

# 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



# 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 and 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



# 数据抽象

## 现实世界

- 由实际事物组成，事物之间联系错综复杂
- 事物和事物特性

## 信息世界

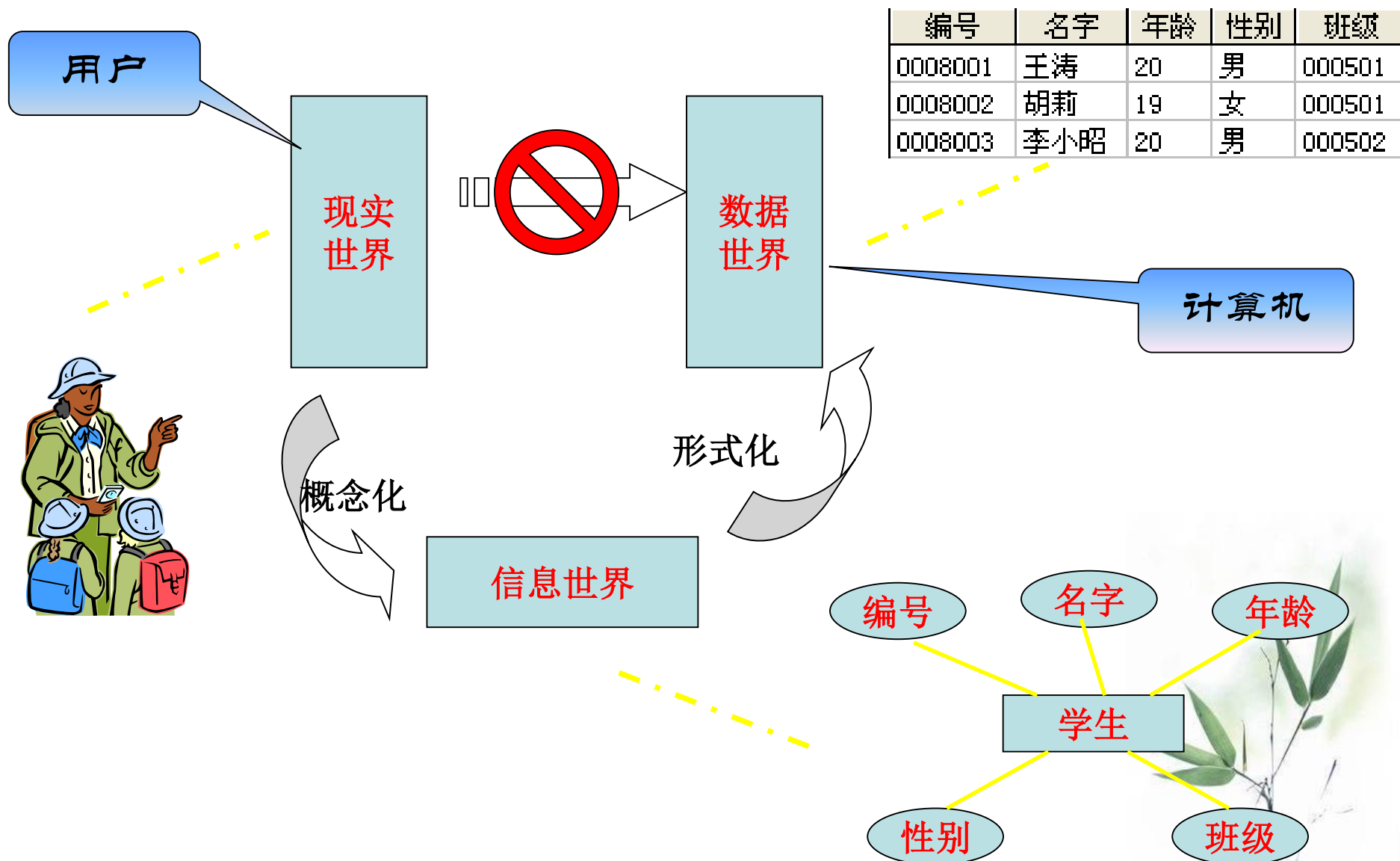
- 现实世界在人脑中的反映
- 事物和事物特性 $\longleftrightarrow$ 实体和实体属性

## 数据世界

- 信息世界数据化后的产物
- 实体和实体属性 $\longleftrightarrow$ 记录和数据项



# 现实世界的的数据化过程



- 如何看待信息

- 信息是现实世界中事物在人们头脑中的一种反映
- 信息可以准确地反映现实世界中事物（描述）
- 可以通过对现实进行抽象，形成信息（抽象）



“牛”



# •信息的取舍——抽象与具体化

- 现实世界中的事物包含了众多信息，哪些需要描述，哪些问题相关？

- “牛”
- “黑色”的牛
- “四条腿”的牛
- “公牛”而不是母牛
- “肥牛”而不是瘦牛
- 只有“1”头牛，而不是有几头牛
- ...



- 数据库设计往往因为忽视了信息之间联系的细致分析而造成设计失误
- 数据库设计能力的高低也体现在信息及其联系的正确分析上，体现在理解现实世界能力的高低



# 数据模型

- 不同范围的人对现实世界中事物的描述和抽象可能是不同的



- 现实的抽象与描述要遵循统一的数据模型：统一的概念与统一的表示方法



- 基本的抽象示例：“型”与“值”的抽象

计算机原理  
数据库系统原理  
离散数学  
高等数学  
英语  
政治经济学  
... ..



<课程>

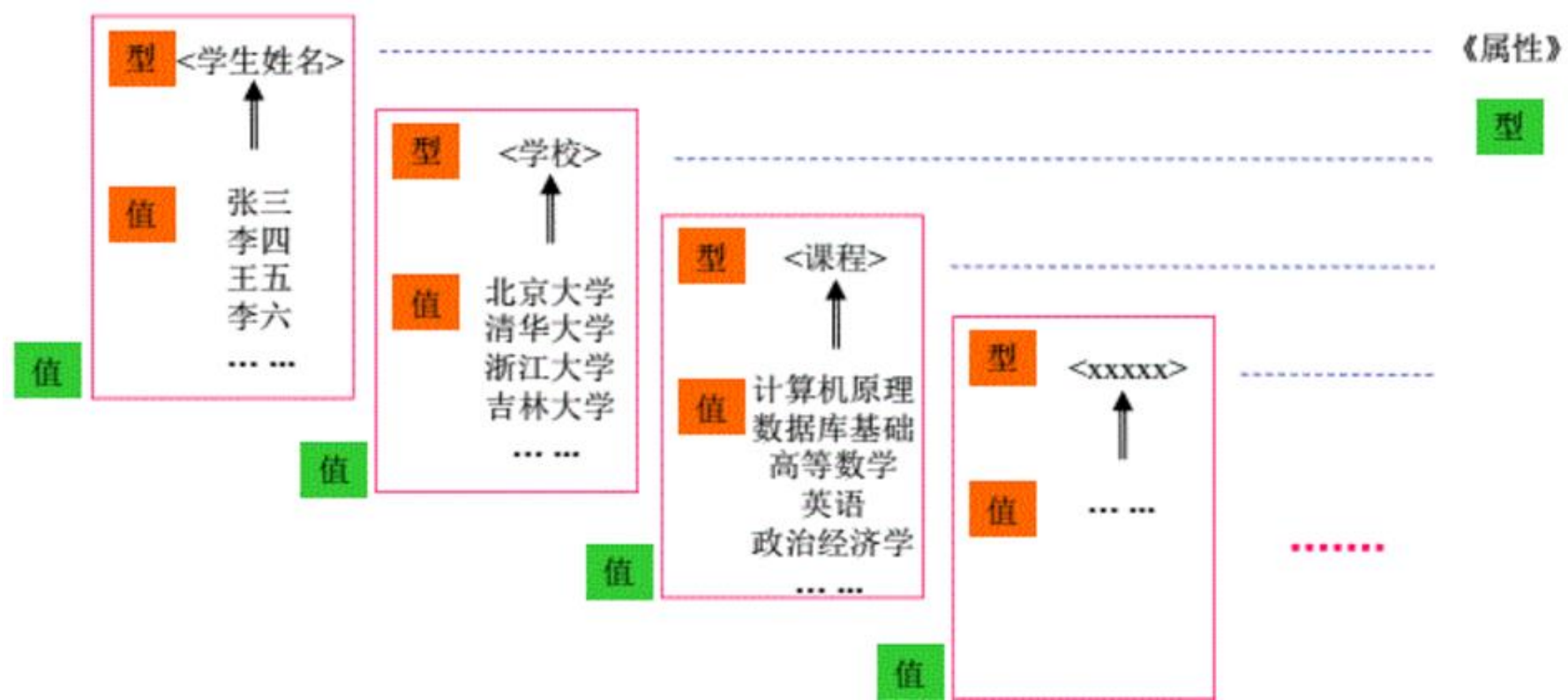
— 值

— 型





- 进一步“型”与“值”抽象
  - 将可无限扩展的内容，或内容暂无法枚举的情况，抽象为可有限描述的概念



# 两大类数据模型

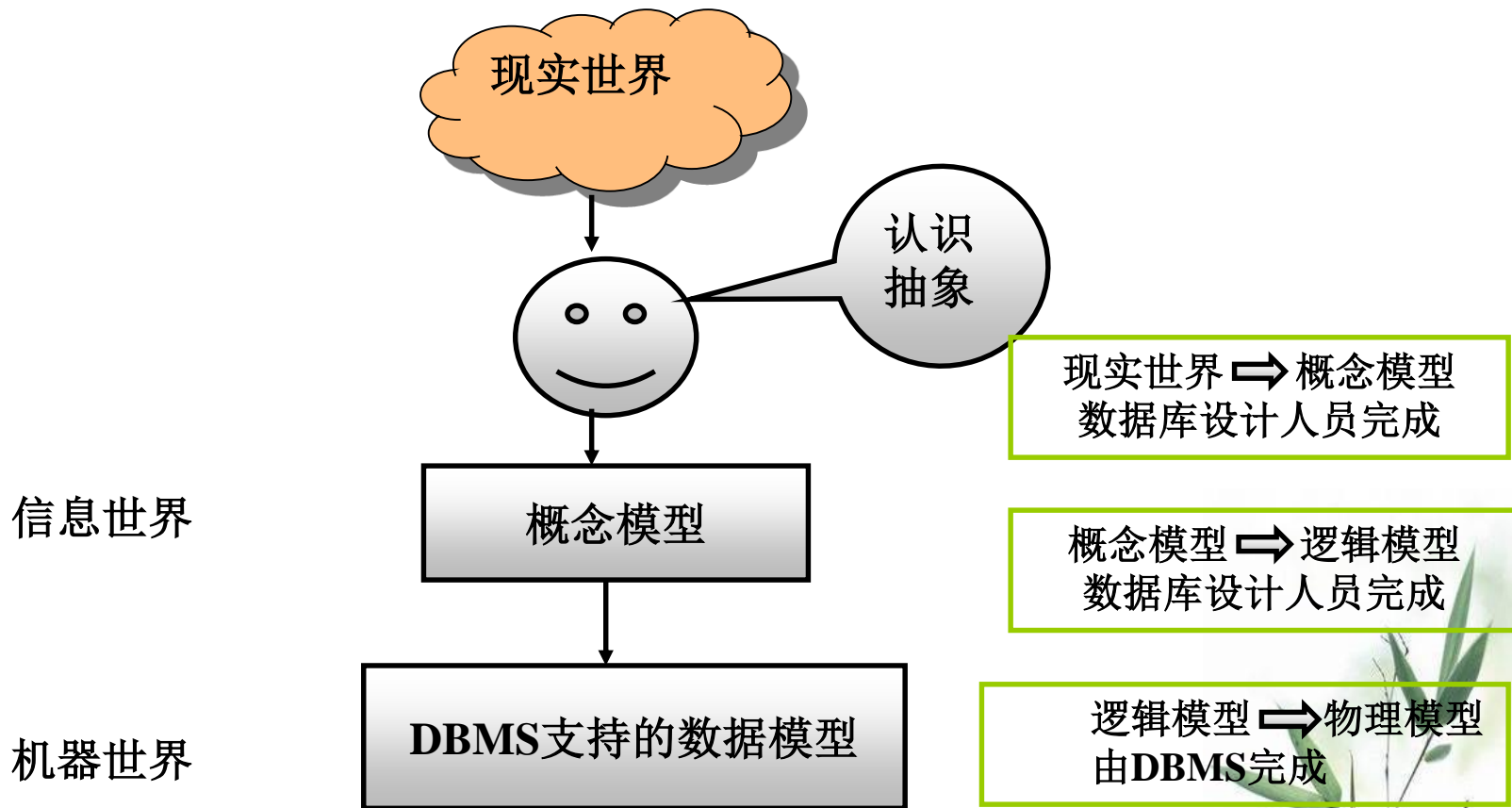
- 数据模型与概念模型

➤ 模型是一组相互关联且已严格定义的概念集合，是用于刻画或描述现实世界、信息世界或计算机世界

➤ 模型分不同的层次：描述计算机世界的称**数据模型**；描述信息世界或现实世界的称概念数据模型，简称**概念模型**



# 两大类数据模型 (续)



现实世界中客观对象的抽象过程

# 现实世界的周杰伦



**An Introduction to Database Systems**



# 信息世界的周杰伦（概念设计）

中文姓名：周杰伦  
英文姓名：Jay Chow  
生日：1979年1月18日  
星座：天蝎座  
血型：O型  
身高：173 厘米  
体重：60 公斤  
学历：淡江中学音乐  
音乐类型：R&B  
最爱物品：鸭舌帽



# 机器世界的周杰伦

- 逻辑设计

```
create table zjl  
( name char(10)  
  sex char(1)  
  birthday datetime  
.....)
```

- 物理设计



# 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

- 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



**T1**

学号	姓名	性别	系号
0101	张	男	X001
0102	李	女	X001
0203	赵	男	X003

**T2**

系号	系名
X001	计算机
X002	英语
X003	数学系

**T3**

学号	课号	成绩
0101	CS145	88
0101	CS148	90
0102	CS180	87
0203	CS145	78

**T4**

课号	课名
CS145	数据库
CS148	操作系统
CS180	数据结构

# 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
  - 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。

- 具体的人、事、物
  - 一个学生、一位运动员
  - 一个零件、一件艺术品
- 抽象的概念
  - 一个企业部门
  - 一门课程
  - 一个院系
- 联系
  - 学生的某个选课
  - 部门的一次订货
  - 老师与学院的工作关系





# 信息世界中的基本概念(续)

## (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。

学生实体：

学号	姓名	性别	出生年月	院系	入学时间
----	----	----	------	----	------

(94002268, 张山, 男, 197605, 计算机系, 1994)

## (3) 码 (Key)

唯一标识实体的属性集称为码。

学生实体：

学号	姓名	性别	出生年月	院系	入学时间
----	----	----	------	----	------

↓      ↓      X  
key   key?



# 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。

- 学号域：8位整数
- 姓名域：字符串集合
- 学生年龄：整数
- 性别：{男，女}
- 学位：{学士、硕士、博士}

## (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

- 同一实体型的实体具有共同的特征和性质
- 学生实体型：学生(学号，姓名，性别，出生年月，院系，入学时间)
- 课程实体型：课程(课程编号，课程名称，学时，教材名称)



# 信息世界中的基本概念(续)

## (6) 实体集 (Entity Set)

同一类型实体的集合称为实体集

- 05级计算一班全体同学
- 数学专业所有本科课程



# 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



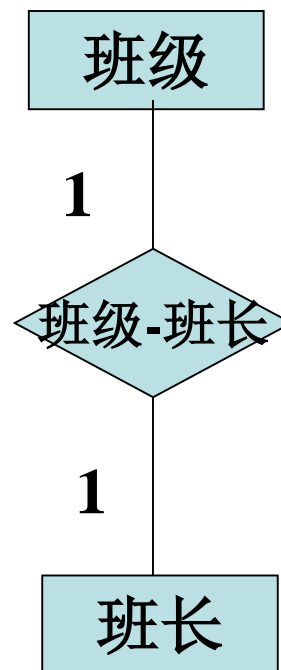
# 两个实体型之间的联系

- 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



1:1联系

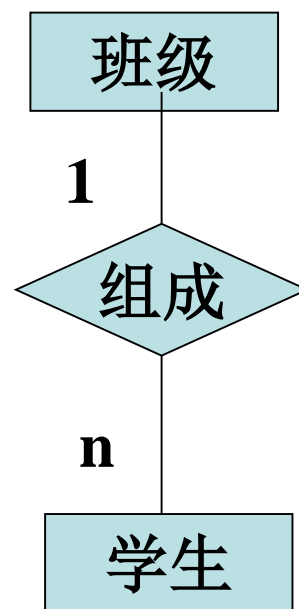


# 两个实体型之间的联系

- 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



1:n联系



# 两个实体型之间的联系

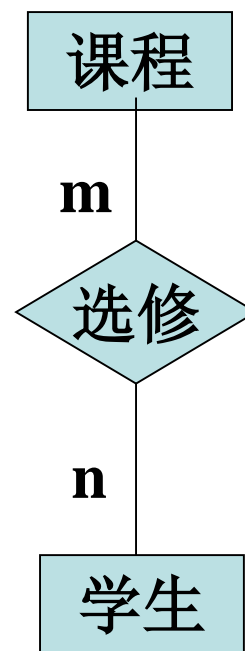
- 多对多联系 (m:n)

- 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



m:n联系



# 两个以上实体型之间的联系

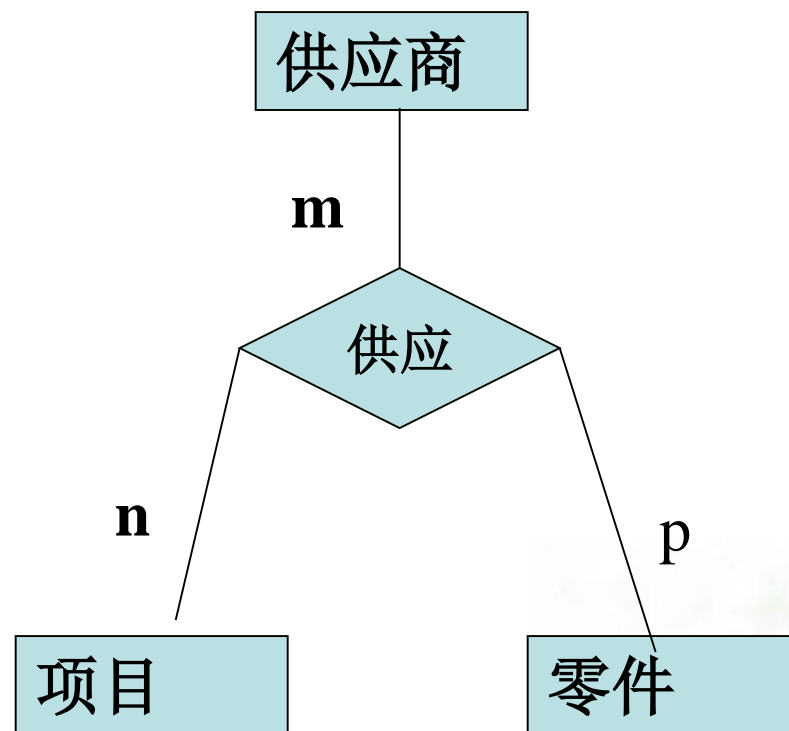
- 两个以上实体型间的多对多联系
  - 实例

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



# 两个以上实体型之间的联系

- 两个以上实体型间的1对多联系

- 实例

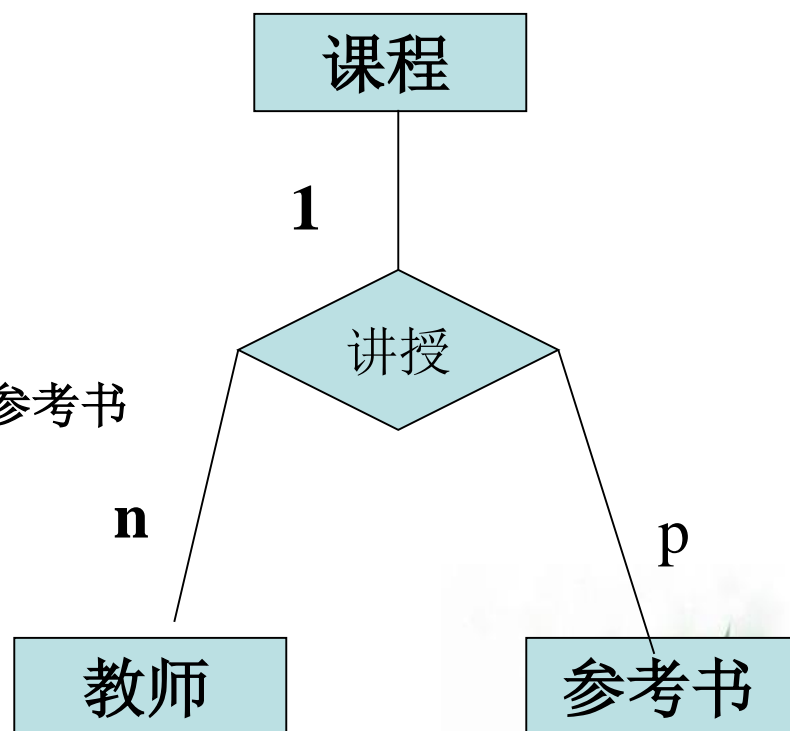
课程、教师与参考书三个实体型

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

- 两个以上实体型间的1对1联系（思考题）



两个以上实体型间1:n联系

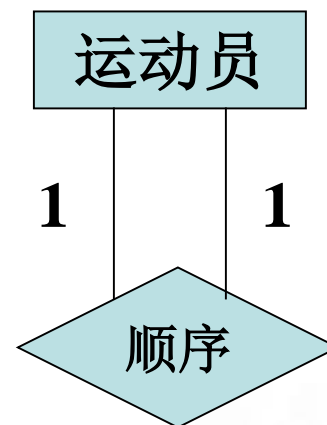
# 单个实体型内的联系

- 一对一联系

- 实例

- ❖ 多对多联系

- ❖ 一对多联系



单个实体型内部  
1:1联系



# 单个实体型内的联系

- 一对多联系

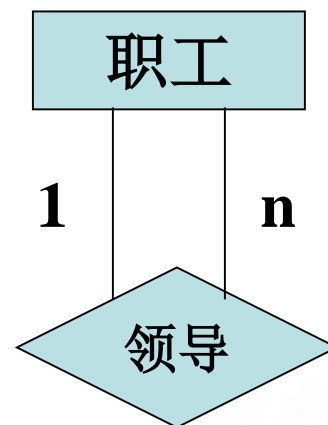
- 实例

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

- ❖ 多对多联系

- ❖ 一对一联系

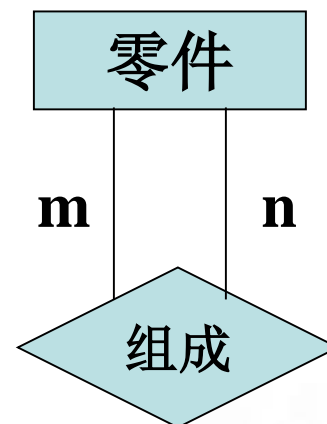
# 单个实体型内的联系

- 多对多联系

- 实例

- ❖ 一对多联系

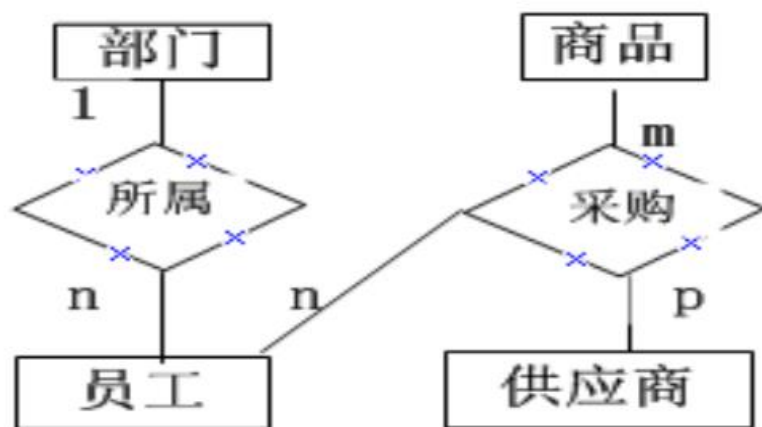
- ❖ 一对一联系



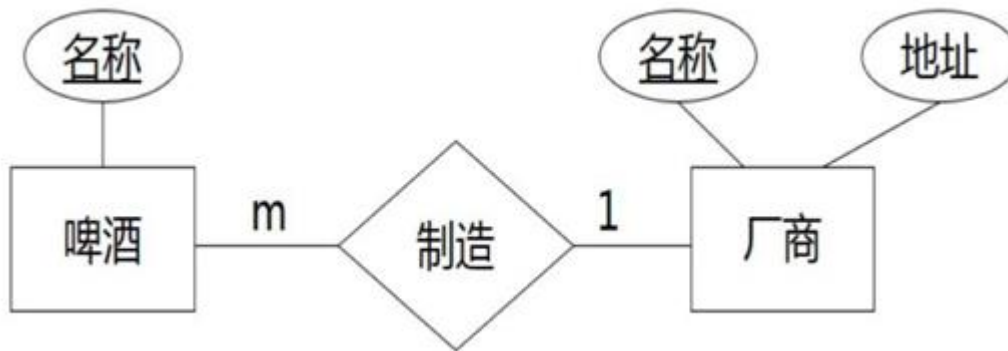
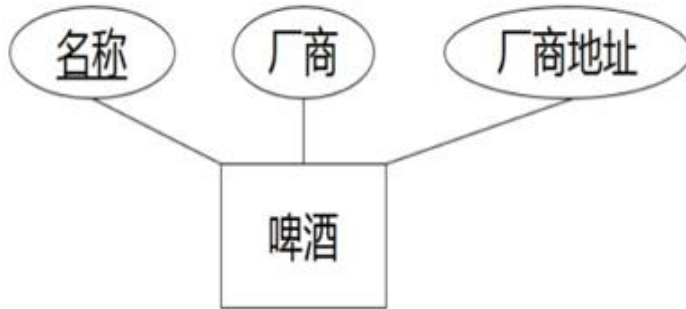
单个实体型内部  
m:n联系



- 例如：超市销售管理系统中的采购商品过程中，采购人员（员工）隶属于采购部（部门），部门与员工之间是1:n关系，采购人员可以跟不同供应商采购多件不同商品。由于一个员工负责可以跟多个供应商采购商品采多种商品，一个供应商供应多个员工多种商品，所以员工与供应商之间是多对多关系，商品与供应商之间是多对多关系，员工与商品之间也是对多对关系。用E-R图表示。



# 分析



# 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)

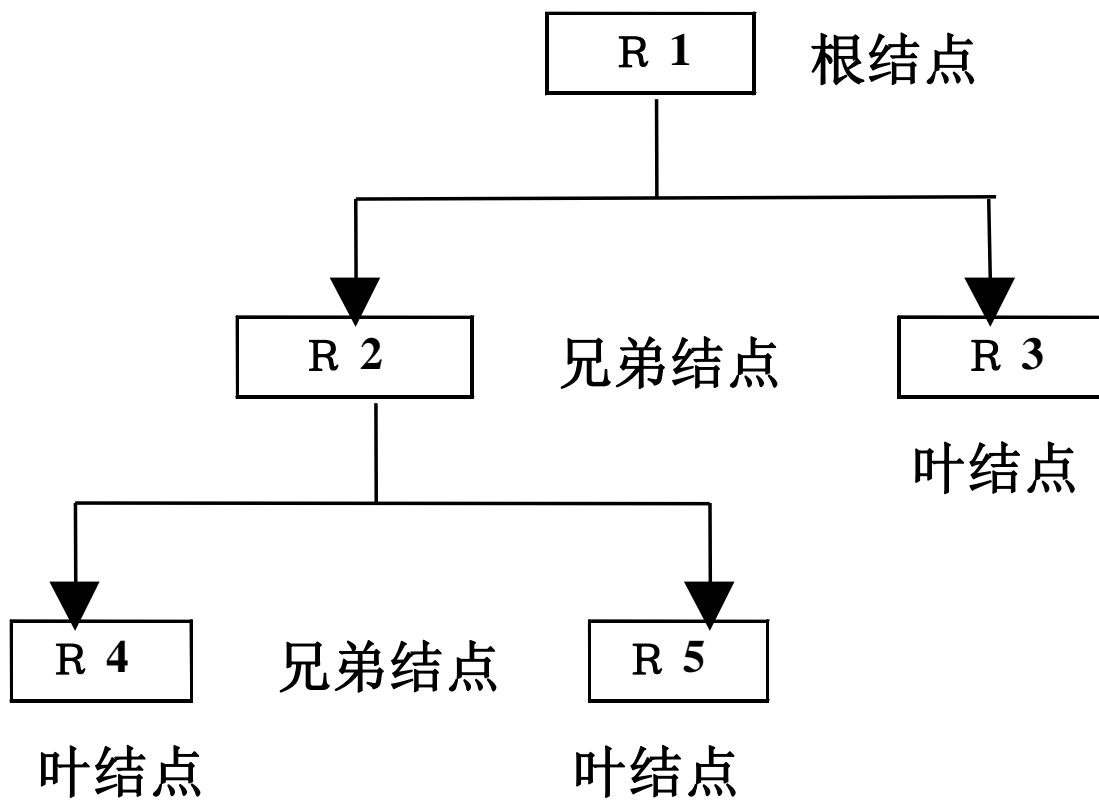


图 一个层次模型的示例



# 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
  - 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

# 层次数据模型的数据结构(续)

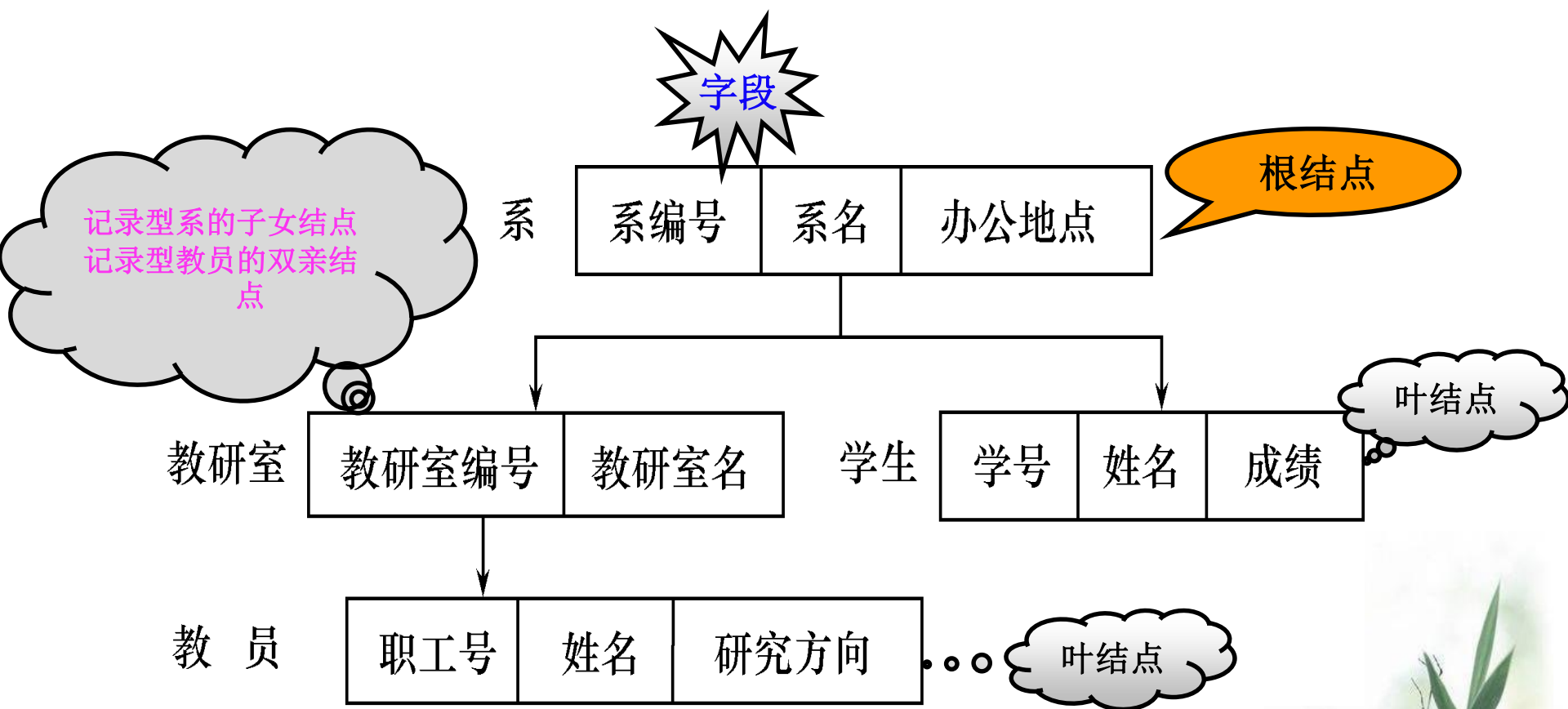


图 教员学生层次数据库模型

# 层次数据模型的数据结构(续)

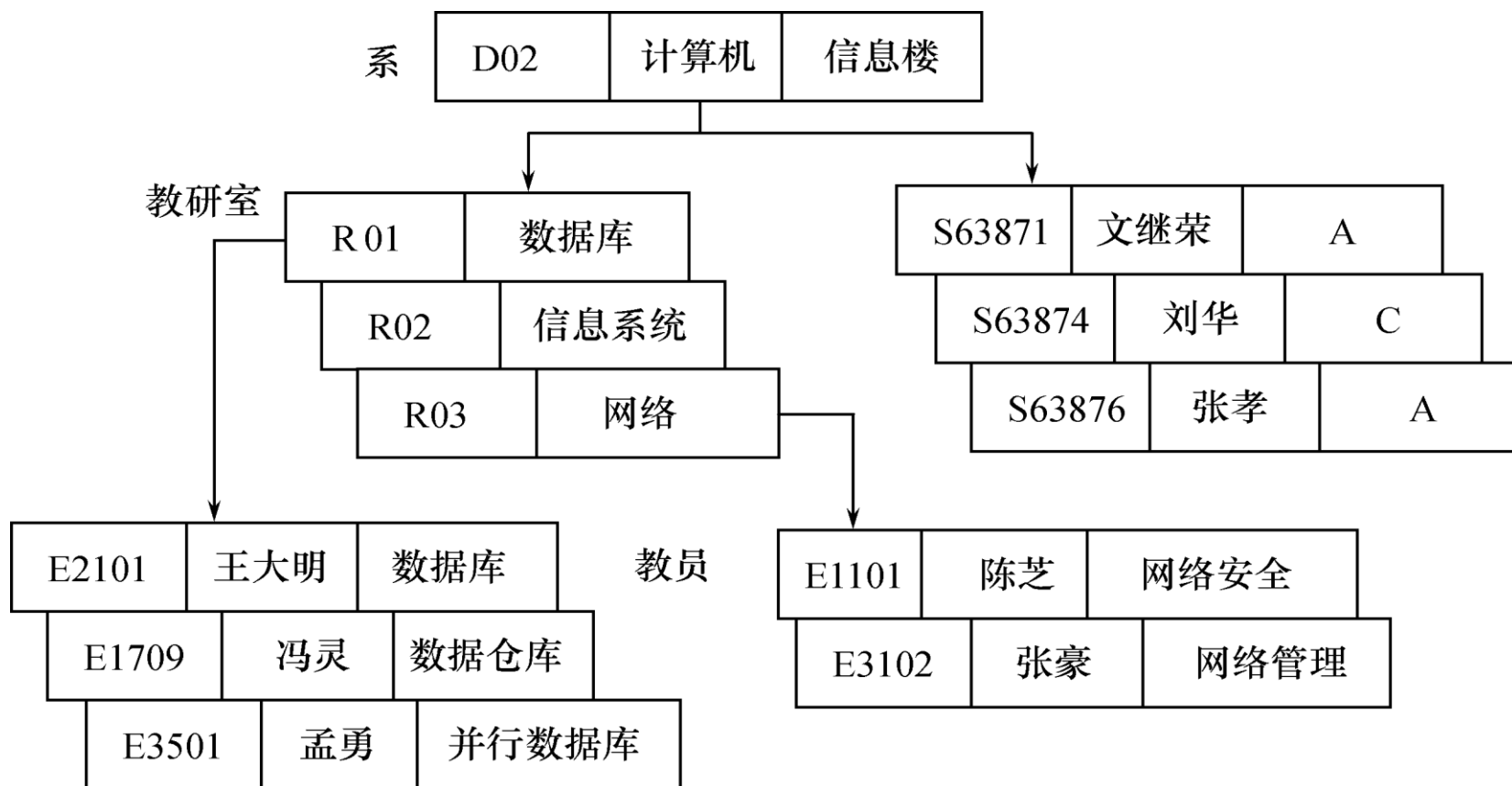


图 教员学生层次数据库的一个值

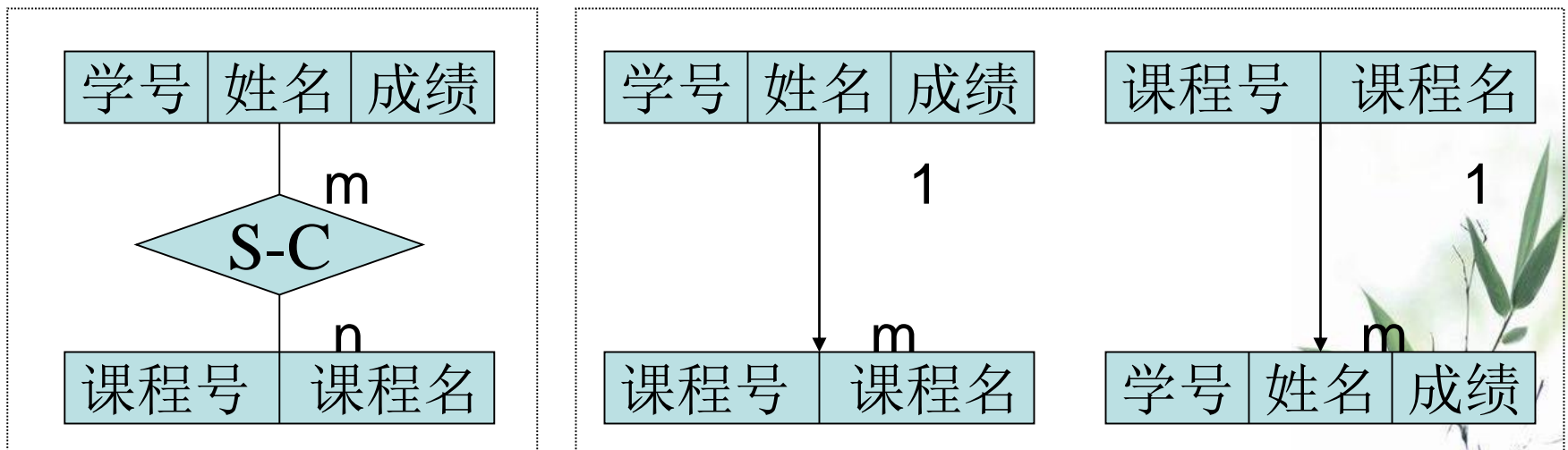
## 二、多对多联系在层次模型中的表示

- 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
    - 虚拟结点法



# 多对多联系在层次模型中的表示

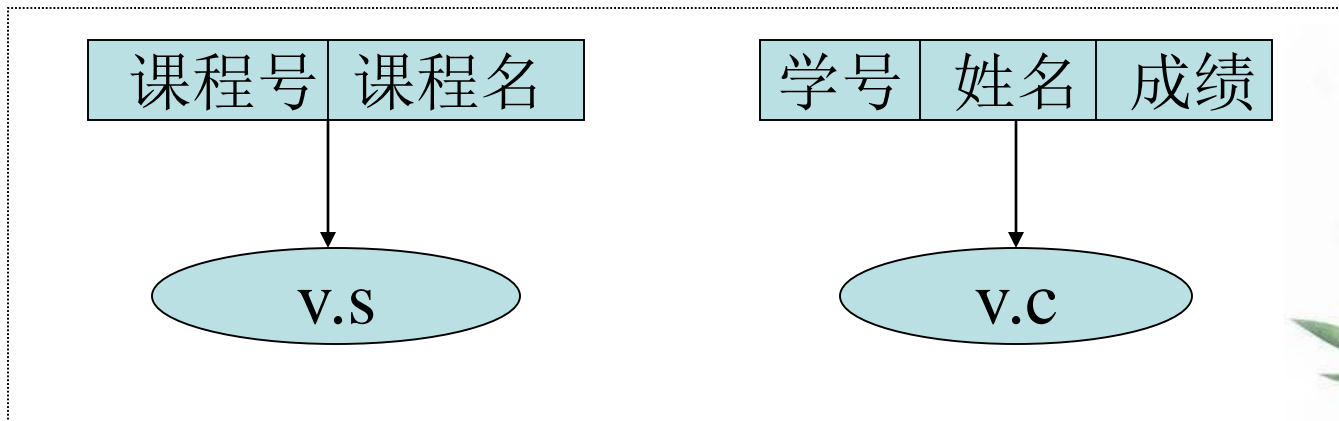
- 冗余结点法：两个实体的多对多联系通过增设两个冗余结点将其转换成两个一对多联系。
  - 优点：结构清晰，允许结点改变存储位置
  - 缺点：需要额外占用存储空间，有潜在的不一致性





# 多对多联系在层次模型中的表示

- 虚拟结点法：将冗余结点转换为虚拟结点，虚拟结点是一个指引元，指向所替代的结点。
  - 优点：减少对存储空间的浪费,避免产生潜在的不一致性
  - 缺点：结点改变存储位置可能引起虚拟结点中指针的修改



# 三、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



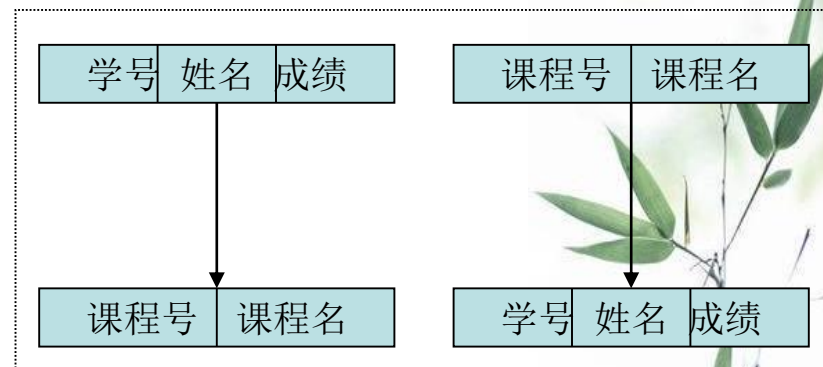
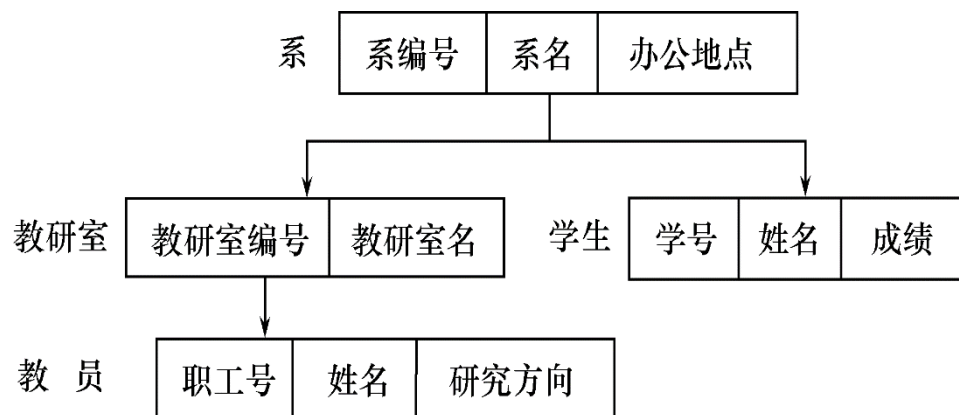
# 层次模型的数据操纵与完整性约束（续）

- 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：若新调入一名教师，但尚未分配到某个教研室，这时就不能将新教员插入到数据库中。

如果删除双亲结点值，则相应的子女结点值也被同时删除：若删除网络教研室，则该教研室所有老师的数据将全部丢失。

更新操作时，应更新所有相应记录，以保证数据的一致性：如果一个学生要改姓名，则两处学生记录值的姓名字段都需修改。



## 四、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



- 层次数据库后台的数据结构是“树”，对于树的操作包括树的遍历、节点的插入、节点的删除、节点的修改、子树的获取等等，必须对数据结构的基本原理熟练掌握才能够写出相应的操作代码。从这个角度来说，相关的命令更像是编写程序，称为趋于程序化。



# 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

- 在现实世界中事物之间的联系更多的是非层次关系的，用层次模型表示非树形结构不直接，网状模型则可以克服这一弊病。
- 网状数据模型的典型代表是DBTG系统。这是20世纪70年代数据系统语言研究会CODASYL（Conference On Data System Language）下属的数据库任务组（Data Base Task Group，简称DBTG）提出的一个系统方案。
- DBTG系统虽然不是实际的软件系统，但是它提出的基本概念、方法和技术具有普遍意义。它对于网状数据库系统的研制和发展起了重大的影响。后来不少的系统都采用DBTG模型或者简化的DBTG模型。例如，Cullient Software公司的IDMS、UniVac公司的DMS1100、Honeywell公司的IDS/2、HP公司的IMAGE等。

# 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



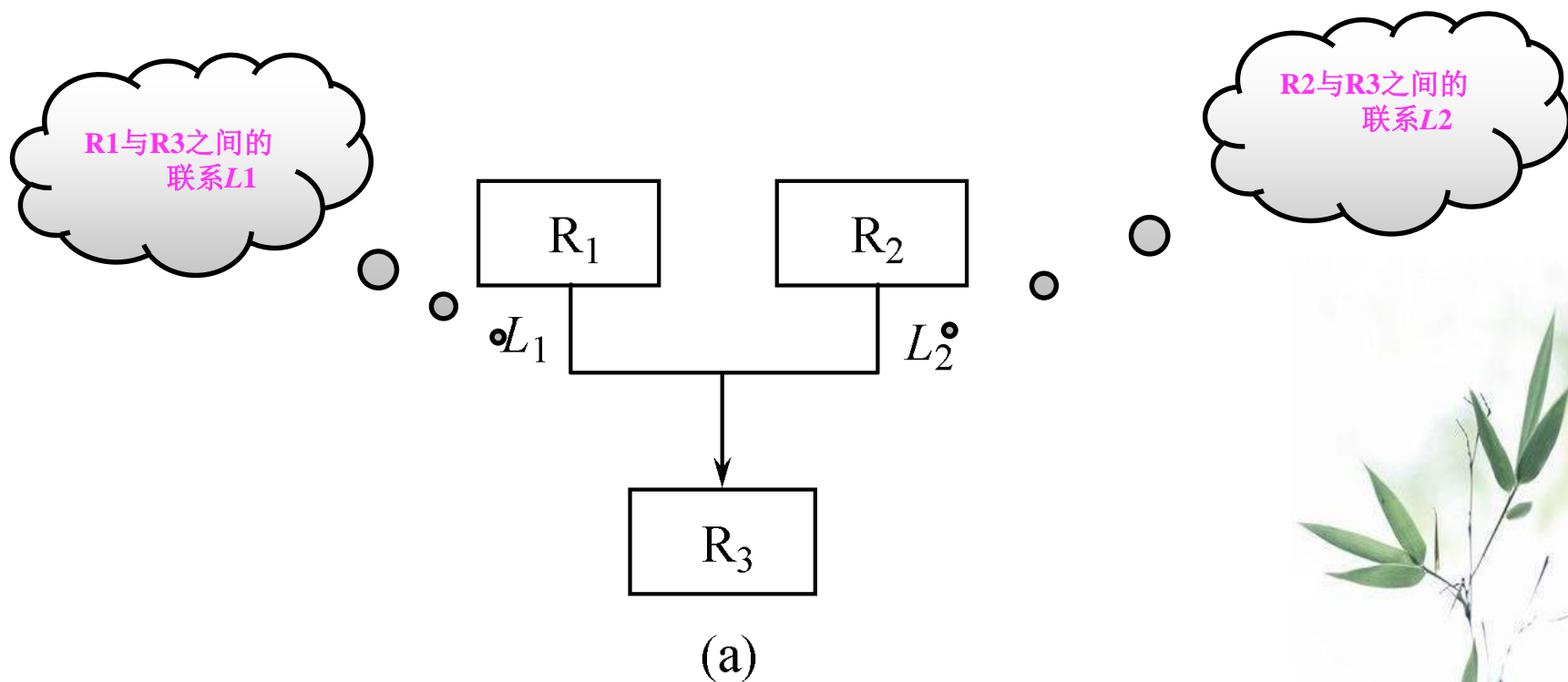
# 网状数据模型的数据结构（续）

- 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

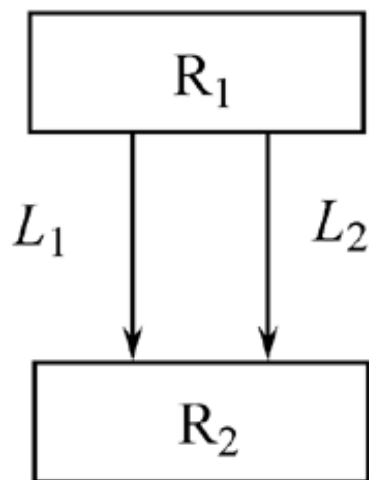


# 网状数据模型的数据结构（续）

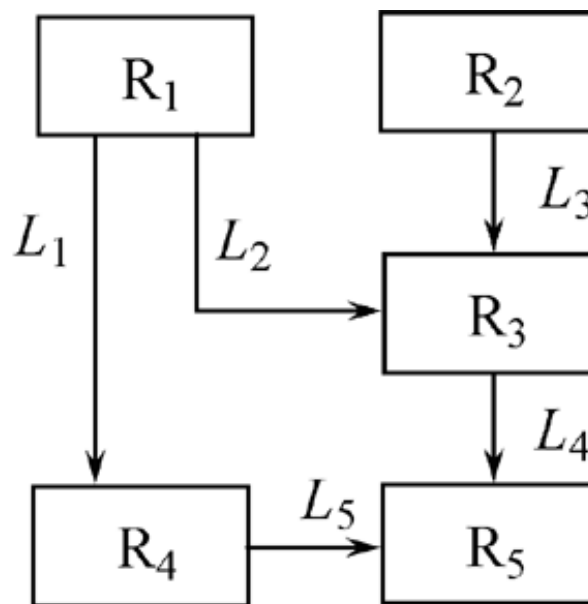
- ❖ 网状模型中子女结点与双亲结点的联系可以不唯一  
要为每个联系命名，并指出与该联系有关的双亲记录和子女记录



# 网状数据模型的数据结构（续）



(b)

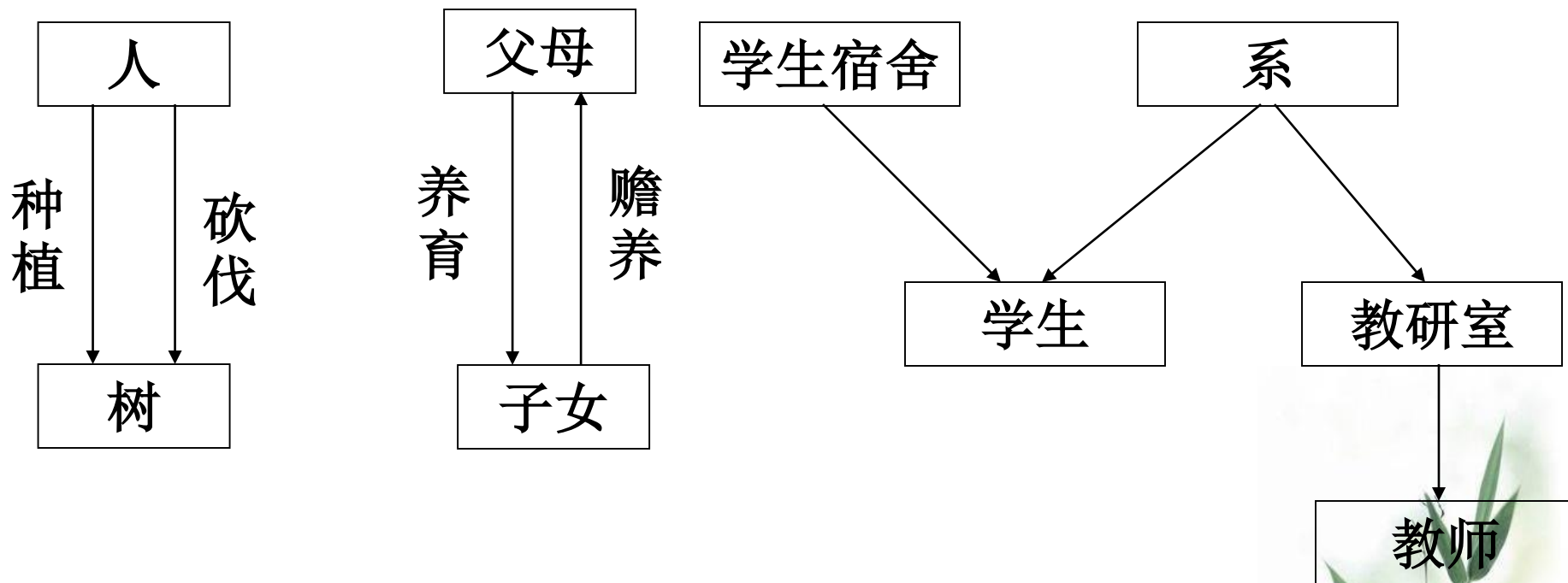


(c)

网状模型的例子



# 网状数据模型的数据结构



# 网状数据模型的数据结构（续）

实际的商品化网状数据库系统对网状数据结构都有不同的限制，例如HP公司的IMAGE3000数据库管理系统是一个网状数据库系统，但是限制网状结构的层次只有两层，是一个简化了的网状结构。

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,表示某个学生选修某一门课程及其成绩

- 学号
- 课程号
- 成绩



# 网状数据模型的数据结构（续）

每个学生可以选修多门课程，对学生记录中的一个值，选课记录中可以有多多个值与之联系，而选课记录中的一个值，只能与学生记录中的一个值联系。学生与选课之间的联系是一对多的联系，联系名为S-SC。

同样，课程与选课之间的联系是一对多的联系，联系名为C-SC。

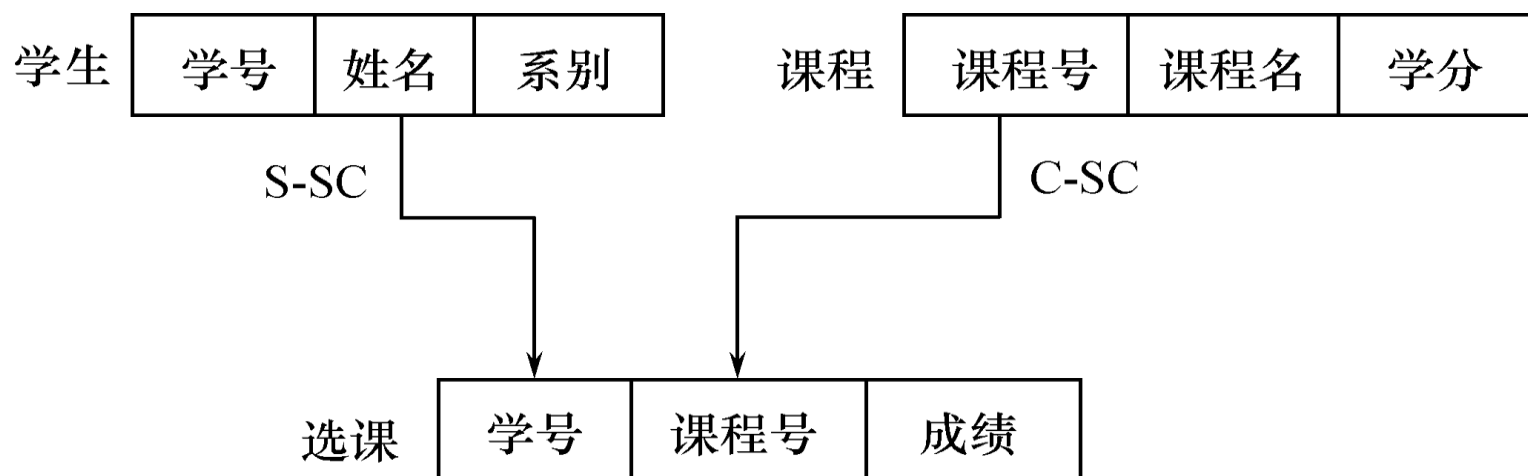


图 学生/选课/课程的网状数据模型





# 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
- 由于记录之间的联系是通过存取路径实现的，应用程序在访问数据库时必须选择适当的存取路径，即用户需要了解系统结构的细节，加重了编写应用程序的负担。



## 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
- 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，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）

- 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
ℵ	ℵ	ℵ	ℵ	ℵ	ℵ	ℵ	ℵ	ℵ

图 一个工资表(表中有表)实例

# 关系数据模型的数据结构（续）

## 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

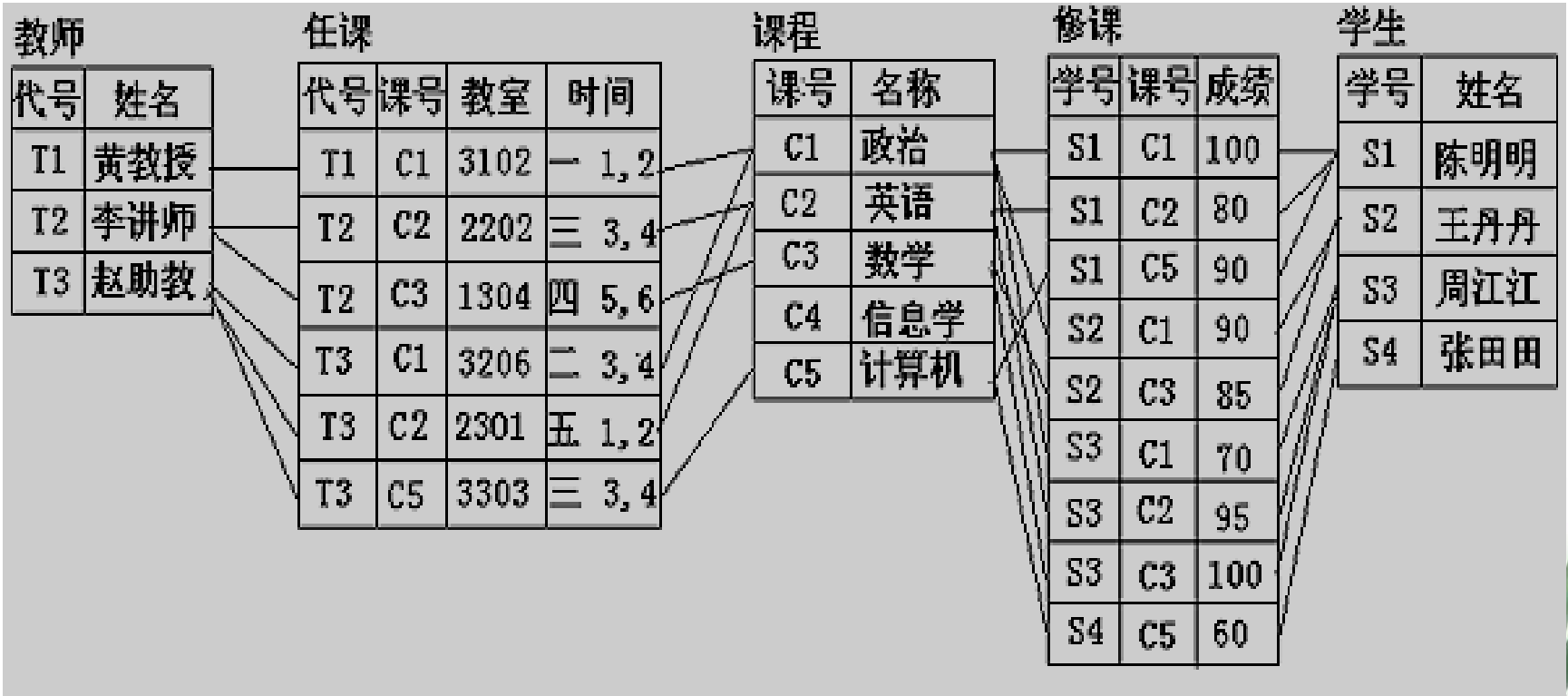
- 数据操作是集合操作，操作对象和操作结果都是关系
  - 查询
  - 插入
  - 删除
  - 更新
- 数据操作是集合操作，操作对象和操作结果都是关系，即若干元组的集合
- 存取路径对用户隐蔽，用户只要指出“干什么”，不必详细说明“怎么干”



# 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



# 常用数据模型的比较

	层次模型	网状模型	关系模型	面向对象模型
创始	1968 IBM IMS系统	1969 CODASYL DBTG报告	1970 E.F.Codd 提出	20世纪80年代
数据结构	复杂 (树结构)	复杂 (有向图结构)	简单 (二维表)	复杂 (嵌套, 递归)
数据联系	通过指针	通过指针	通过表间的公 共属性	面向对象标识
查询语言	过程性语言	过程性语言	非过程性语言	面向对象语言
典型产品	IMS	IDS/II, IMAGE/ 3000, IDMS	Oracle, Sybase, DB2, SQL Server	ONTOS DB
盛行期	20世纪70年 代	20世纪70年代 到80年代中期	20世纪80年代 到现在	20世纪90年代到现 在