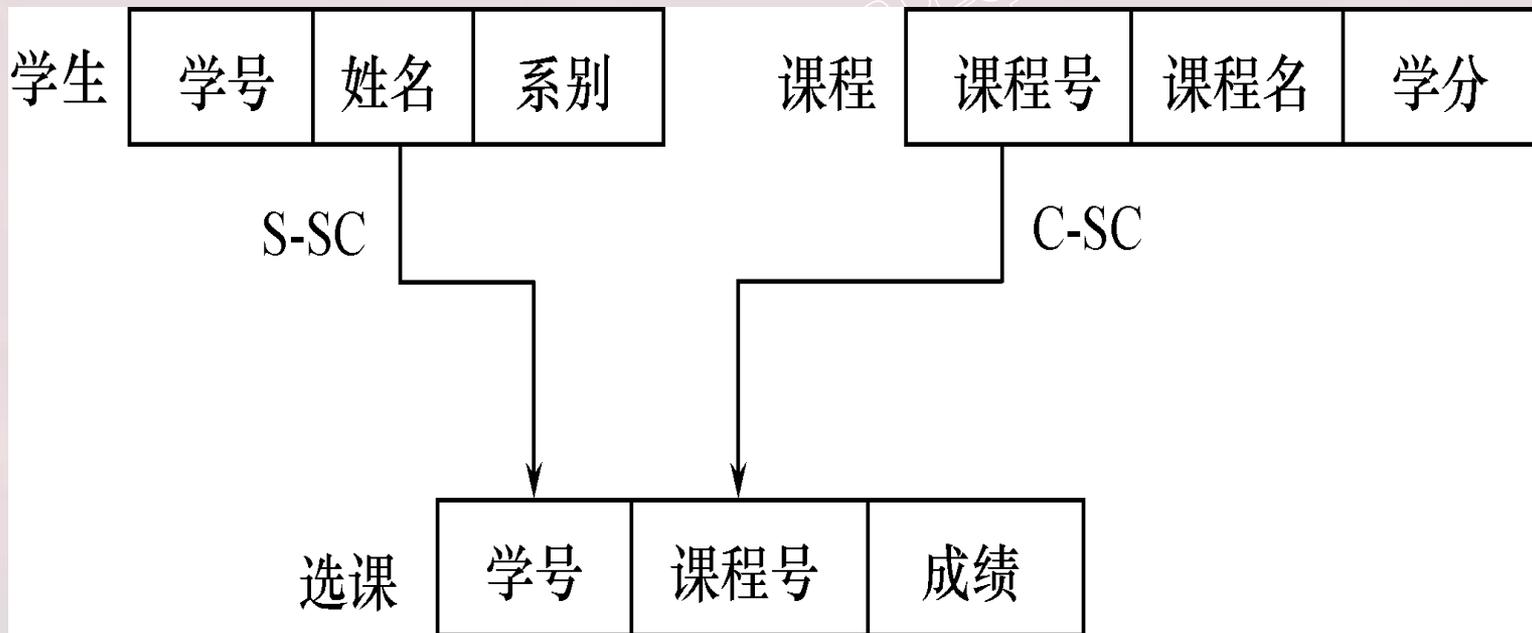


图1.13 学生/选课/课程的网状数据模型



## 2. 网状模型的操纵与完整性约束

❖ 网状数据库系统（如**DBTG**）对数据操纵加了一些限制，提供了一定的完整性约束

- 码：唯一标识记录的数据项的集合
- 一个联系中双亲记录与子女记录之间是一对多联系
- 支持双亲记录和子女记录之间某些约束条件



# 3. 网状模型的优缺点

## ❖ 优点

- 能够更为直接地描述现实世界，如一个结点可以有多个双亲
- 具有良好的性能，存取效率较高

## ❖ 缺点

- 结构比较复杂，而且随着应用环境的扩大，数据库的结构就变得越来越复杂，不利于最终用户掌握
- DDL、DML语言复杂，用户不容易使用
- 记录之间联系是通过存取路径实现的，用户必须了解系统结构的细节



# 1.2 数据模型

1.2.1 两大类数据模型

1.2.2 数据模型的组成要素

1.2.3 概念模型

1.2.4 最常用的数据模型

1.2.5 层次模型

1.2.6 网状模型

1.2.7 关系模型



## 1.2.7 关系模型

- ❖ 关系数据库系统采用关系模型作为数据的组织方式
- ❖ 1970年美国IBM公司San Jose研究室的研究员  
E.F.Codd首次提出了数据库系统的关系模型
- ❖ 计算机厂商新推出的数据库管理系统几乎都支持关系模型



# 1. 关系模型的数据结构

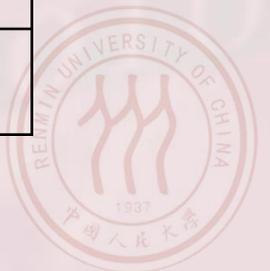
- ❖ 在用户观点下，关系模型中数据的逻辑结构是一张二维表，它由行和列组成。

学生登记表

属性

元组

学号	姓名	年龄	性别	系名	年级
2013004	王小明	19	女	社会学	2013
2013006	黄大鹏	20	男	商品学	2013
2013008	张文斌	18	女	法律	2013
...	...	...	...	...	...



# 关系模型的数据结构（续）

## ■ 关系（Relation）

- 一个关系对应通常说的一张表

## ■ 元组（Tuple）

- 表中的一行即为一个元组

## ■ 属性（Attribute）

- 表中的一列即为一个属性，给每一个属性起一个名称即属性名

## ■ 主码（Key）

- 也称码键。表中的某个属性组，它可以唯一确定一个元组



# 关系模型的数据结构（续）

## ■ 域（Domain）

- 是一组具有相同数据类型的值的集合。属性的取值范围来自某个域。

## ■ 分量

- 元组中的一个属性值。

## ■ 关系模式

- 对关系的描述

关系名（属性1，属性2，...，属性n）

学生（学号，姓名，年龄，性别，系名，年级）



# 关系模型的数据结构（续）

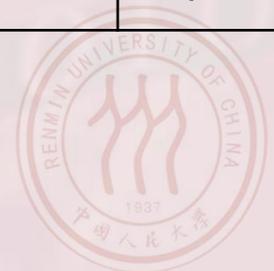
❖ 关系必须是规范化的，满足一定的规范条件

最基本的规范条件：关系的每一个分量必须是一个不可分的数据项，不允许表中还有表

图1.15中工资和扣除是可分的数据项，不符合关系模型要求

职工号	姓名	职称	工资			扣除		实发
			基本工资	岗位津贴	业绩津贴	三险	个人所得税	
86051	陈平	讲师	1305	1200	1850	160	112	4083
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

图1.15 一个工资表（表中有表）实例



# 关系模型的数据结构（续）

表1.5 术语对比

关系术语	一般表格的术语
关系名	表名
关系模式	表头（表格的描述）
关系	（一张）二维表
元组	记录或行
属性	列
属性名	列名
属性值	列值
分量	一条记录中的一个列值
非规范关系	表中有表（大表中嵌有小表）

## 2. 关系模型的操纵与完整性约束

❖ 数据操作是集合操作，操作对象和操作结果都是关系

■ 查询

■ 插入

■ 删除

■ 更新

❖ 存取路径对用户隐蔽，用户只要指出“干什么”，不必详细说明“怎么干”



# 关系模型的操纵与完整性约束（续）

## ❖ 关系的完整性约束条件

- 实体完整性
- 参照完整性
- 用户定义的完整性



# 3. 关系模型的优缺点

## ❖ 优点

- 建立在严格的数学概念的基础上
- 概念单一
  - 实体和各类联系都用关系来表示
  - 对数据的检索结果也是关系
- 关系模型的存取路径对用户透明
  - 具有更高的数据独立性，更好的安全保密性
  - 简化了程序员的工作和数据库开发建立的工作



# 关系模型的优缺点（续）

## ❖ 缺点

- 存取路径对用户透明，查询效率往往不如格式化数据模型
- 为提高性能，必须对用户的查询请求进行优化，增加了开发数据库管理系统的难度



# 第一章 绪论

1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统的结构

1.4 数据库系统的组成

1.5 小结



# 1.3 数据库系统的结构

❖ 从数据库应用开发人员角度看，数据库系统通常采用三级模式结构，是数据库系统内部的系统结构

从数据库最终用户角度看，数据库系统的结构分为：

- 单用户结构
- 主从式结构
- 分布式结构
- 客户-服务器
- 浏览器-应用服务器 / 数据库服务器多层结构等



# 数据库系统的结构（续）

## 1.3.1 数据库系统模式的概念

## 1.3.2 数据库系统的三级模式结构

## 1.3.3 数据库的二级映像功能与数据独立性



# 1.3.1 数据库系统模式的概念

## ❖ “型” 和 “值” 的概念

### ■ 型 (Type)

- 对某一类数据的结构和属性的说明

### ■ 值 (Value)

- 是型的一个具体赋值

例如

学生记录:

(学号, 姓名, 性别, 系别, 年龄, 籍贯)

一个记录值:

(**201315130**, 李明, 男, 计算机系, **19**, 江苏南京市)



# 数据库系统模式的概念（续）

## ❖ 模式（Schema）

- 数据库逻辑结构和特征的描述
- 是型的描述，不涉及具体值
- 反映的是数据的结构及其联系
- 模式是相对稳定的

## ❖ 实例（Instance）

- 模式的一个具体值
- 反映数据库某一时刻的状态
- 同一个模式可以有很多实例
- 实例随数据库中的数据更新而变动



# 数据库系统模式的概念（续）

例如：在学生选课数据库模式中，包含学生记录、课程记录和学生选课记录

■ 2013年的一个学生数据库实例，包含：

- 2013年学校中所有学生的记录
- 学校开设的所有课程的记录
- 所有学生选课的记录

■ 2012年度学生数据库模式对应的实例与

2013年度学生数据库模式对应的实例是**不同的**



# 数据库系统结构（续）

**1.3.1 数据库系统模式的概念**

**1.3.2 数据库系统的三级模式结构**

**1.3.3 数据库的二级映像功能与数据独立性**



## 1.3.2 数据库系统的三级模式结构

- ❖ 模式 (Schema)
- ❖ 外模式 (External Schema)
- ❖ 内模式 (Internal Schema)



# 数据库系统的三级模式结构（续）

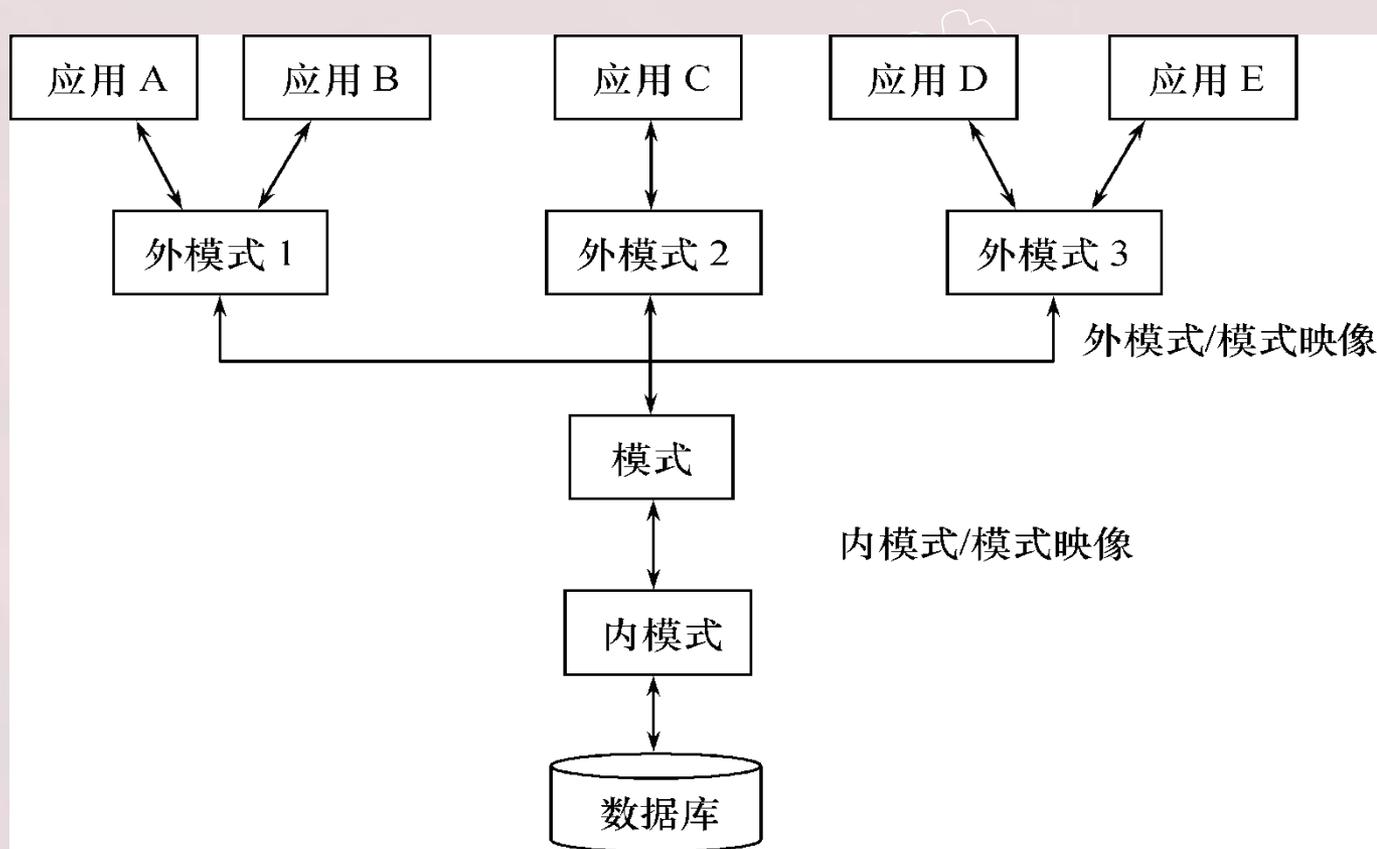


图1.16 数据库系统的三级模式结构



# 1. 模式 (Schema)

## ❖ 模式 (也称逻辑模式)

- 数据库中全体数据的逻辑结构和特征的描述
- 所有用户的公共数据视图

## ❖ 一个数据库只有一个模式

## ❖ 模式的地位：是数据库系统模式结构的中间层

- 与数据的物理存储细节和硬件环境无关
- 与具体的应用程序、开发工具及高级程序设计语言无关



# 模式（续）

## ❖ 模式的定义

- 数据的逻辑结构（数据项的名字、类型、取值范围等）
- 数据之间的联系
- 数据有关的安全性、完整性要求



## 2. 外模式 (External Schema)

### ❖ 外模式 (也称子模式或用户模式)

- 数据库用户 (包括应用程序员和最终用户) 使用的局部数据的逻辑结构和特征的描述
- 数据库用户的数据视图, 是与某一应用有关的数据的逻辑表示



# 外模式（续）

## ❖ 外模式的地位：介于模式与应用之间

### ■ 模式与外模式的关系：一对多

- 外模式通常是模式的子集
- 一个数据库可以有多个外模式。反映了不同的用户的应用需求、看待数据的方式、对数据保密的要求
- 对模式中同一数据，在外模式中的结构、类型、长度、保密级别等都可以不同

### ■ 外模式与应用的关系：一对多

- 同一外模式也可以为某一用户的多个应用系统所使用
- 但一个应用程序只能使用一个外模式



# 外模式（续）

## ❖ 外模式的用途

- 保证数据库安全性的一个有力措施
- 每个用户只能看见和访问所对应的外模式中的数据



# 3. 内模式 (Internal Schema)

## ❖ 内模式 (也称存储模式)

- 是数据物理结构和存储方式的描述

- 是数据在数据库内部的表示方式

- 记录的存储方式 (例如, 顺序存储, 按照B树结构存储, 按hash方法存储等)

- 索引的组织方式

- 数据是否压缩存储

- 数据是否加密

- 数据存储记录结构的规定

## ❖ 一个数据库只有一个内模式



# 数据库系统结构（续）

**1.3.1 数据库系统模式的概念**

**1.3.2 数据库系统的三级模式结构**

**1.3.3 数据库的二级映像功能与数据独立性**



# 数据库的二级映像功能与数据独立性

- ❖ 三级模式是对数据的三个抽象级别
- ❖ 二级映像 在数据库管理系统内部实现这三个抽象层次的联系和转换
  - 外模式 / 模式映像
  - 模式 / 内模式映像



# 1. 外模式 / 模式映像

- ❖ 模式：描述的是数据的全局逻辑结构
- ❖ 外模式：描述的是数据的局部逻辑结构
- ❖ 同一个模式可以有任意多个外模式
- ❖ 每一个外模式，数据库系统都有一个外模式 / 模式映像，定义外模式与模式之间的对应关系
- ❖ 映像定义通常包含在各自外模式的描述中



# 外模式 / 模式映像 (续)

## 保证数据的逻辑独立性

- 当模式改变时，数据库管理员对外模式 / 模式映像作相应改变，使外模式保持不变
- 应用程序是依据数据的外模式编写的，应用程序不必修改，保证了数据与程序的逻辑独立性，简称数据的逻辑独立性



## 2. 模式 / 内模式映像

- ❖ 模式 / 内模式映像定义了数据全局逻辑结构与存储结构之间的对应关系。
  - 例如，说明逻辑记录和字段在内部是如何表示的
- ❖ 数据库中模式 / 内模式映像是唯一的
- ❖ 该映像定义通常包含在模式描述中



# 模式 / 内模式映像 (续)

## ❖ 保证数据的物理独立性

- 当数据库的存储结构改变了（例如选用了另一种存储结构），数据库管理员修改模式 / 内模式映像，使模式保持不变。
- 应用程序不受影响。保证了数据与程序的物理独立性，简称数据的物理独立性。



# 模式 / 内模式映像 (续)

## ❖ 数据库模式

- 即全局逻辑结构是数据库的中心与关键
- 独立于数据库的其他层次
- 设计数据库模式结构时应首先确定数据库的逻辑模式



# 模式 / 内模式映像 (续)

## ❖ 数据库的内模式

- 依赖于它的全局逻辑结构
- 独立于数据库的用户视图，即外模式
- 独立于具体的存储设备
- 将全局逻辑结构中所定义的数据结构及其联系按照一定的物理存储策略进行组织，以达到较好的时间与空间效率



# 模式 / 内模式映像 (续)

## ❖ 数据库的外模式

- 面向具体的应用程序
- 定义在逻辑模式之上
- 独立于存储模式和存储设备
- 当应用需求发生较大变化，相应外模式不能满足其视图要求时，该外模式就得做相应改动
- 设计外模式时应充分考虑到应用的扩充性



# 模式 / 内模式映像 (续)

## ❖ 特定的应用程序

- 在外模式描述的数据结构上编制的
- 依赖于特定的外模式
- 与数据库的模式和存储结构独立
- 不同的应用程序有时可以共用同一个外模式



# 模式 / 内模式映像 (续)

## ❖ 数据库的二级映像

- 保证了数据库外模式的稳定性
- 从底层保证了应用程序的稳定性，除非应用需求本身发生变化，否则应用程序一般不需要修改



# 模式 / 内模式映像 (续)

- ❖ 数据与程序之间的独立性，使得数据的定义和描述可以从应用程序中分离出去
- ❖ 数据的存取由数据库管理系统管理
  - 简化了应用程序的编制
  - 大大减少了应用程序的维护和修改



# 第一章 绪论

1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统的结构

1.4 数据库系统的组成

1.5 小结



# 1.4 数据库系统的组成

- ❖ 数据库
- ❖ 数据库管理系统（及其开发工具）
- ❖ 应用程序
- ❖ 数据库管理员



# 数据库系统的组成（续）

- ❖ 硬件平台及数据库
- ❖ 软件
- ❖ 人员



# 1. 硬件平台及数据库

## ❖ 数据库系统对硬件资源的要求

- 足够大的内存
- 足够的大的磁盘或磁盘阵列等设备
- 较高的通道能力，提高数据传送率



## 2. 软件

- ❖ 数据库管理系统
- ❖ 支持数据库管理系统运行的操作系统
- ❖ 与数据库接口的高级语言及其编译系统
- ❖ 以数据库管理系统为核心的应用开发工具
- ❖ 为特定应用环境开发的数据库应用系统



# 3. 人员

- ❖ 数据库管理员
- ❖ 系统分析员和数据库设计人员
- ❖ 应用程序员
- ❖ 最终用户



# 人员（续）

■不同的人员涉及不同的数据抽象级别，具有不同的数据视图，如下图所示

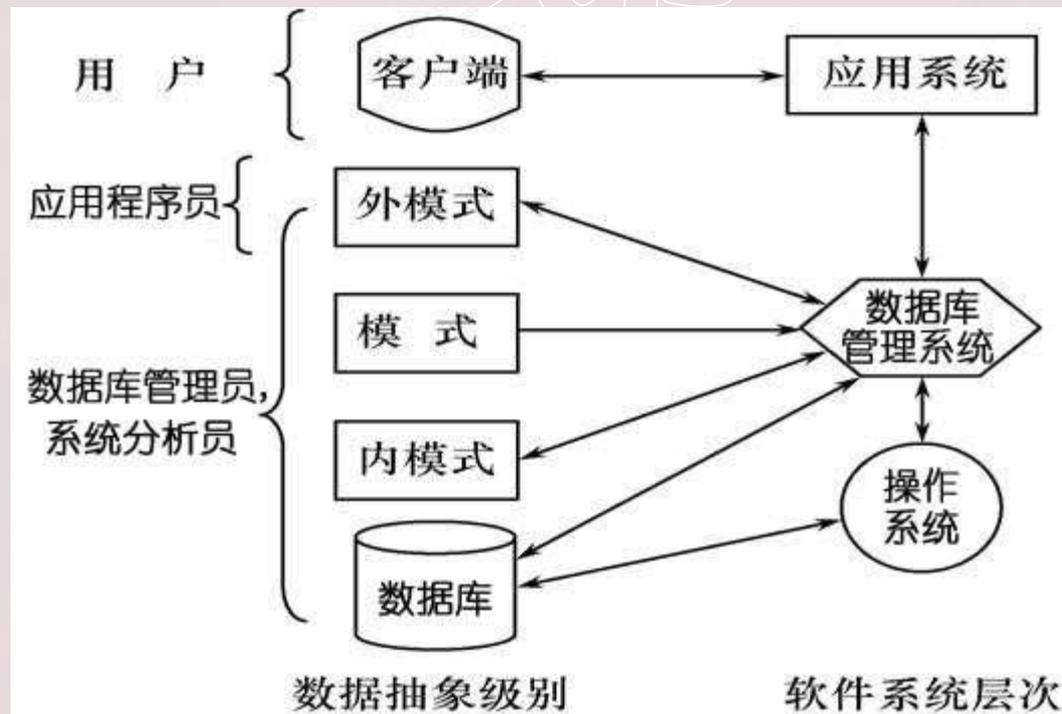


图1.17 各种人员的数据视图



# 1. 数据库管理员 (DBA)

具体职责:

1. 决定数据库中的信息内容和结构
2. 决定数据库的存储结构和存取策略
3. 定义数据的安全性要求和完整性约束条件



# 数据库管理员（续）

## 4. 监控数据库的使用和运行

### ■ 周期性转储数据库

- 数据文件
- 日志文件

### ■ 系统故障恢复

### ■ 介质故障恢复

### ■ 监视审计文件



# 数据库管理员（续）

## 5. 数据库的改进和重组

- 性能监控和调优
- 定期对数据库进行重组，以提高系统的性能
- 需求增加和改变时，数据库须需要重构造



## 2. 系统分析员和数据库设计人员

### ❖ 系统分析员

- 负责应用系统的需求分析和规范说明
- 与用户及数据库管理员结合，确定系统的软硬件配置
- 参与数据库系统的概要设计



# 系统分析员和数据库设计人员（续）

## ❖ 数据库设计人员

- 参加用户需求调查和系统分析
- 确定数据库中的数据
- 设计数据库各级模式



# 3. 应用程序员

- ❖ 设计和编写应用系统的程序模块
- ❖ 进行调试和安装



# 4. 用户

❖ 用户是指最终用户（**End User**）。最终用户通过应用系统的用户接口使用数据库。

## 1. 偶然用户

- 不经常访问数据库，但每次访问数据库时往往需要不同的数据库信息
- 企业或组织机构的高中级管理人员



# 用户（续）

## 2. 简单用户

- 主要工作是查询和更新数据库
- 银行的职员、机票预定人员、旅馆总台服务员

## 3. 复杂用户

- 工程师、科学家、经济学家、科技工作者等
- 直接使用数据库语言访问数据库，甚至能够基于数据库管理系统的应用程序接口编制自己的应用程序

# 第一章 绪论

## 1.1 数据库系统概述

## 1.2 数据模型

## 1.3 数据库系统的结构

## 1.4 数据库系统的组成

## 1.5 小结



# 1.5 小结

## ❖ 数据库系统概述

- 数据库的基本概念
- 数据管理的发展过程
- 数据库系统的特点

## ❖ 数据模型

- 数据模型的三要素
- 三种主要数据库模型



# 小结（续）

## ❖ 数据库系统内部的系统结构

- 数据库系统三级模式结构

- 数据库系统两层映像系统结构

## ❖ 数据库系统的组成

