

# 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  
2015**

# 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。

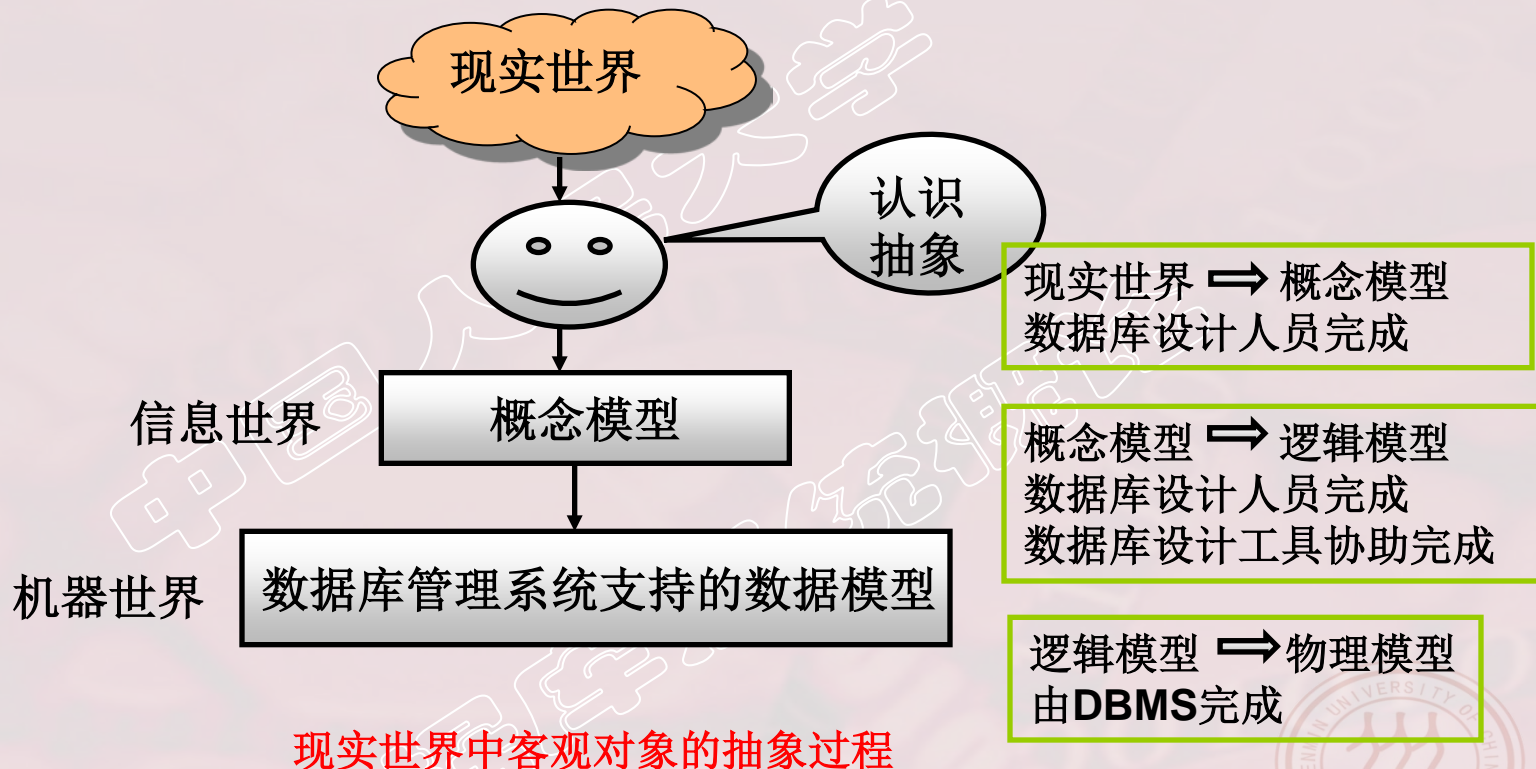
- 物理模型是对数据最底层的抽象

描述数据在系统内（磁盘上）的表示方式和存取方法。





# 两类数据模型



# 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



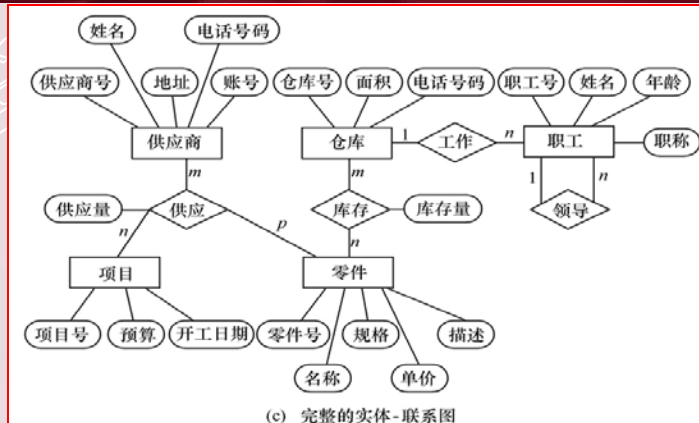
## 1.2.2 概念模型

### ❖ 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



例：工厂物质管理的概念模型



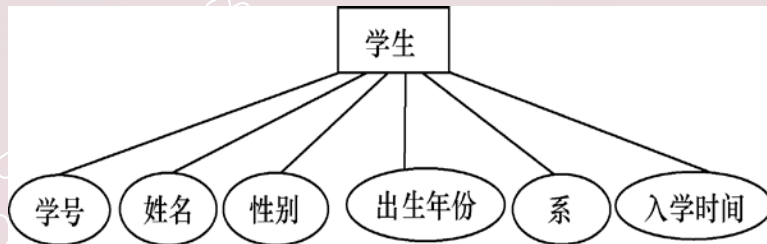


# 1. 信息世界中的基本概念

## (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。

## (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）

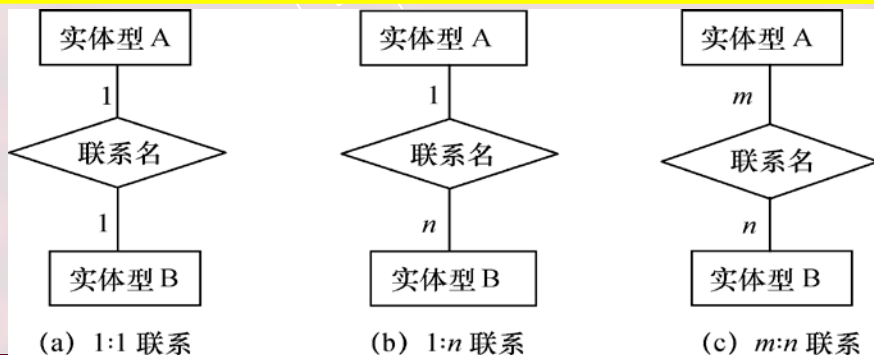
## （6）联系（Relationship）

- 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：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（1:1）、一对多（1:n）和多对多（m:n）等多种类型



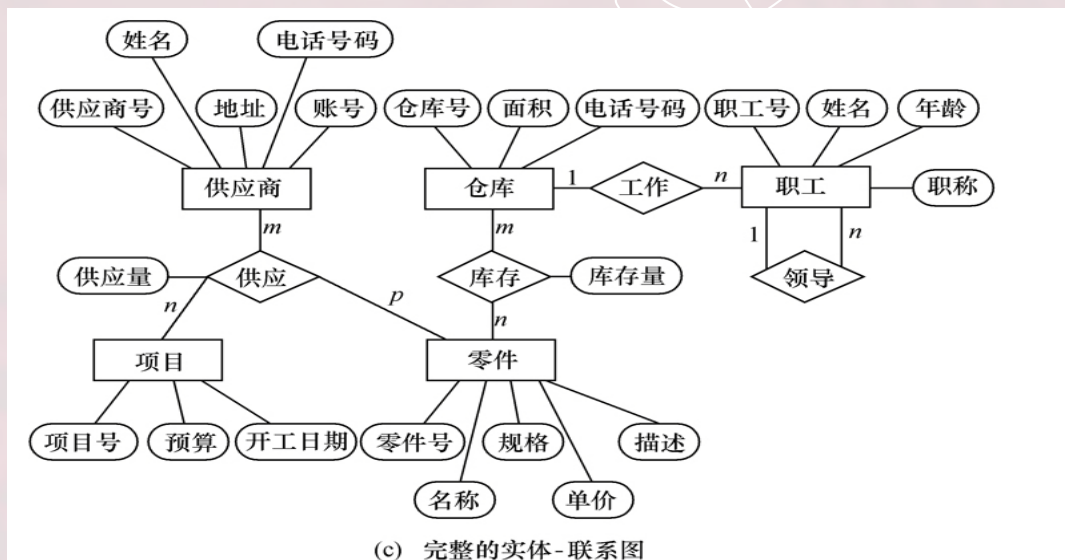
# 实体-联系方法

❖ 概念模型的一种表示方法:

❖ **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

## ❖ 数据模型是严格定义的一组概念的集合

精确地描述了系统的静态特性、动态特性和完整性约束条件 (Integrity Constraints)。

## ❖ 数据模型由三部分组成

- 1 数据结构--描述系统的静态特性
- 2 数据操作--描述系统的动态特性
- 3 完整性约束





# 1. 数据结构

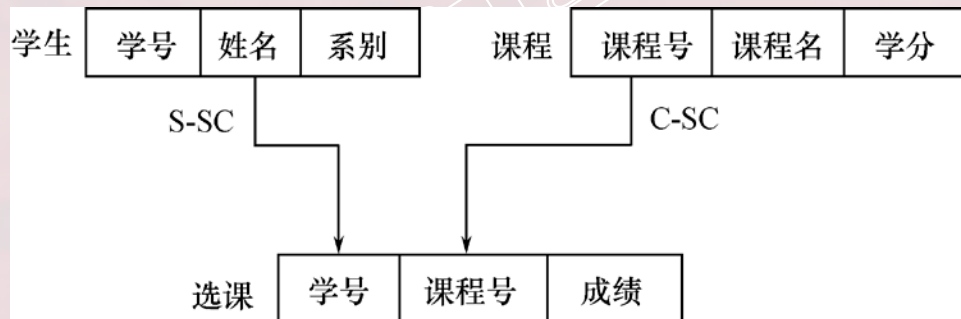
## ❖ 刻画数据模型性质的重要方面

**数据结构**的类型来命名数据模型

层次结构-层次模型、网状结构--网状模型、关系结构—关系模型

## ❖ 描述数据库的组成对象--对象的类型、内容、性质

## ❖ 描述对象之间的联系



记录:

学生--由学号、姓名所在的专业系名等组成

**SET TYPE:**

**S-SC**--学生记录和学生选课记录之间的联系

一个网状数据模型实示例

# 2. 数据操作

## ❖ 数据操作

- 对数据库中各种对象的实例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）

## ❖ 数据操作语言

- 定义数据操作的确切含义、符号、优先级别
- 实现数据操作的语言
  - 查询语言——Query Language
  - 更新语言——Insert、Delete、Update



# 3. 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

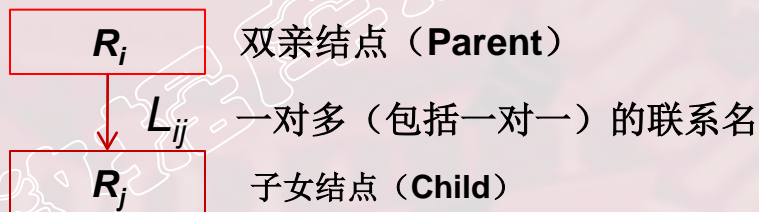




## 1.2.4 常用的数据模型

- ❖ 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 (Semi-structure Data Model) —如XML
  - ❖ 非结构化数据模型、图模型 .....
- 格式化模型
- 对象模型

格式化模型中数据结构的单位：基本层次联系



# 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

❖ 层次模型用树形结构来表示各类实体以及实体间的联系

❖ 表示方法

**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

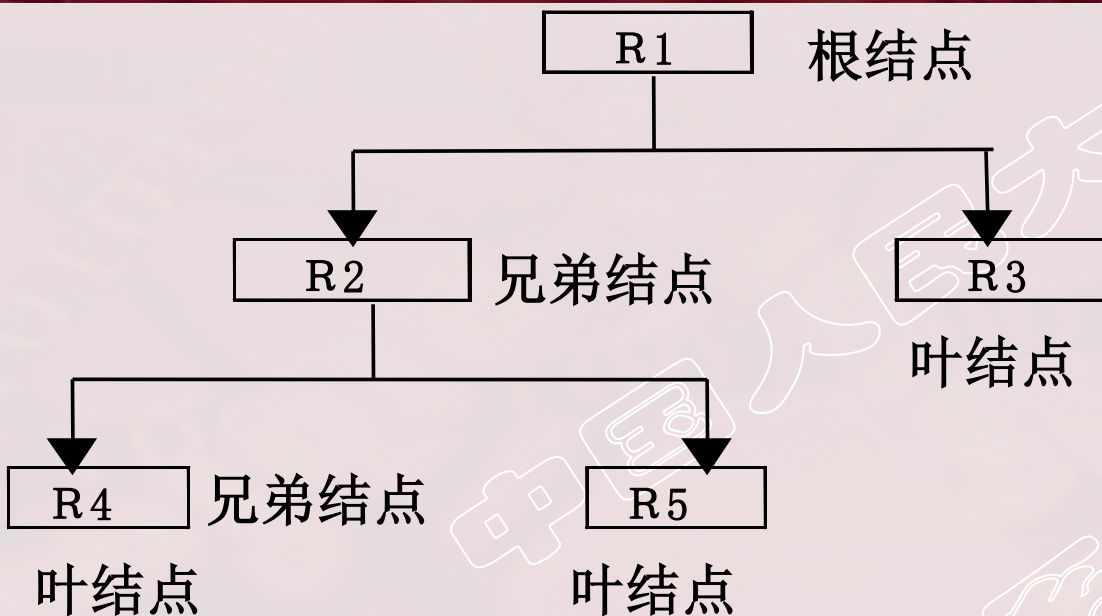
❖ 层次模型的定义

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



# 1. 层次模型的数据结构



## 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
- 任何记录值只有按其路径查看
- 没有一个子女记录值能够脱离双亲记录值而独立存在

图1.9 一个层次模型的示例



# 层次模型的数据结构（续）

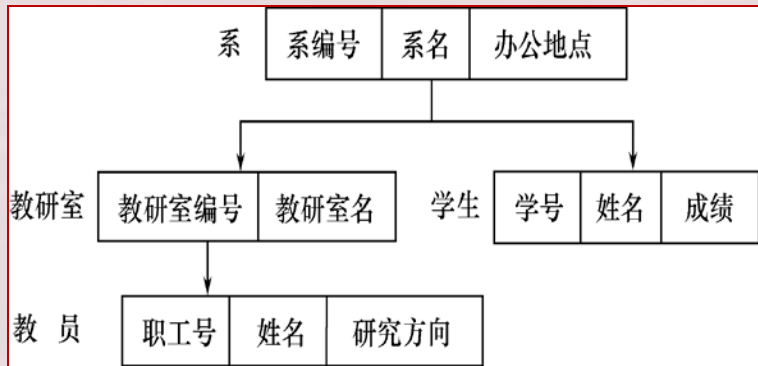


图1.10 教员学生层次数据库模型

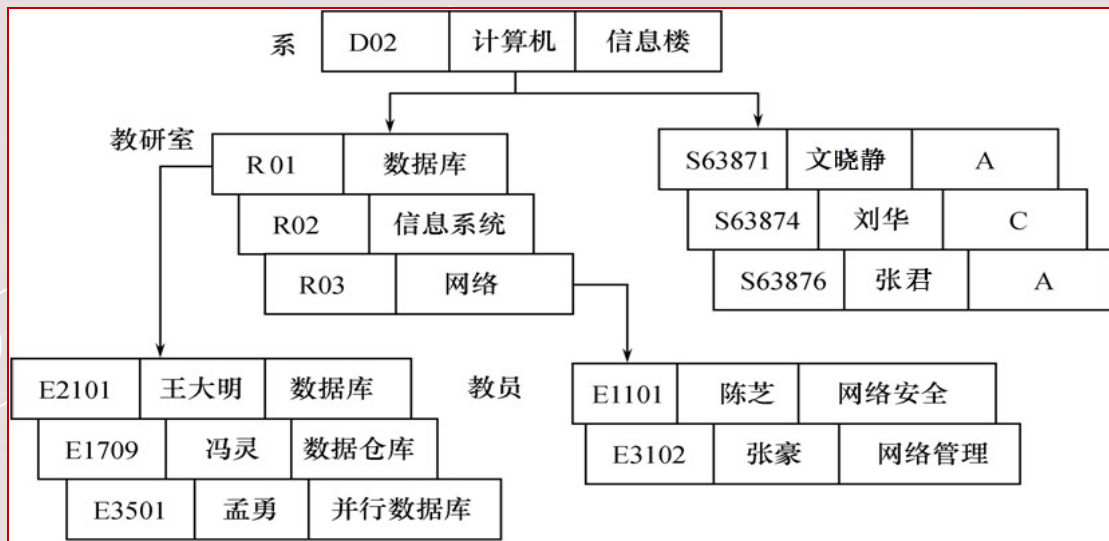


图1.11 教员学生层次数据库的一个值





## 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

- 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
- 层次数据库的命令（语言）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结构来表示各类实体以及实体间的联系

❖ 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

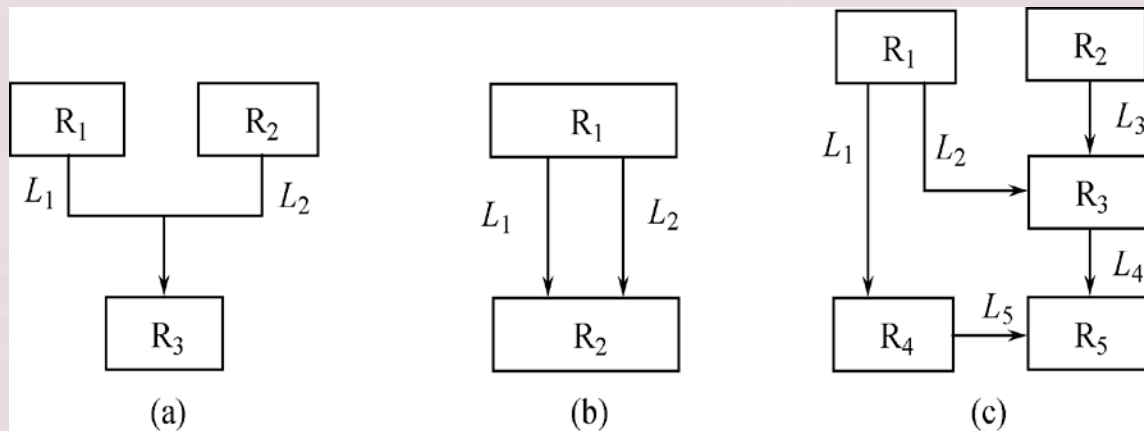
❖ 网状模型的定义

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。



# 1. 网状模型的数据结构



网状模型的例子

## 特点

- 允许多个结点没有双亲结点
- 允许一个结点有多个双亲结点
- 允许两个结点之间有多种联系
- 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





# 网状模型的数据结构（续）

多对多联系在网状模型中的表示

❖ 网状模型**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**

例：

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：**

选课（学号，课程号，成绩）

/\* 某个学生选修某一门课程及其成绩 \*/

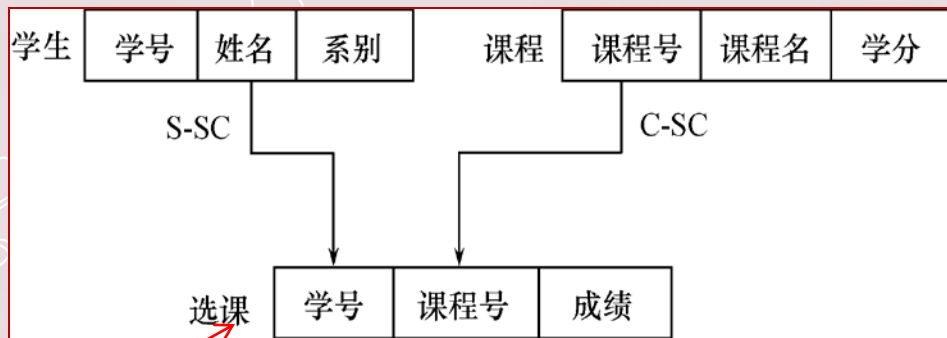


图1.13 学生/选课/课程的网状数据模型



## 2. 网状模型的数据操纵与完整性约束

❖ 导航式的查询语言和增删改操作语言

❖ 完整性约束条件不严格

- 允许插入尚未确定双亲结点值的子女结点值
- 允许只删除双亲结点值



## 2. 网状模型的数据操纵与完整性约束

❖ 实际的网状数据库系统**提供了一定的完整性约束**

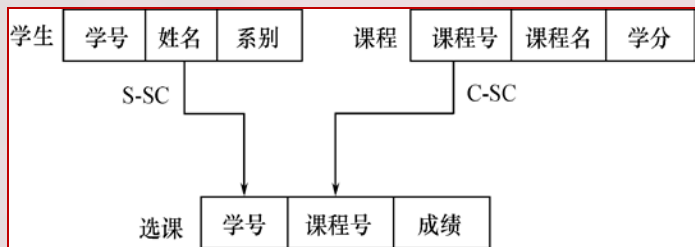
- 支持码的概念：唯一标识记录的数据项的集合，取唯一的值；
- 保证一个联系中双亲记录与子女记录之间是一对多联系；
- 可以定义双亲记录和子女记录之间某些约束条件。

**例：“属籍类别”的概念**

要求双亲记录存在才能插入子女记录，双亲记录删除时也连同删除。

选课记录就应该满足这种约束条件：

学生选课记录值中学号必须是学生记录中存在的某一学生的学号，  
课程号必须是课程记录中存在的某一门课程号。



# 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，应用程序必须选择存取路径，加重了程序员的负担。





# 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。

学生登记表

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
...	...	...	...	...	...

属性

元组



# 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
- 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。

例：学生年龄属性的域（15~45岁），  
性别的域是（男，女），  
系名的域是一个学校所有系名的集合；

- 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	3305	2200	2850	360	812	7183
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

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“找什么”，不必详细说明“怎么找”。

——> 提高了数据的独立性，提高了用户生产率。



# 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

关系的两个不变性



# 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

## 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

# 1.2 数据模型

## ① 需要了解的

层次数据模型及网状数据模型的基本概念

层次和网状数据库的内容十分精简

这两类系统虽然有它们的缺点，但是执行效率高是他们的显著优点

## ② 需要牢固掌握的

概念模型的基本概念

数据模型的3个组成要素

关系数据模型的相关概念

## ③ 难点

数据模型的概念对于刚刚学习数据库的读者来说会感到比较抽象。





