



数据库系统概论

An Introduction to Database Systems

南京农业大学人工智能学院

2021年9月

教材及参考书(1)



教材

- 王珊，萨师煊：数据库系统概论(第5版)
高等教育出版社，2014.9
- 数据库系统教程（上册）
王能斌 电子工业出版社

教材及参考书(2)



参考书

- 王 珊：数据库系统概论学习指导与习题解答
高等教育出版社，2006.6
- 单建魁：数据库系统实验指导
清华大学出版社2004年12月
- 史嘉权：数据库系统概论---习题、实验与考试辅导
清华大学出版社2006年2月

<http://www.chinadb.org/>

上机软件

- KingBase ES 教学版
系统可以从人大金仓公司的网站<http://www.kingbase.com.cn/>上免费下载
- SQL server 2010

学习方式



听课

(启发式、讨论式)

读书

(预习、复习)

课外实验 (计划外)

(上机练习)



考试成绩



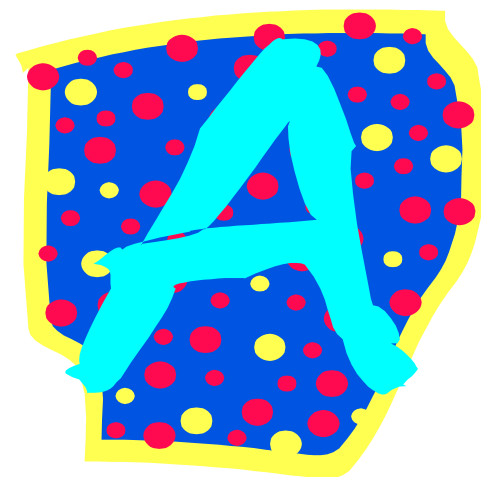
❖ 平时成绩:

(书面作业、考勤、课堂提问)

❖ 阶段测试: 1-2次

(综合练习)

❖ 期末考试(不超过70%)



内容安排(1)



基础篇

- 第一章 绪论
- 第二章 关系数据库
- 第三章 关系数据库标准语言SQL
- 第四章 数据库安全性
- 第五章 数据库完整性

设计与应用开发篇

- 第六章 关系数据理论
- 第七章 数据库设计
- 第八章 数据库编程

内容安排(2)



系统篇

- 第九章 关系查询处理和查询优化
- 第十章 数据库恢复技术
- 第十一章 并发控制
- * 第十二章 数据库管理系统

第一至第十一章是本科专业的基本教程（书中有*号的部分除外）
第十二至第十七章是高级教程



数据库系统概论

An Introduction to Database Systems

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，是数据管理的最新技术，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。
- ❖ 数据库的建设规模、数据库信息量的大小和使用频度已成为衡量一个国家信息化程度的重要标志。

第一章 绪论



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

1.1.1 四个基本概念



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

一、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
 - 文本、图形、图像、音频、视频、学生的档案记录、货物的运输情况等
- ❖ 数据的特点
 - 数据与其语义是不可分的

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2003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，197205，江苏南京市，计算机系，1990）

- 语义：学生姓名、性别、出生年月、籍贯、所在院系、
入学时间
- 解释：李明是个大学生，1972年5月出生，江苏南京市人，
1990年考入计算机系

请给出另一个解释和语义

二、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

三、数据库管理系统



❖ 什么是DBMS

- 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

❖ DBMS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中的位置



DBMS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

DBMS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DBMS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

DBMS的主要功能



- **数据库的建立和维护功能(实用程序)**

- 数据库初始数据装载转换

- 数据库转储

- 介质故障恢复

- 数据库的重组织

- 性能监视分析等

- **其它功能**

- DBMS与网络中其它软件系统的通信

- 两个DBMS系统的数据转换

- 异构数据库之间的互访和互操作



典型的DBMS: MS---- ACCESS

IBM DB/2

ORACLE ORACLE

此外: sybase informix MS SQL Server

四、数据库系统



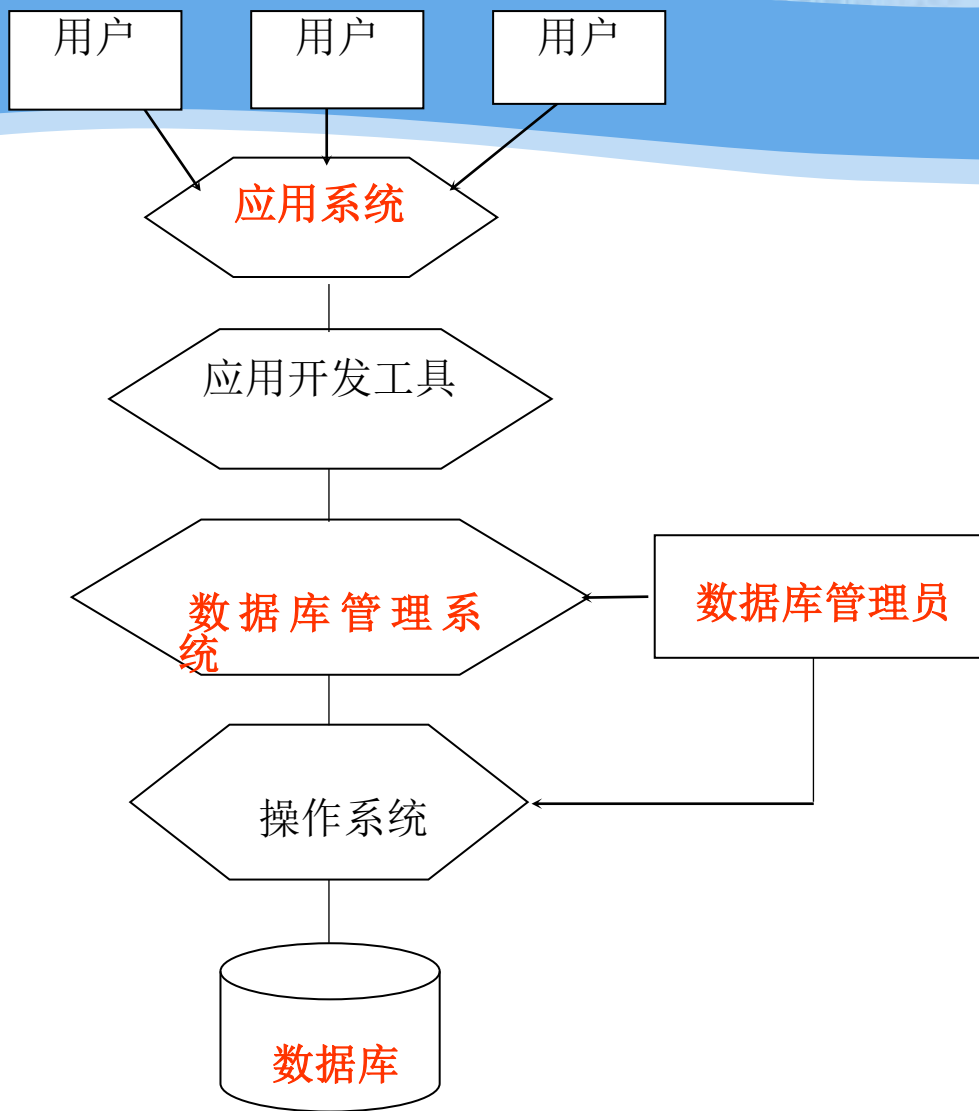
❖ 什么是数据库系统（**Database System**，简称**DBS**）

在计算机系统中引入数据库后的系统构成

❖ 数据库系统的构成

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

（数据库管理员、系统分析员与数据库设计人员、应用程序员、用户）



数据库系统



某大公司工资系统。指出那些人员可以执行下面的功能：

- 1 编写一个应用程序以生成和打印帐单（程序员）
- 2 改变一个已经搬迁的员工在数据库中的地址信息（数据库管理员）
- 3 为新来的职工建立一个新的用户帐号（数据库管理员）

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40年代中--50年代中**)
- 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

一、人工管理阶段



❖ 时期

- 20世纪40年代中--50年代中

❖ 产生的背景

- 应用需求 科学计算
- 硬件水平 无直接存取存储设备
- 软件水平 没有操作系统
- 处理方式 批处理

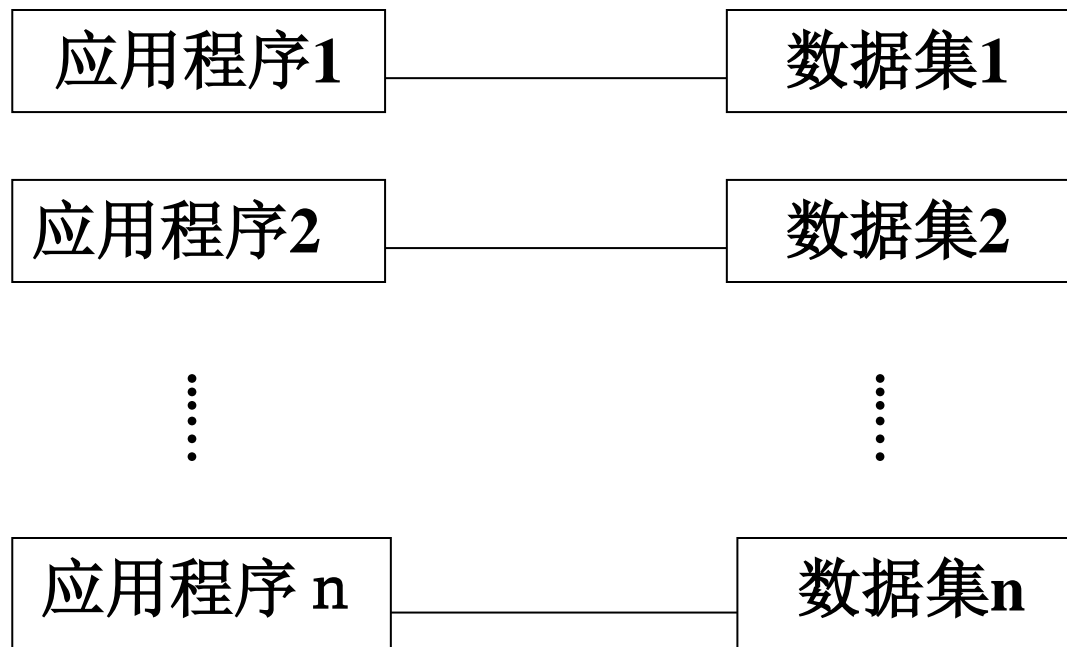
人工管理阶段(续)



❖ 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

二、文件系统阶段



❖ 时期

- 20世纪50年代末--60年代中

❖ 产生的背景

- 应用需求 科学计算、管理
- 硬件水平 磁盘、磁鼓
- 软件水平 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：某一应用程序

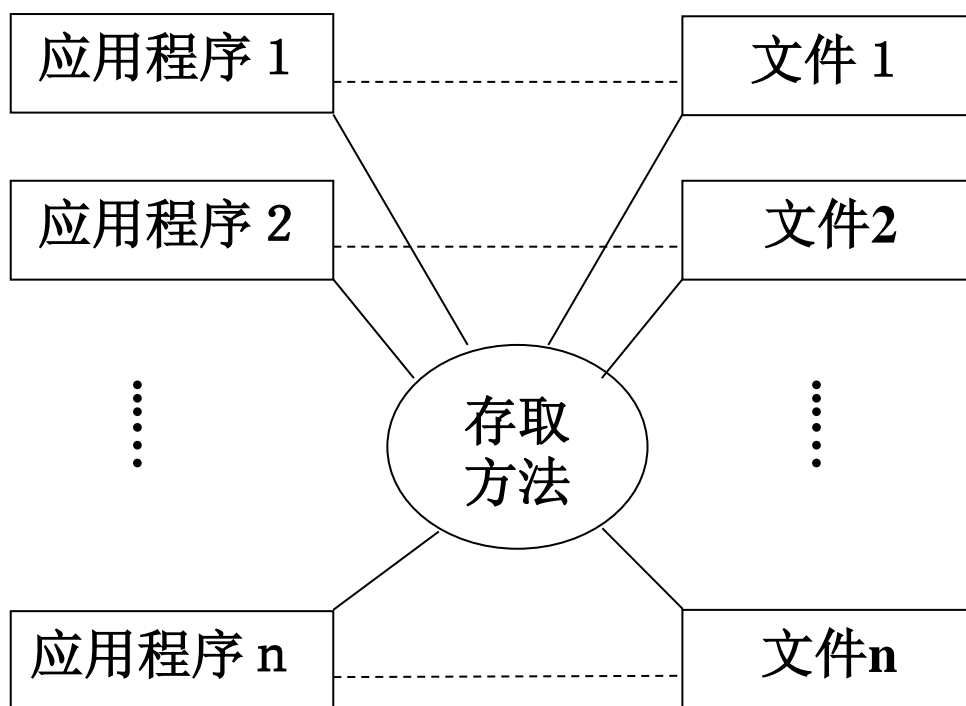
数据的共享程度：共享性差、冗余度大

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，数据的逻辑结构改变必须
修改应用程序

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

文件系统中数据的结构



- ❖ 记录内有结构。
 - 可以间接实现数据变长要求，但访问相应数据的应用程序复杂了。
- ❖ 数据的结构是靠程序定义和解释的。
- ❖ 数据只能是定长的。
 - 可以间接实现数据变长要求，但访问相应数据的应用程序复杂了。
- ❖ 文件间是独立的，因此数据整体无结构。
 - 可以间接实现数据整体的有结构，但必须在应用程序中对描述数据间的联系。
- ❖ 数据的最小存取单位是记录。

三、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

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

1.1.3 数据库系统的特点



- ❖ 数据结构化：整体数据结构化
- ❖ 数据的共享性高，冗余度低，易扩充
- ❖ 数据独立性高
- ❖ 数据由**DBMS**统一管理和控制

数据结构化



❖ 整体数据的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

❖ 数据库中实现的是数据的真正结构化

- 数据的结构用数据模型描述，无需程序定义和解释
- 数据可以变长
- 数据的最小存取单位是数据项

数据的共享性高，冗余度低，易扩充



- ❖ 数据库系统从整体角度看待和描述数据，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存储在磁盘上的数据库中数据是相互独立的。
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，用户程序也可以不变。

❖ 数据独立性是由DBMS的二级映像功能来保证的

数据由DBMS统一管理和控制



❖ DBMS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

将数据控制在有效的范围内，或保证数据之间满足一定的关系。

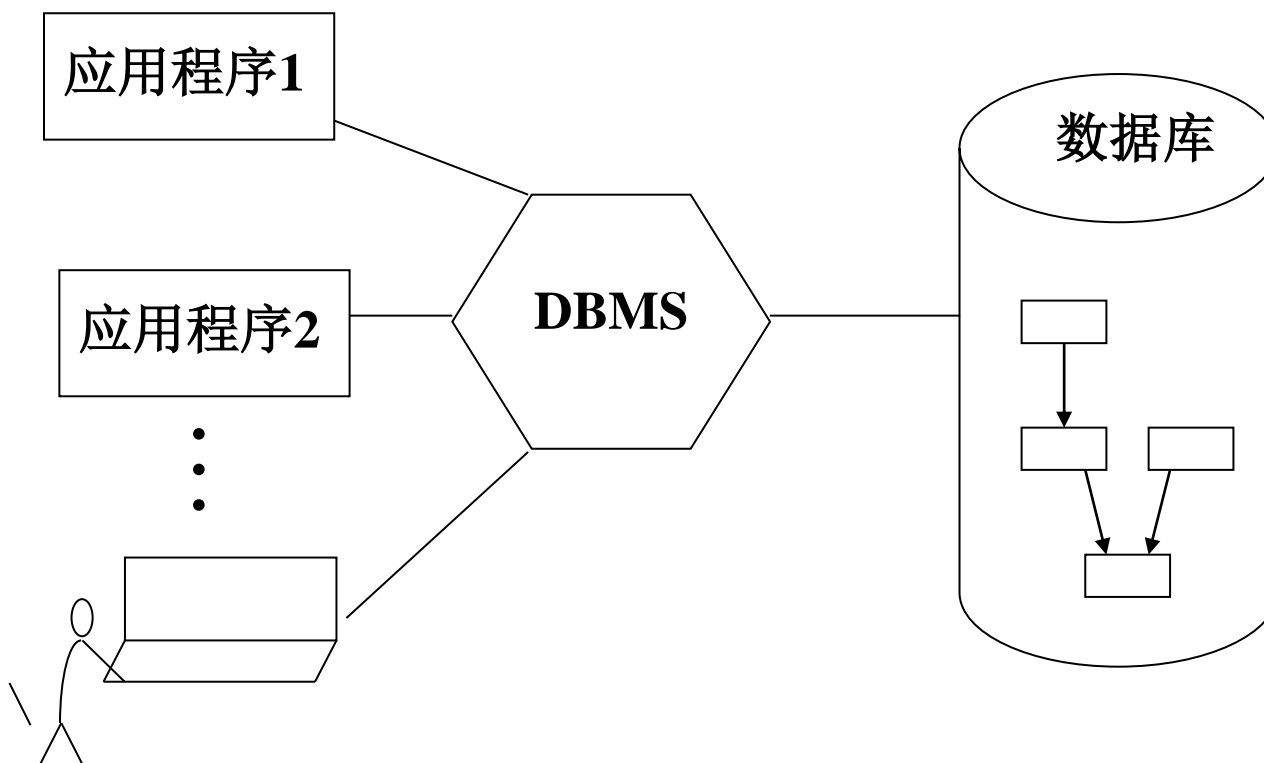
■ (3)并发（Concurrency）控制

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)



数据库系统阶段应用程序与数据之间的对应关系

第一章 绪论



1.1 数据库系统概述

1.2 数据模型

1.3 数据库系统结构

1.4 数据库系统的组成

1.5 小结

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. 现实世界

现实世界即客观存在的世界，各种事物及事物之间的联系。一个事物可以有許多特征，通常都是选用人们感兴趣的以及最能表征该事物的若干特征来描述该事物。以人为例，常选用姓名、性别、年龄、籍贯等描述一个人的特征。事物间的关联是多方面的。

❖ 2. 信息世界

现实世界中的事物及其联系由人们的感官感知，经过人们头脑的分析、归纳、抽象，形成信息。对这些信息进行记录、整理、归类和格式化后，它们就构成了信息世界。对所研究的信息世界建立一个抽象的模型，称之为信息模型（即概念模型）。目前较为流行的一种信息模型是实体联系模型。

❖ 3. 机器世界

用计算机管理信息，必须对信息进行数据化，数据化后的信息称之为数据，数据是能够被机器识别并处理的。数据化了的信息世界称之为机器世界。

数据模型



- ❖ 在数据库中用数据模型这个工具来抽象、表示和处理现实世界中的数据和信息。
- ❖ 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

1.2.1 两大类数据模型



❖ 数据模型分为两类（分属两个不同的层次）

(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。

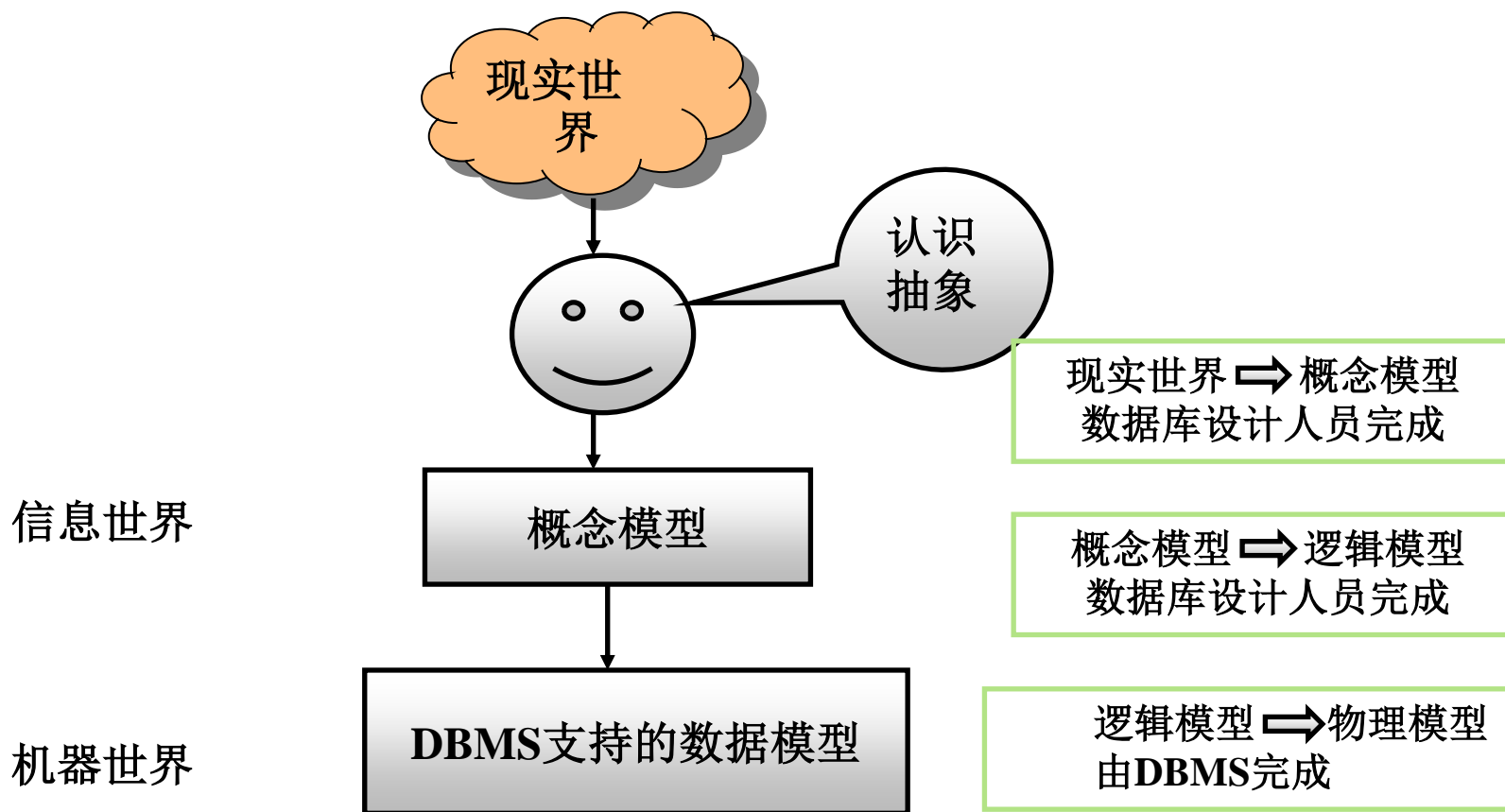
两大类数据模型 (续)



❖ 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；
- 把概念模型转换为某一**DBMS**支持的数据模型。

两大类数据模型 (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 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

1.2.2 数据模型的组成要素



- ❖ 数据结构
- ❖ 数据操作
- ❖ 完整性约束条件

一、数据结构



❖ 什么是数据结构

- 描述数据库的组成对象，以及对象之间的联系

❖ 描述的内容

- 与数据类型、内容、性质有关的对象
- 与数据之间联系有关的对象

❖ 数据结构是对系统静态特性的描述

二、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

三、数据的完整性约束条件



❖ 数据的完整性约束条件

- 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储存规则
- 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、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

- 反映和规定本数据模型必须遵守的基本的通用的完整性约束条件。例如在关系模型中，任何关系必须满足实体完整性和参照完整性两个条件。
- 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 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 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
- 能够方便、直接地表达应用中的各种语义知识
- 简单、清晰、易于用户理解

一、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) 域 (Domain)

属性的取值范围称为该属性的域。

(5) 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

(6) 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)



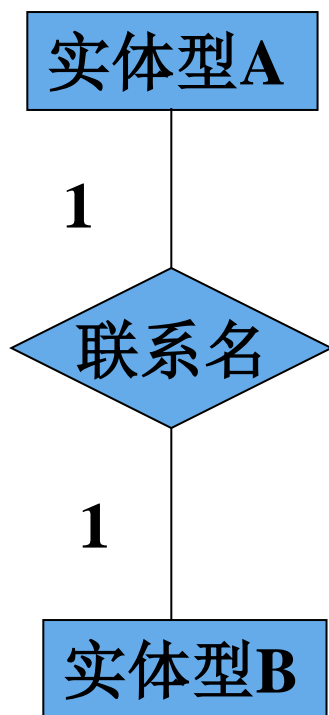
(7) 联系 (Relationship)

- 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内部的联系和实体之间的联系。
- **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

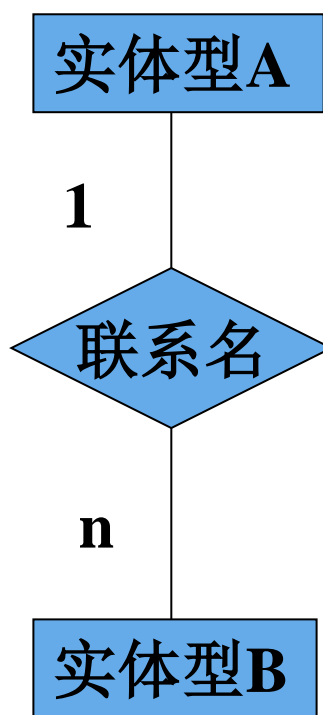
二、两个实体型之间的联系



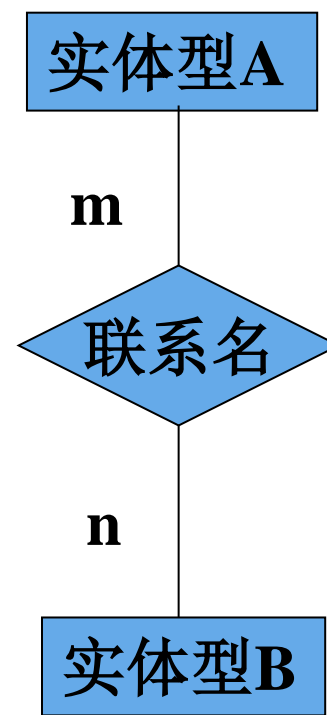
用图形来表示两个实体型之间的这三类联系



1:1联系



1:n联系



m:n联系

二、两个实体型之间的联系（续）



❖ 一对一联系（1:1）

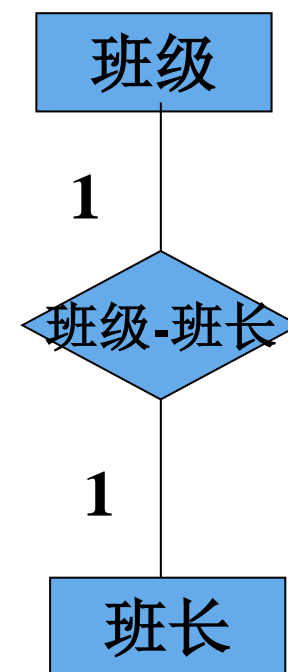
■ 实例

一个班级只有一个正班长

一个班长只在一个班中任职

■ 定义：

如果对于实体集**A**中的每一个实体，实体集**B**中至多有一个（也可以没有）实体与之联系，反之亦然，**则称实体集A与实体集B具有一对一联系，记为1:1**



1:1联系

两个实体型之间的联系 (续)



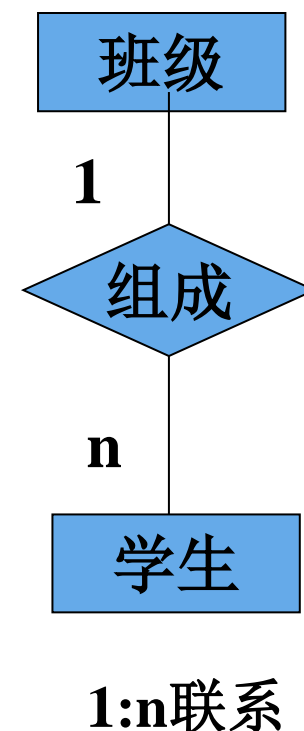
❖ 一对多联系 (1: n)

■ 实例

一个班级中有若干名学生，
每个学生只在一个班级中学习

■ 定义：

如果对于实体集**A**中的每一个实体，实体集**B**中有**n**个实体 ($n \geq 0$) 与之联系，反之，对于实体集**B**中的每一个实体，实体集**A**中至多只有一个实体与之联系，则称**实体集A与实体集B**有一对多联系，记为**1:n**



两个实体型之间的联系 (续)



❖ 多对多联系 (m:n)

■ 实例

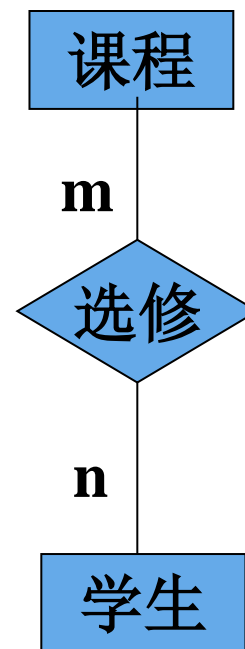
课程与学生之间的联系：

一门课程同时有若干个学生选修

一个学生可以同时选修多门课程

■ 定义：

如果对于实体集**A**中的每一个实体，实体集**B**中有**n**个实体 ($n \geq 0$) 与之联系，反之，对于实体集**B**中的每一个实体，实体集**A**中也有**m**个实体 ($m \geq 0$) 与之联系，则称实体集**A**与实体**B**具有多对多联系，记为**m:n**



m:n联系

三、两个以上实体型之间的联系



❖ 两个以上实体型之间一对多联系

- 若实体集 E_1, E_2, \dots, E_n 存在联系, 对于实体集 E_j ($j=1, 2, \dots, i-1, i+1, \dots, n$) 中的给定实体, 最多只和 E_i 中的一个实体相联系, 则我们说 E_i 与 $E_1, E_2, \dots, E_{i-1}, E_{i+1}, \dots, E_n$ 之间的联系是一对多的

两个以上实体型之间的联系(续)



❖ 实例

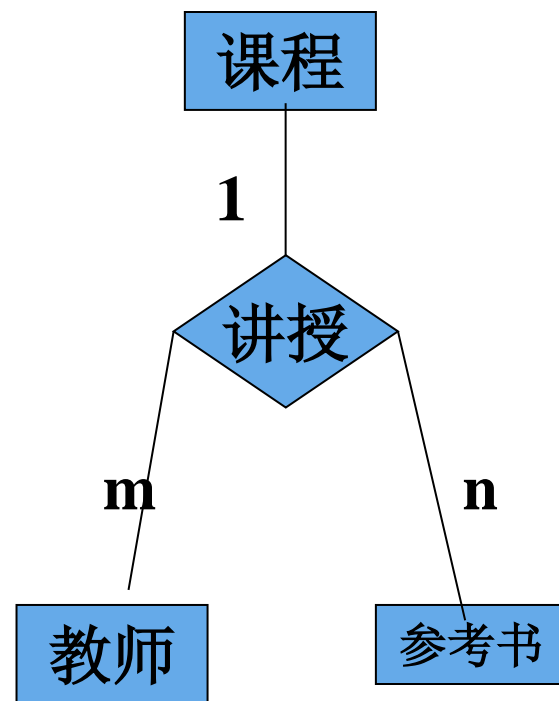
课程、教师与参考书三个实体型

一门课程可以有若干个教师讲授,

使用若干本参考书,

每一个教师只讲授一门课程,

每一本参考书只供一门课程使用



两个以上实体型间1:n联系

两个以上实体型之间的联系(续)



- ❖ 多个实体型间的一对一联系
- ❖ 两个以上实体型间的多对多联系

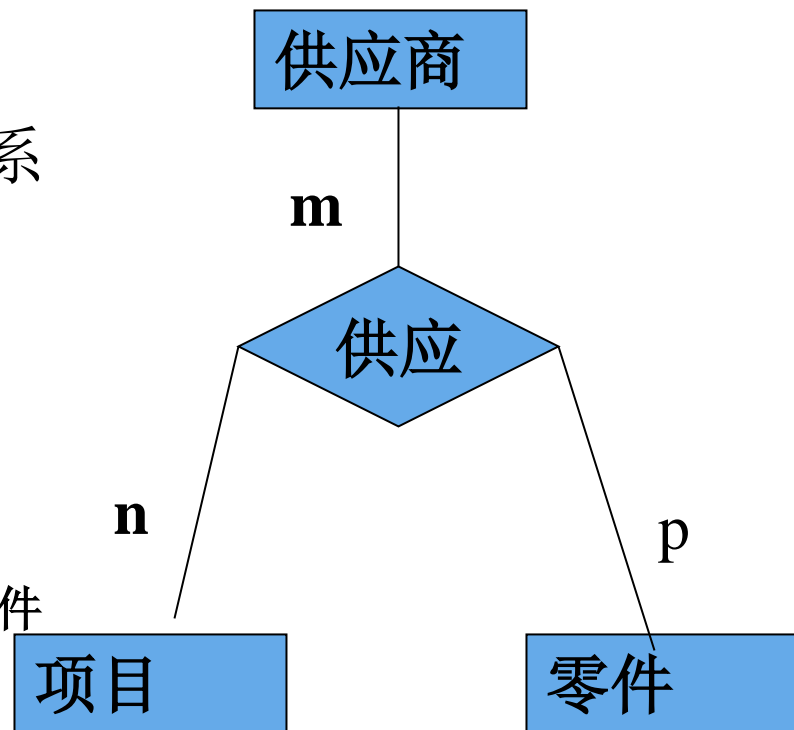
- 实例

供应商、项目、零件三个实体型

一个供应商可以供给多个项目多种零件

每个项目可以使用多个供应商供应的零件

每种零件可由不同供应商供给



两个以上实体型间m:n联系

四、单个实体型内的联系



❖ 一对多联系

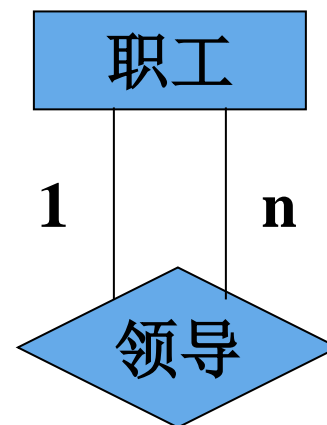
■ 实例

职工实体型内部具有领导与被领导的联系

某一职工（干部）“领导”若干名职工

一个职工仅被另外一个职工直接领导

这是一对多的联系



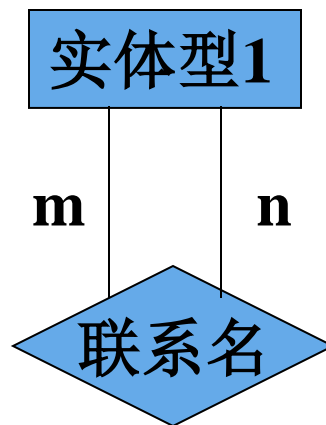
单个实体型内部
1:n联系

❖ 一对一联系

单个实体型内的联系



❖ 多对多联系



单个实体型内的
m:n联系

五、 概念模型的一种表示方法



❖ 实体—联系方法(E-R方法)

- 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

E-R图



❖ 实体型

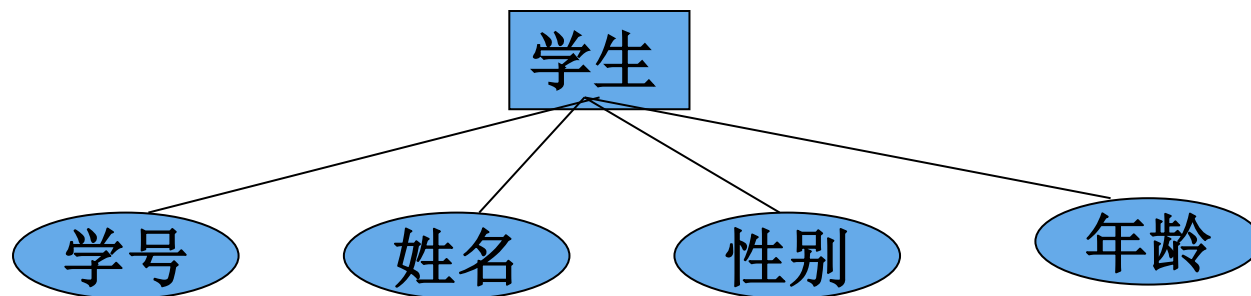
用矩形表示，矩形框内写明实体名。

学生

教师

❖ 属性

用椭圆形表示，并用无向边将其与相应的实体连接起来



E-R图(续)

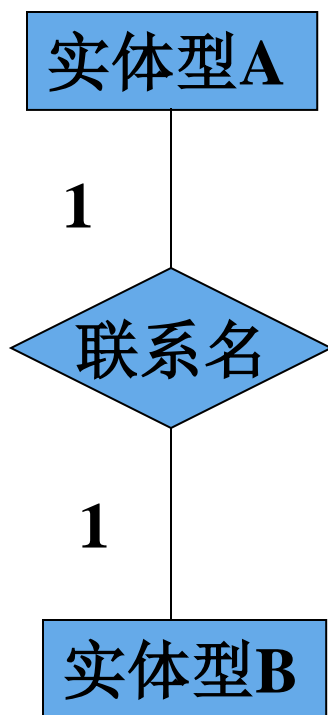


❖ 联系

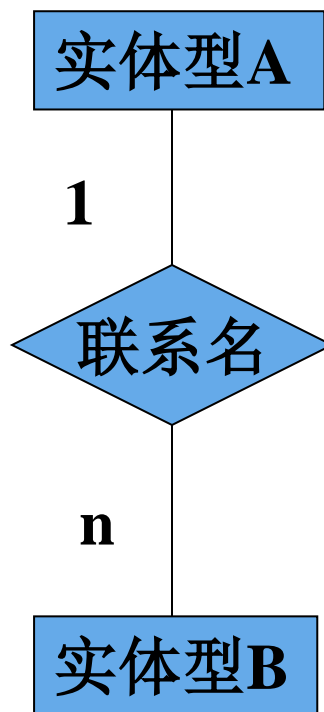
■ 联系本身：

用菱形表示，菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时，在无向边旁标上联系类型（1:1、1:n或m:n）

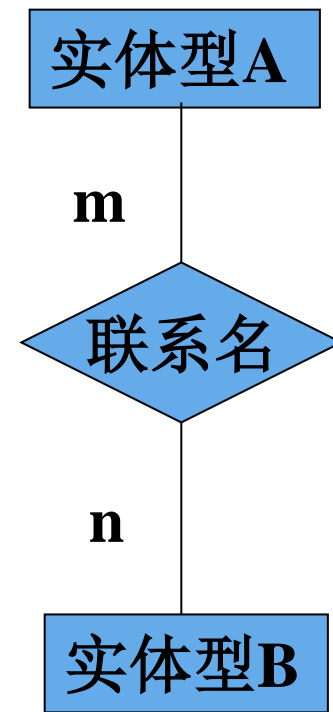
联系的表示方法



1:1联系

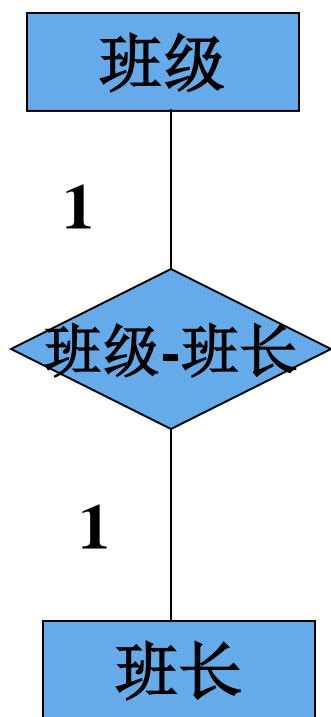


1:n联系

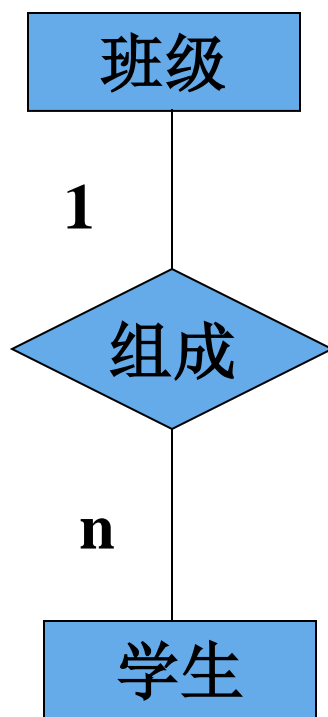


m:n联系

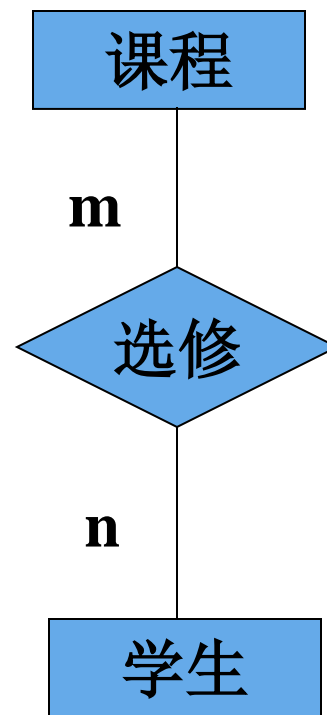
联系的表示方法示例



1:1联系



1:n联系



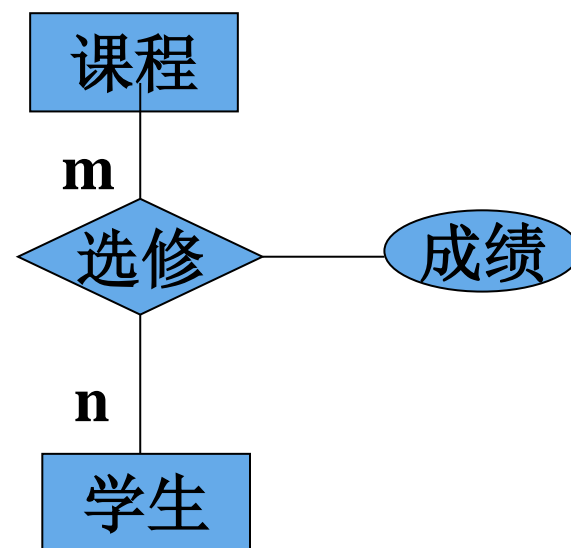
m:n联系

联系的属性



❖ 联系的属性:

联系本身也是一种实体型，也可以有属性。如果一个联系具有属性，则这些属性也要用无向边与该联系连接起来



六、一个实例



用E-R图表示某个工厂物资管理的概念模型

❖ 实体

- 仓库： 仓库号、面积、电话号码
- 零件： 零件号、名称、规格、单价、描述
- 供应商： 供应商号、姓名、地址、电话号码、帐号
- 项目： 项目号、预算、开工日期
- 职工： 职工号、姓名、年龄、职称

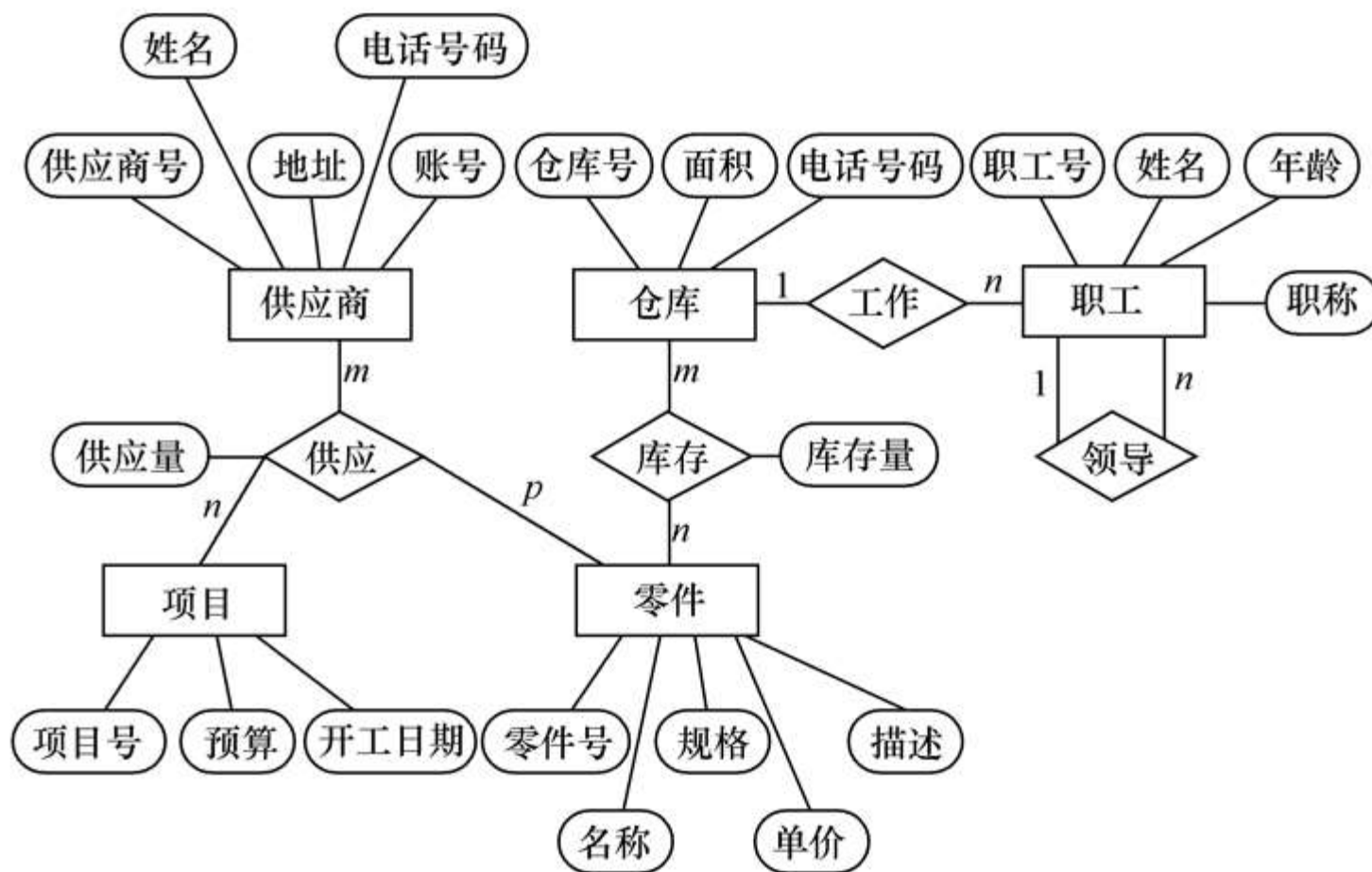
一个实例



❖ 实体之间的联系如下：

- (1) 一个仓库可以存放多种零件，一种零件可以存放在多个仓库中。仓库和零件具有多对多的联系。用库存量来表示某种零件在某个仓库中的数量。
- (2) 一个仓库有多个职工当仓库保管员，一个职工只能在一个仓库工作，仓库和职工之间是一对多的联系。职工实体型中具有一对多的联系
- (3) 职工之间具有领导-被领导关系。即仓库主任领导若干保管员。
- (4) 供应商、项目和零件三者之间具有多对多的联系

一个实例



(c) 完整的实体-联系图

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

1.2.4 最常用的数据模型



❖ 非关系模型

- 层次模型(Hierarchical Model)
- 网状模型(Network Model)

❖ 关系模型(Relational Model)

❖ 面向对象模型(Object Oriented Model)

❖ 对象关系模型(Object Relational Model)

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1968年推出的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

一、层次数据模型的数据结构



❖ 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

层次数据模型的数据结构(续)

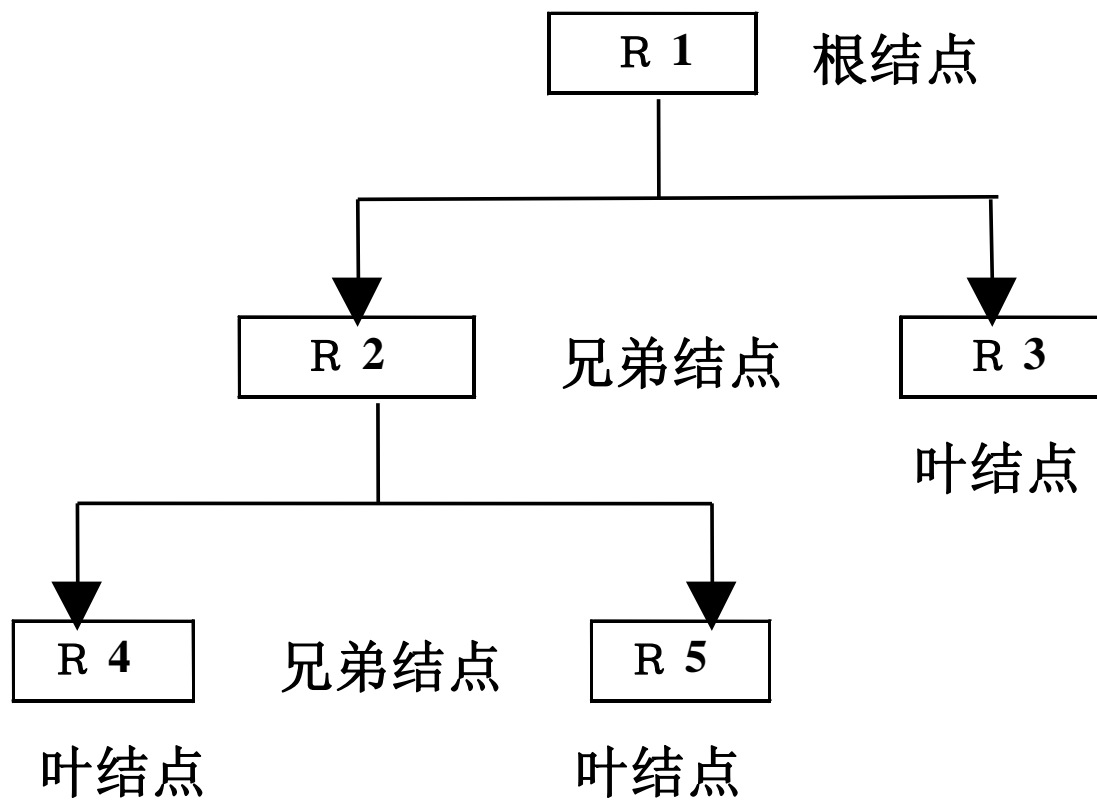


图1.16 一个层次模型的示例

层次数据模型的数据结构(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
- 在层次模型中具有一定的存取路径，它仅允许自顶向下的单向查询。
- 没有一个子女记录值能够脱离双亲记录值而独立存在

层次数据模型的数据结构(续)

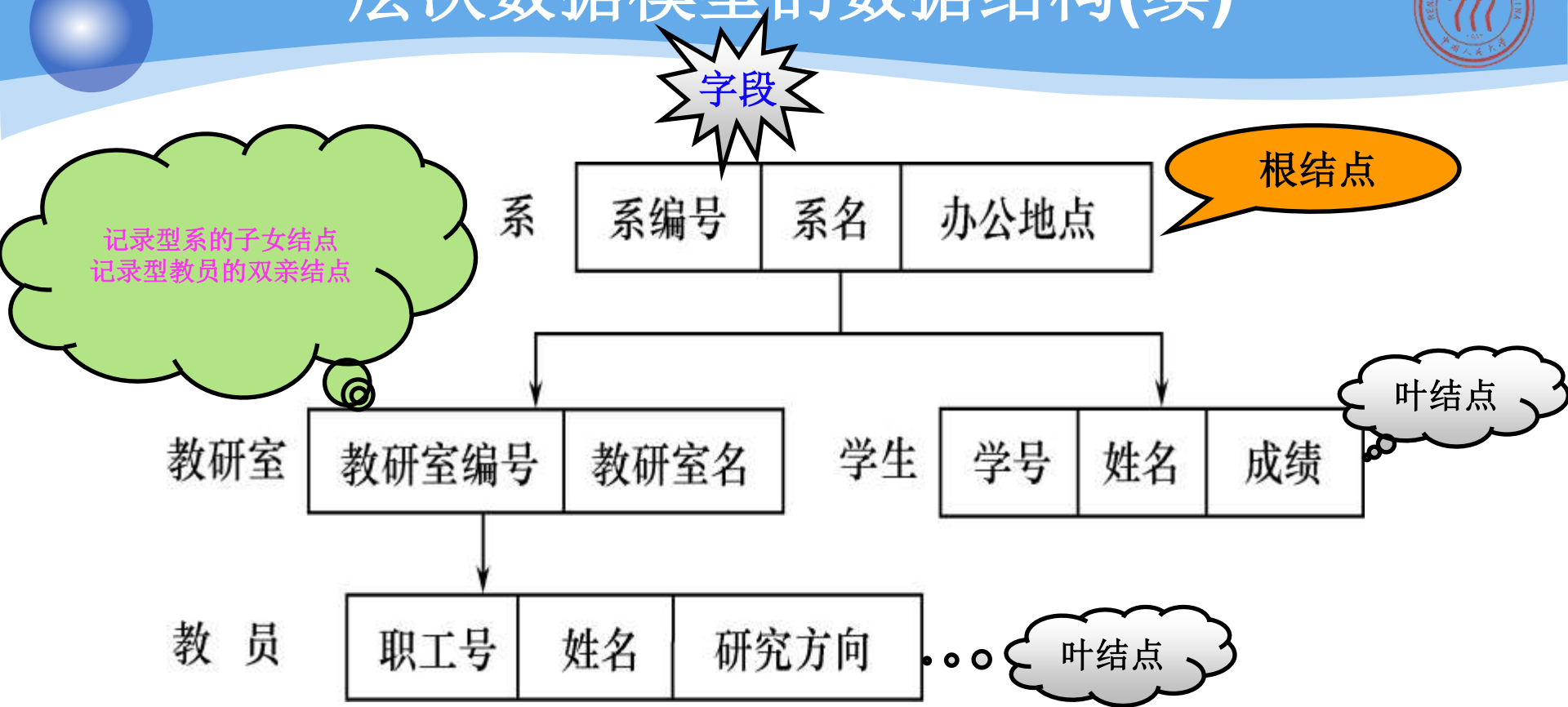


图1.17 教员学生层次数据库模型

层次数据模型的数据结构(续)

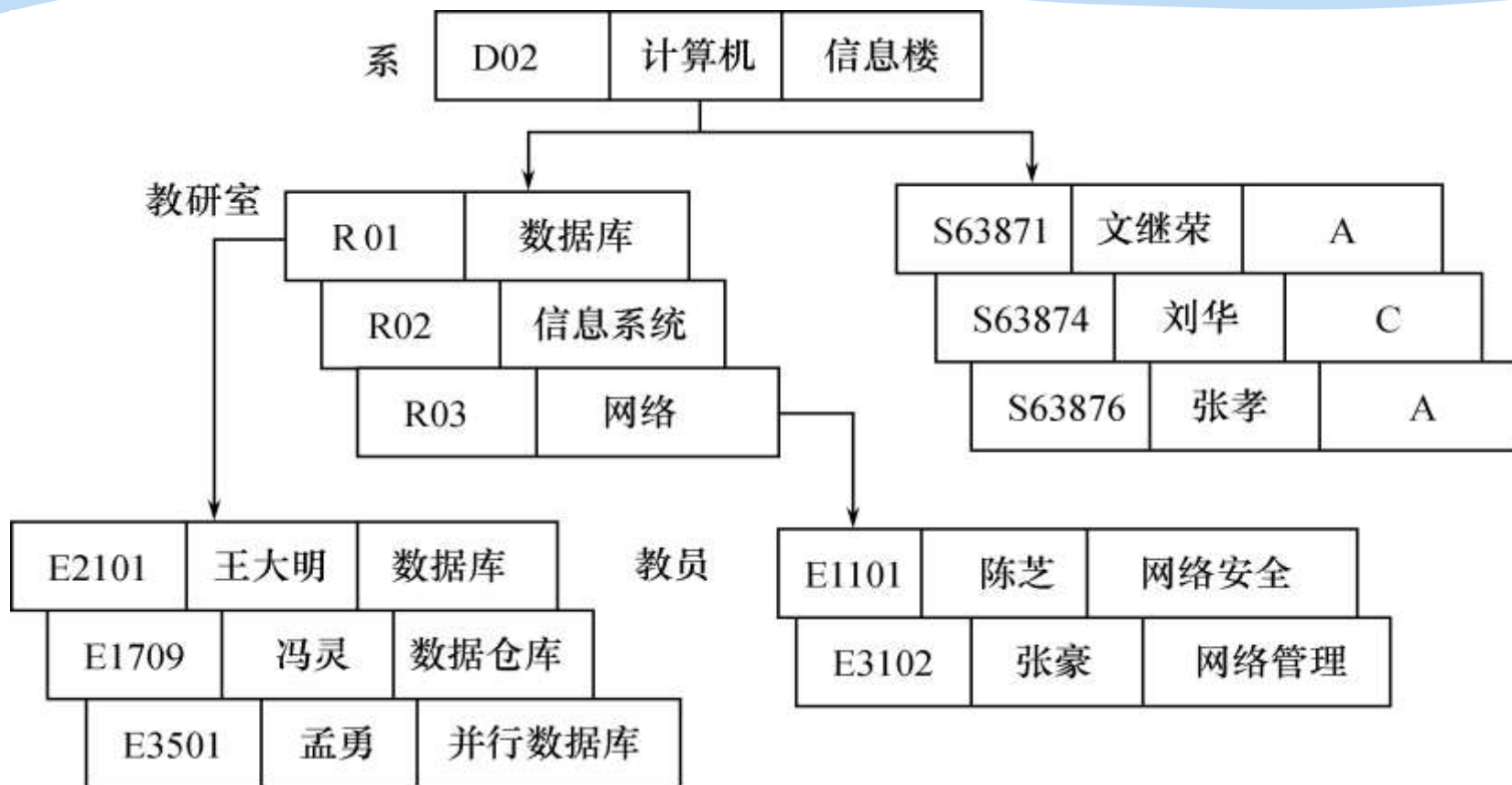


图1.18 教员学生层次数据库的一个值

二、多对多联系在层次模型中的表示



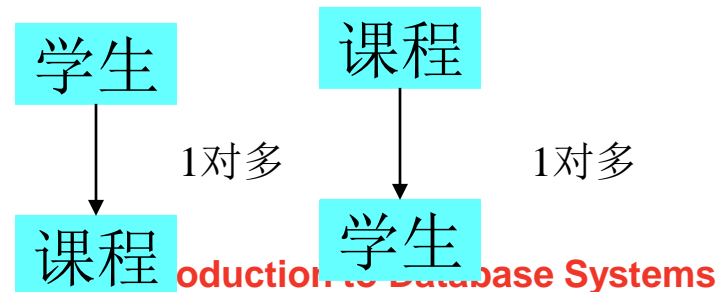
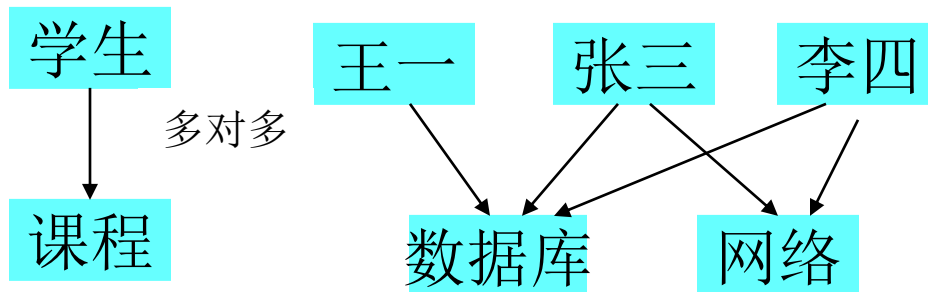
❖ 多对多联系在层次模型中的表示

- 用层次模型间接表示多对多联系
- 方法

将多对多联系分解成一对多联系

- 分解方法

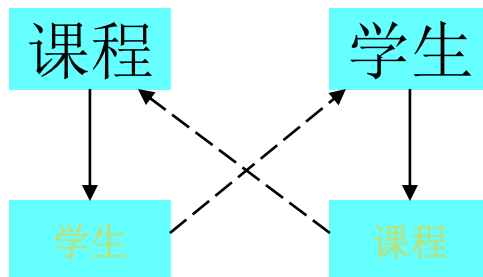
- 冗余结点法



三、层次模型的数据操纵与完整性约束



❖ 虚拟结点法



❖ 层次模型的数据操纵

- 查询
- 插入
- 删除
- 更新

层次模型的数据操纵与完整性约束（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

四、层次数据模型的存储结构



❖ 邻接法

按照层次树前序遍历的顺序把所有记录值依次邻接存放，即通过物理空间的位置相邻来实现层次顺序

层次数据模型的存储结构（续）



按邻接法存放图1.20（b）中以根记录A1为首的层次记录实例集

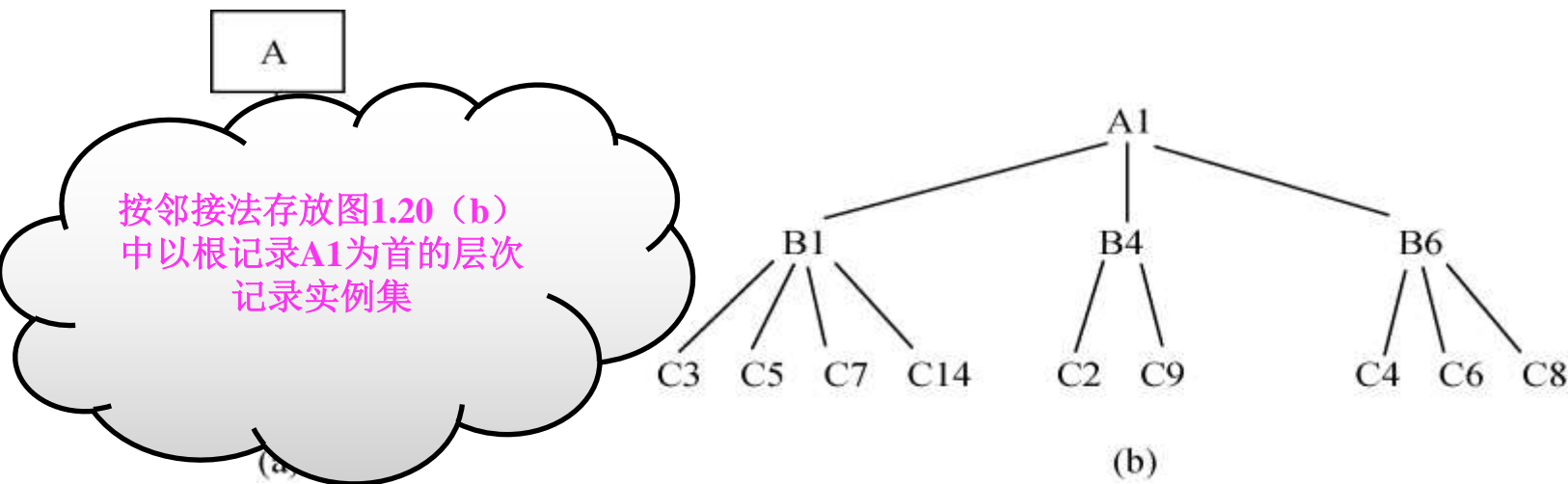


图1.20 层次数据库及其实例

A1	B1	C3	C5	C7	C14	B4	C2	C9	B6	C4	C6	C8	A2	A1	...
----	----	----	----	----	-----	----	----	----	----	----	----	----	----	----	-----

图1.21 邻接法

层次数据模型的存储结构（续）



❖ 链接法

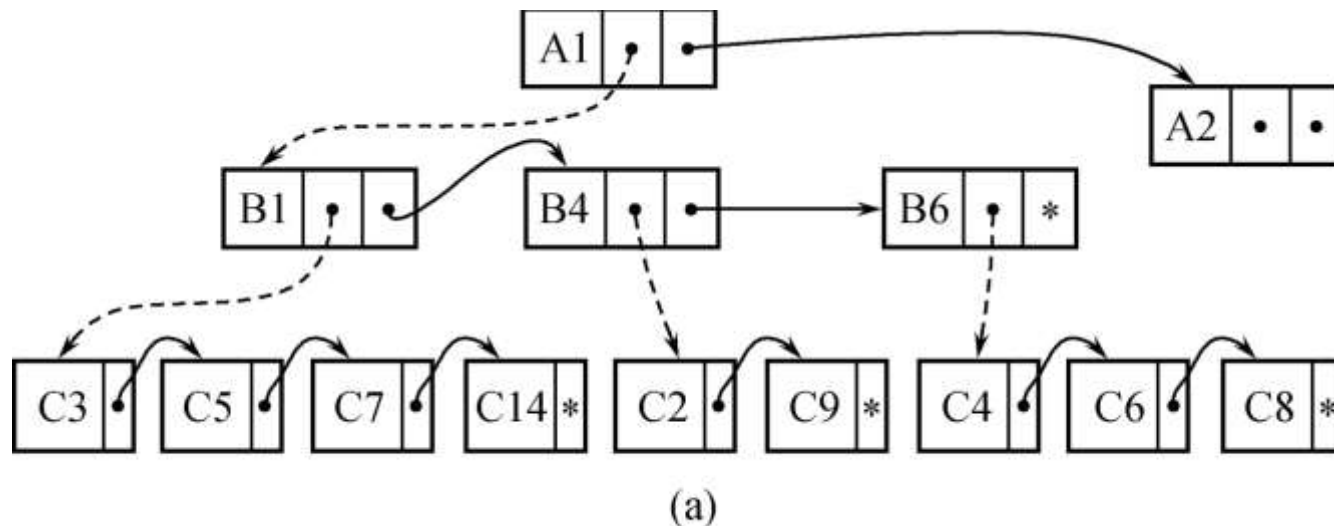
- 用指针来反映数据之间的层次联系
- 子女—兄弟链接法
- 层次序列链接法

层次数据模型的存储结构（续）



■ 子女-兄弟链接法

每个记录设两类指针，分别指向最左边的子女（每个记录型对应一个）和最近的兄弟

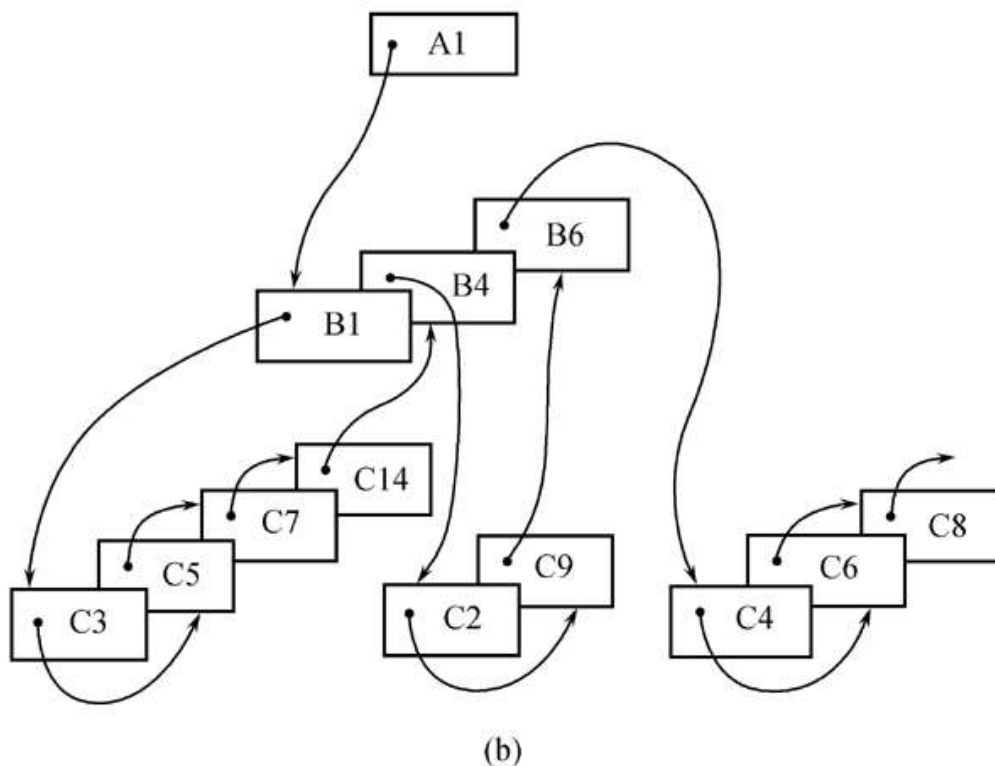


层次数据模型的存储结构（续）



❖ 层次序列链接法

按树的前序穿越顺序链接各记录值



五、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

- 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
- 由于结构严密，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（数据系统语言研究会）系统
 - 70年代由DBTG（Data Base Task Group）提出的一个系统方案
 - 奠定了数据库系统的基本概念、方法和技术
- ❖ 实际系统
 - Cullinet Software Inc.公司的 IDMS
 - Univac公司的 DMS1100
 - Honeywell公司的IDS/2
 - HP公司的IMAGE

1.网状数据模型的数据结构



❖ 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. 一个结点可以有多于一个的双亲。

网状数据模型的数据结构（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

网状数据模型的数据结构（续）



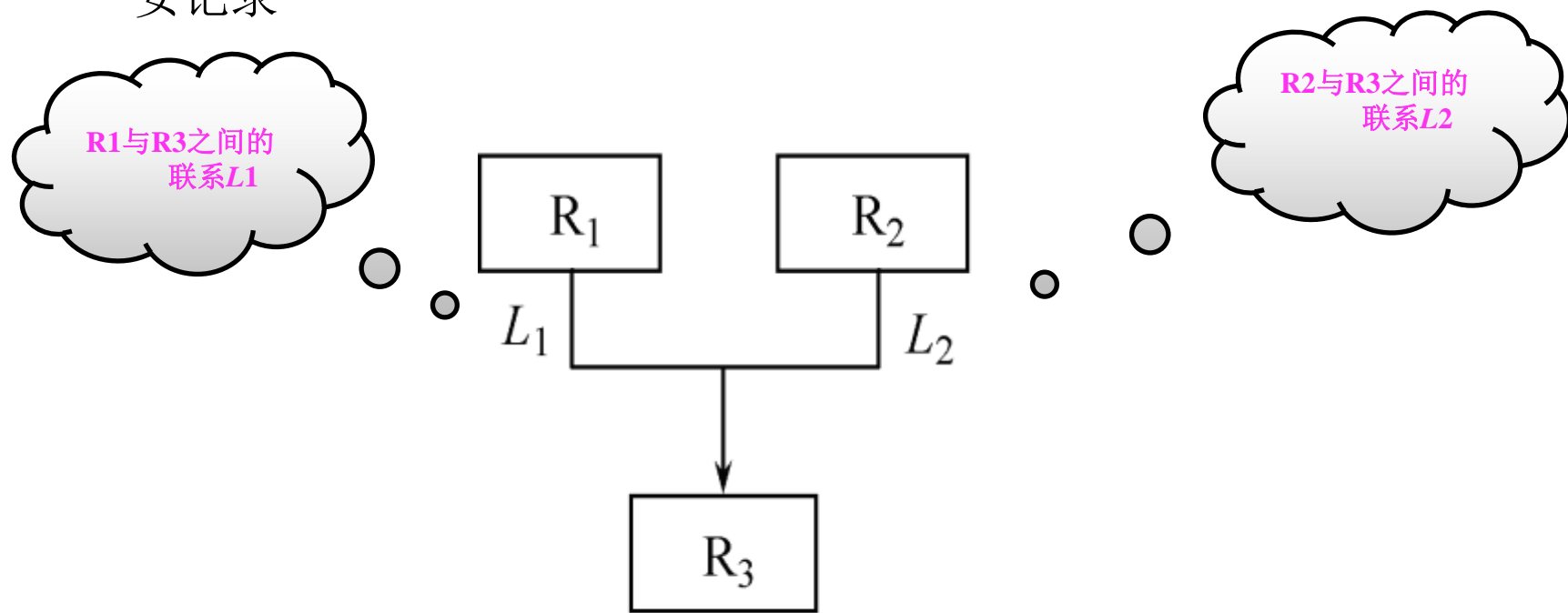
❖ 网状模型与层次模型的区别

- 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去描述现实世界
- 层次模型实际上是网状模型的一个特例

网状数据模型的数据结构（续）

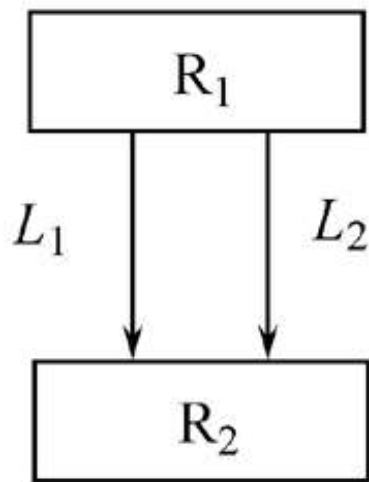


- ❖ 网状模型中子女结点与双亲结点的联系可以不唯一
要为每个联系命名，并指出与该联系有关的双亲记录和子女记录

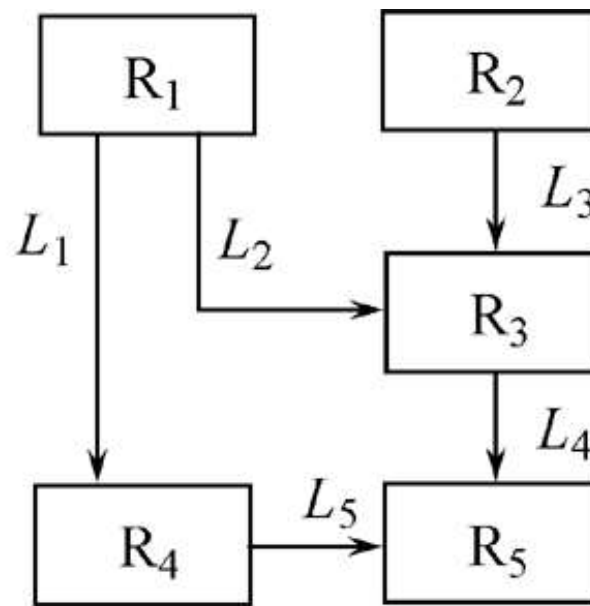


(a)

网状数据模型的数据结构（续）



(b)



(c)

网状模型的例子

网状数据模型的数据结构（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

网状数据模型的数据结构（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系记录，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

网状数据模型的数据结构（续）

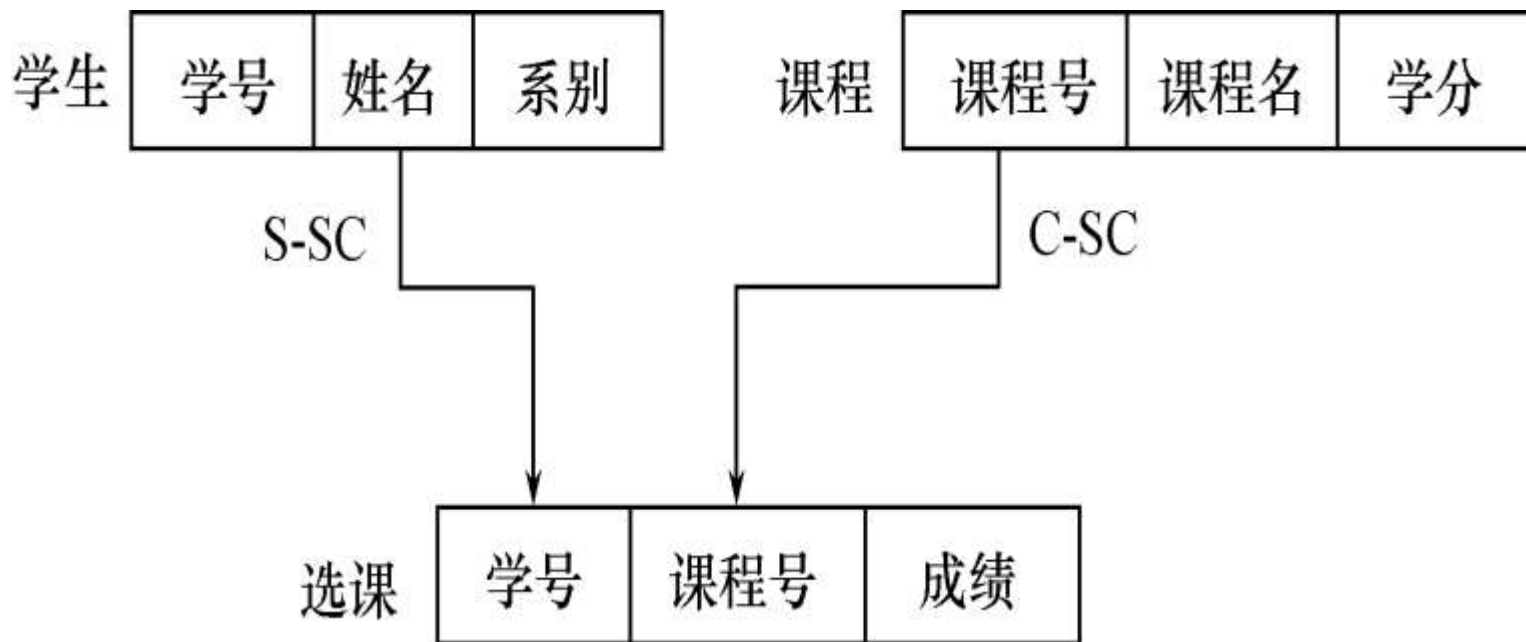
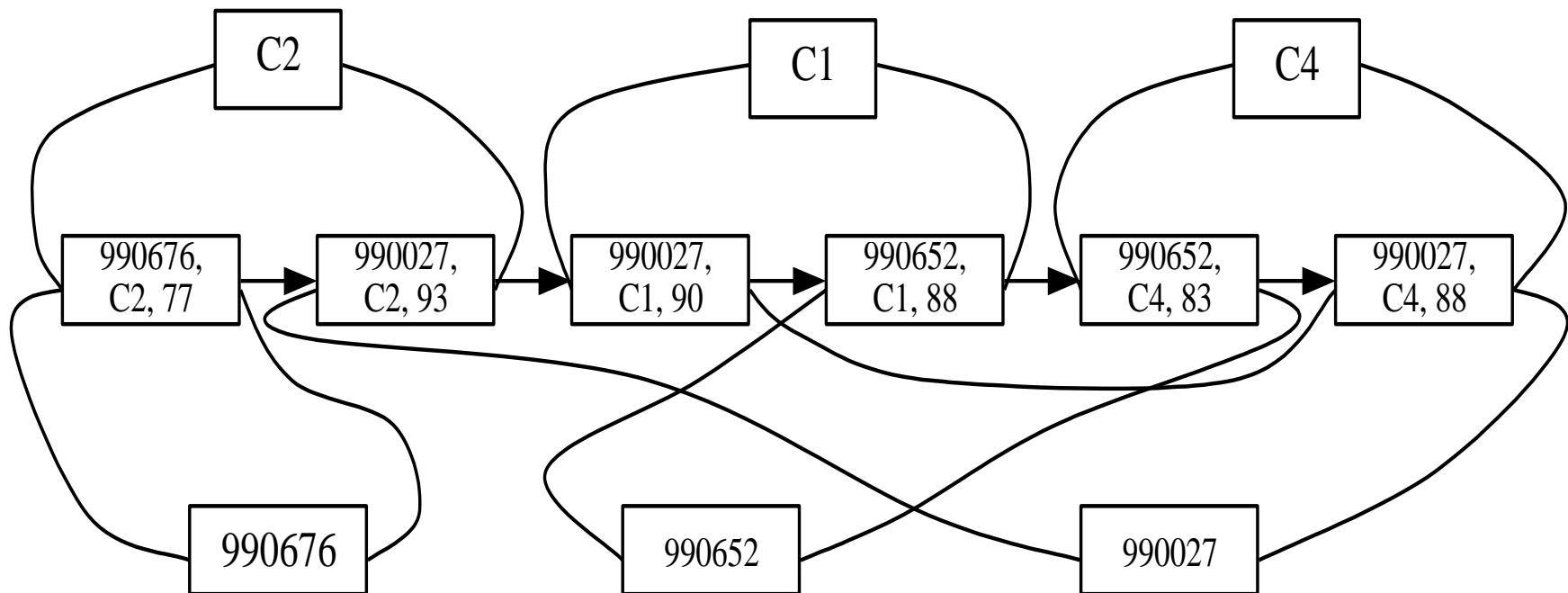


图1.24 学生/选课/课程的网状数据模型



(a)



(b)

网状数据模型的操纵与完整性约束（续）



- ❖ 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

三、网状数据模型的存储结构



❖ 关键

- 实现记录之间的联系

❖ 常用方法

- 单向链接
- 双向链接
- 环状链接
- 向首链接

网状数据模型的存储结构（续）

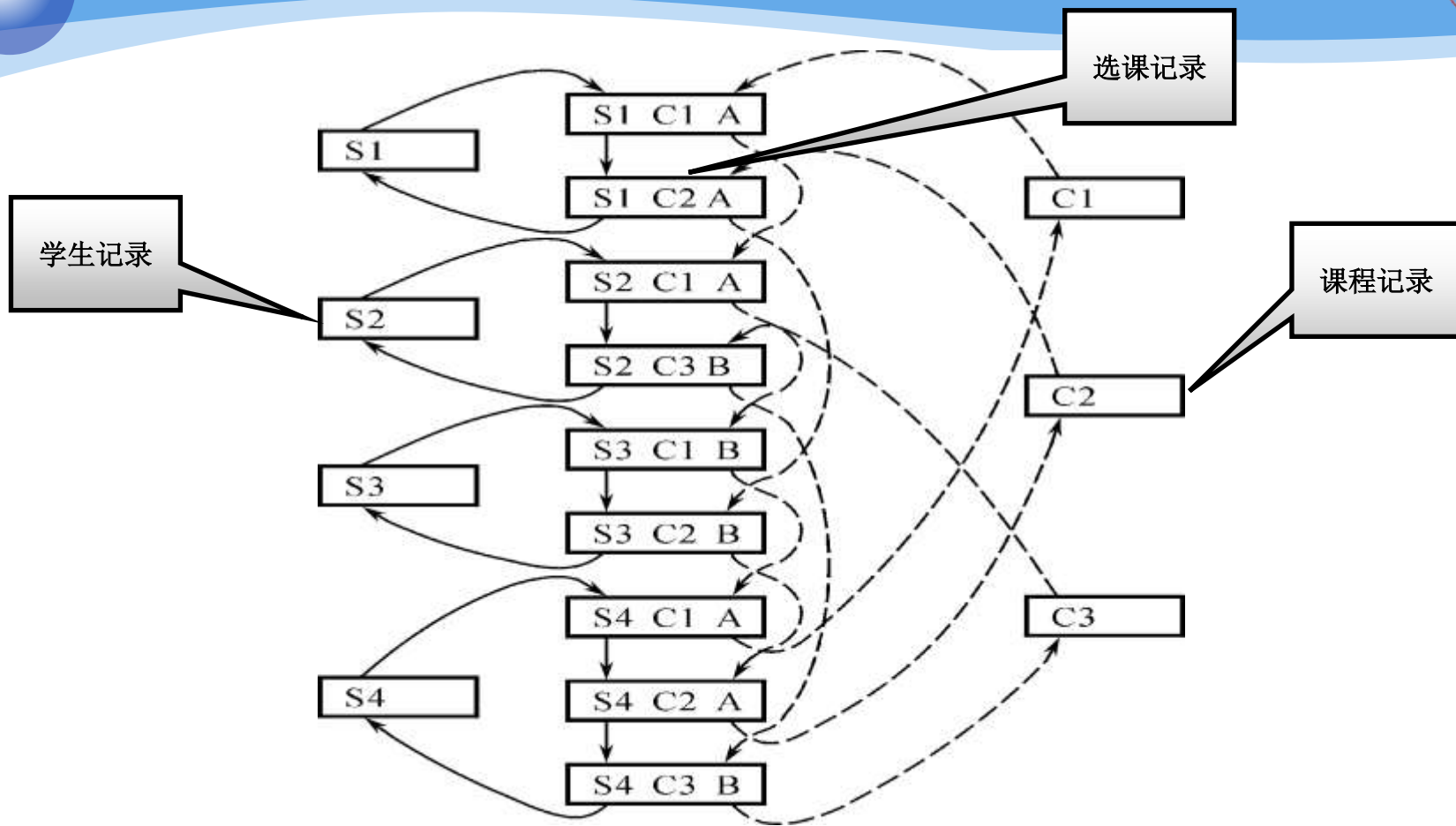


图1.25 学生/选课/课程的网状数据库实例

四、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

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

一、关系数据模型的数据结构



- ❖ 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

学 号	姓 名	年 龄	性 别	系 名	年 级
2005004	王小明	19	女	社会学	2005
2005006	黄大鹏	20	男	商品学	2005
2005008	张文斌	18	女	法律	2005
...

属性

元组

关系数据模型的数据结构（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

关系数据模型的数据结构（续）



- **主码 (Key)**

表中的某个属性组，它可以唯一确定一个元组。

- **域 (Domain)**

属性的取值范围。

- **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，年级）

关系数据模型的数据结构（续）



例1

学生、系、系与学生之间的一对多联系：

学生（学号，姓名，年龄，性别，系号，年级）

系（系号，系名，办公地点）

例2

系、系主任、系与系主任间的一对一联系

关系数据模型的数据结构（续）



例3

学生、课程、学生与课程之间的多对多联系：

学生（学号，姓名，年龄，性别，系号，年级）

课程（课程号，课程名，学分）

选修（学号，课程号，成绩）

关系数据模型的数据结构（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1.27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	津 贴	职 务	房 租	水 电	
86051	陈 平	讲 师	1305	1200	50	160	112	2283
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

图1.27 一个工资表(表中有表)实例

关系数据模型的数据结构（续）



表1.2 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）

二、关系数据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
- ❖ 数据操作是集合操作，操作对象和操作结果都是关系，即若干元组的集合
- ❖ 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”

关系数据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

三、关系数据模型的存储结构



- ❖ 实体及实体间的联系都用表来表示
- ❖ 表以文件形式存储
 - 有的DBMS一个表对应一个操作系统文件
 - 有的DBMS自己设计文件结构

四、关系数据模型的优缺点



❖ 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

关系数据模型的优缺点（续）



❖ 缺点

- 存取路径对用户透明导致查询效率往往不如非关系数据模型
- 为提高性能，必须对用户的查询请求进行优化增加了开发**DBMS**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 数据库系统结构

1.4 数据库系统的组成

1.5 小结

1.3 数据库系统结构



- ❖ 从数据库管理系统角度看，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（数据库系统外部的体系结构），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

学生记录型：

(学号，姓名，性别，系别，年龄，籍贯)

一个记录值：

(900201，李明，男，计算机，22，江苏)

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
- 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

- 2003年的一个学生数据库实例，包含：
 - 2003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
- 2002年度学生数据库模式对应的实例与2003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）

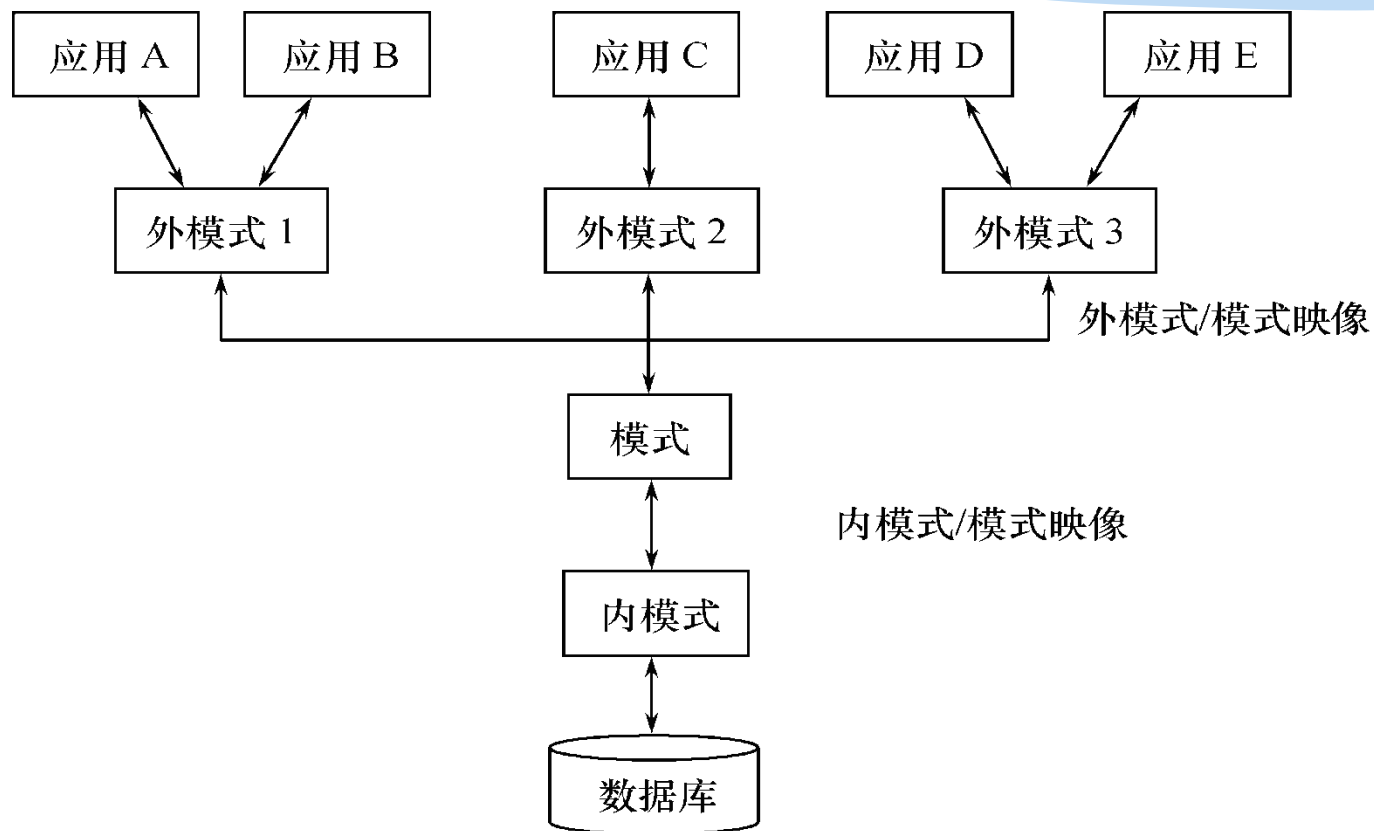


图1.28 数据库系统的三级模式结构

一、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，综合了所有用户的需求

❖ 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

二、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

三、内模式（Internal Schema）

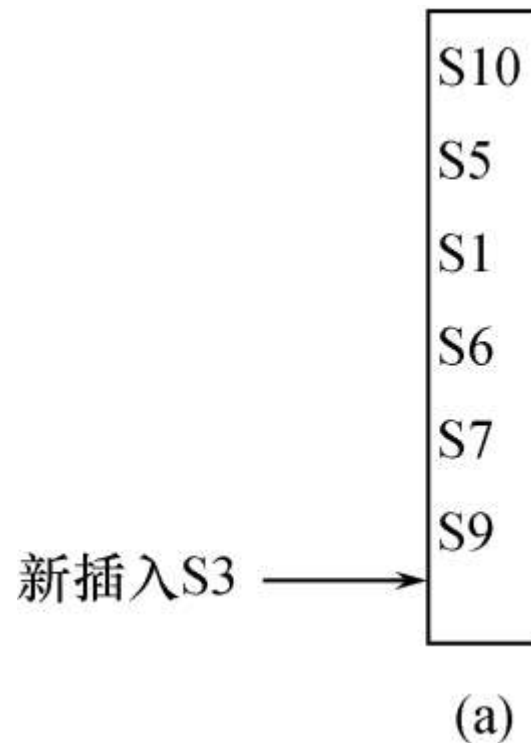


- ❖ 内模式（也称存储模式）
 - 是数据物理结构和存储方式的描述
 - 是数据在数据库内部的表示方式
 - 记录的存储方式（顺序存储，按照**B**树结构存储，按**hash**方法存储）
 - 索引的组织方式
 - 数据是否压缩存储
 - 数据是否加密
 - 数据存储记录结构的规定
- ❖ 一个数据库只有一个内模式

内模式（续）



- ❖ 例如学生记录，如果按堆存储，则插入一条新记录总是放在学生记录存储的最后，如右图所示



内模式（续）



- ❖ 如果按学号升序存储，则插入一条记录就要找到它应在的位置插入，如图 1.29（b）所示
- ❖ 如果按照学生年龄聚簇存放，假如新插入的S3是16岁，则应插入的位置如图 1.29（c）所示

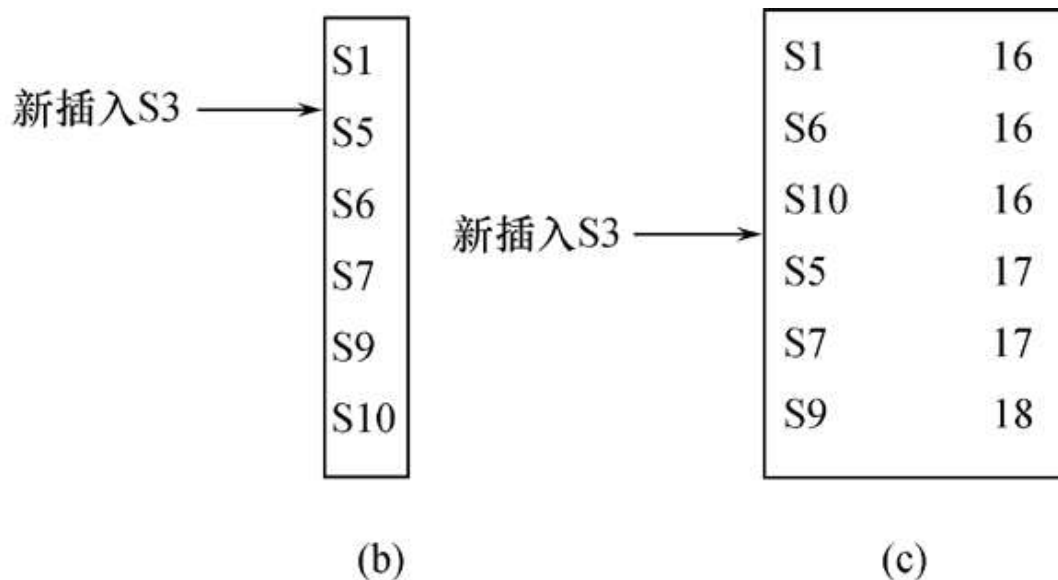


图1.29 记录不同的存储方式示意图

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

1.3.3 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
- ❖ 二级映像是在**DBMS**内部实现这三个抽象层次的关系和转换
 - 外模式 / 模式映像
 - 模式 / 内模式映像

一、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，数据库管理员修改有关的外模式 / 模式映像，使外模式保持不变
- 应用程序是依据数据的外模式编写的，从而应用程序不必修改，保证了数据与程序的逻辑独立性，简称数据的逻辑独立性。

二、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 / 内模式映象（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 / 内模式映象，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 / 内模式映象（续）



❖ 数据库模式

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

❖ 数据库的内模式

- 依赖于它的全局逻辑结构
- 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

❖ 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 / 内模式映象（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**DBMS**管理
 - 用户不必考虑存取路径等细节
 - 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

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）
- ❖ 应用系统
- ❖ 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

一、硬件平台及数据库



❖ 数据库系统对硬件资源的要求

(1) 足够大的内存

- 操作系统
- **DBMS**的核心模块
- 数据缓冲区
- 应用程序

硬件平台及数据库（续）



(2) 足够大的外存

- 磁盘或磁盘阵列
 - 数据库
- 光盘、磁带
 - 数据备份

(3) 较高的通道能力，提高数据传送率

二、软件



- ❖ DBMS
- ❖ 支持DBMS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DBMS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

三、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
- ❖ 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

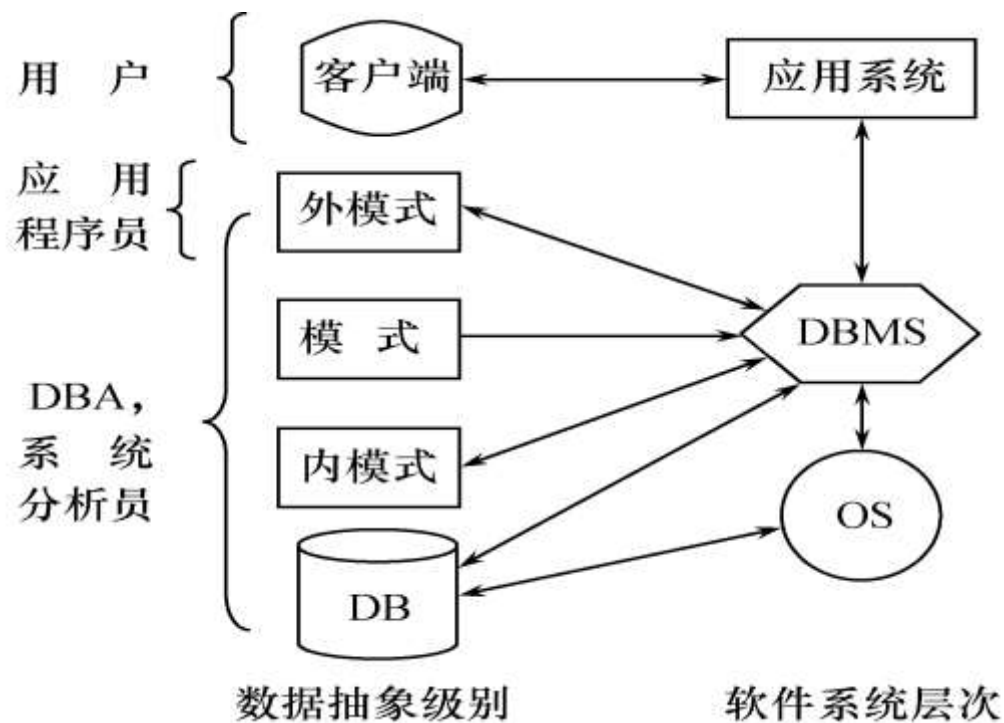


图1.30 各种人员的数据视图

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DBA协商，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API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

❖ 数据模型

- 数据模型的三要素
- 概念模型， E-R 模型
- 三种主要数据库模型

小结(续)



❖ 数据库系统的结构

- 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

下课了。。。



追求



休息一会儿。。。

