
第2章 数据模型

第2章 数据模型

- 2.1 数据模型的基本概念
- 2.2 数据模型的四个世界
- 2.3 概念世界与概念模型
- 2.4 信息世界与逻辑模型
- 2.5 计算机世界与物理模型

2.1 数据模型的基本概念

数据库是一个统一、集中的数据管理机构，它在保存用户所需数据的同时，也必须具有向外界提供数据服务的功能，即为用户提供**存取**数据库中数据的手段。因此如何描述现实世界中各种各样的数据和它们之间的各种复杂的关系，实现用户对数据的操作要求，并最终以计算机及数据库所允许的形式反映到数据库中去，这是一个非常重要的问题

在这里，我们使用**数据模型**这个概念来描述现实世界中的数据及其相互关系，定义在这些数据上可以执行的操作

2.1 数据模型的基本概念

□ 什么是数据模型？

➤ 数据是对于现实世界的符号抽象，而数据模型则是对数据特征的抽象，为数据库系统的信息表示和操作提供一个抽象框架，是数据库系统的核心与基础

—数据模型应该能比较真实地模拟现实世界、易于人理解、便于在计算机上实现

数据模型

特征抽象

数据

符号化

现实世界

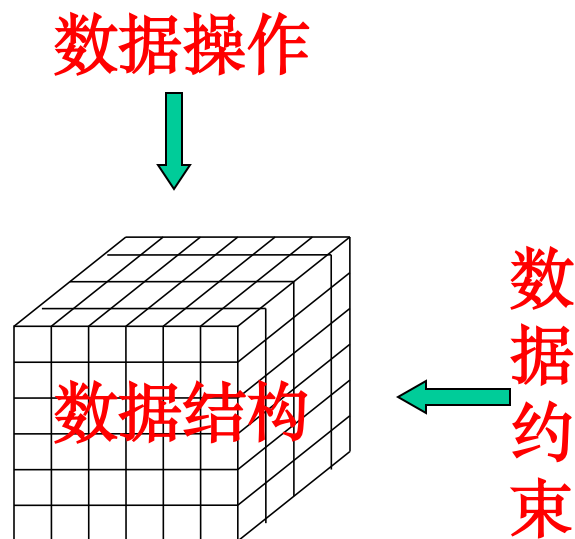
2.1 数据模型的基本概念

□ 数据模型 (data model)

【定义】描述数据的结构，定义在该数据结构上可以执行的操作以及数据之间必须满足的约束条件

➤ 数据模型的组成

- 数据结构
- 数据操作
- 数据约束



2.1 数据模型的基本概念

□ 数据模型 – 数据结构

➤ 描述数据的类型、内容、性质以及数据间的联系

– 数据结构是一个数据模型的基础，数据操作与数据约束均是建立在相应的数据结构上的

▪ 在这之前，数据模型中的数据结构被称为‘数据模式’

– 这也是不同类型数据模型的划分依据

2.1 数据模型的基本概念

□ 数据模型 – 数据操作

- 在相应数据结构上可以执行的操作类型与操作方式
 - 在不同的数据结构上可以提供不同的操作方式与操作类型

□ 数据模型 – 数据约束

- 主要描述数据结构内数据间的相互关系，包括：
 - 数据间的语法/语义联系
 - 数据间的制约与依存关系
 - 数据（间）的动态变化规则
- 其目的是确保数据的正确、有效与相容

2.1 数据模型的基本概念

- 数据模型的核心是数据结构，如何将现实世界中我们需要的数据及其复杂关系最终反映到数据库中去，这需要有一个逐步转化的过程
- 我们用建立在不同抽象层次上的‘**数据模型**’来表示每一步转化的结果：
 - **概念数据模型** (conceptual data model)
 - 又简称为 ‘**概念模型**’
 - **逻辑数据模型** (logic data model)
 - 又简称为 ‘**数据模型**’
 - **物理数据模型** (physical data model)
 - 又简称为 ‘**物理模型**’

2.1 数据模型的基本概念

□ 概念数据模型

- 侧重于对客观世界中复杂事物的结构描述及它们之间的内在联系的刻画，不涉及具体的描述细节和物理实现因素
 - 是一种面向客观世界和用户的模型，与具体采用的DBMS及计算机实现无关
 - 主要的几种概念模型
 - E-R模型，EE-R模型
 - 面向对象模型
 - 谓词模型

2.1 数据模型的基本概念

□ 概念数据模型 (cont.)

- 概念数据模型主要描述这些客观对象的数据特征及其相互关系
- 【例】 学生，教师，运动员，教练

2.1 数据模型的基本概念

□ 这些客观对象的数据特征如下：

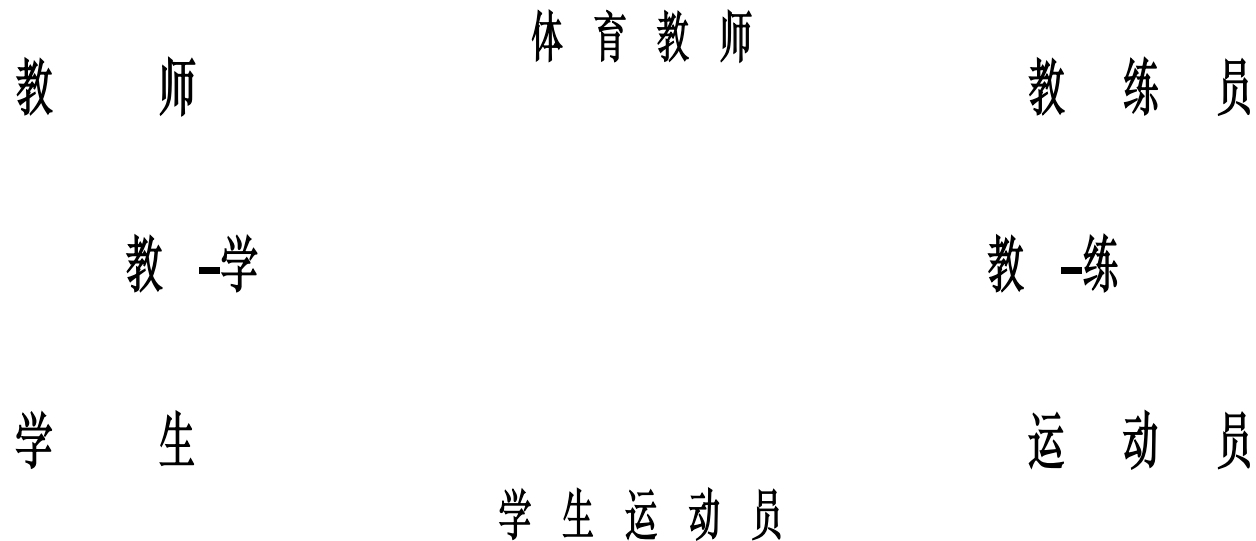
对 象	数 据 特 征
学 生	学 号 ， 姓 名 ， 系 ， 专 业 ， 性 别 ， 入 学 年 份
教 师	姓 名 ， 性 别 ， 出 生 日 期 ， 职 称 ， 学 历 ， 参 加 工 作 时 间 ， 工 作 证 编 号 ， 身 份 证 号
运 动 员	姓 名 ， 性 别 ， 出 生 日 期 ， 从 事 运 动 项 目 ， 身 高 ， 体 重
教 练 员	姓 名 ， 性 别 ， 出 生 日 期 ， 职 称 ， 从 事 运 动 项 目 ， 参 加 工 作 时 间 ， 工 作 证 编 号 ， 身 份 证 号

— 要了解每个数据项的语义含义，但并不需要定义其实现细节（如数据类型，取值的约束等）

2.1 数据模型的基本概念

□ 概念数据模型 (cont.)

➤ 相互关系的描述



2.1 数据模型的基本概念

□ 逻辑数据模型

- 着重于数据模型在数据库系统一级的实现，即利用具体的DBMS所提供的工具（DDL）来定义的数据模型
 - 是一种面向数据库系统的模型，概念数据模型只有在转换成逻辑数据模型后才能在数据库中得以表示
 - 是一个中介模型，具有承上启下的作用

2.1 数据模型的基本概念

□ 逻辑数据模型 (cont.)

➤ 成熟并（曾经）得到大量使用的逻辑数据模型有：

— 层次模型、网状模型

— 关系模型、面向对象模型、谓词模型

— 对象关系模型

2.1 数据模型的基本概念

□ 逻辑数据模型 (cont.)

- 需要描述每个客观事物及其相互关系在选定的DBMS中的实现结构
- 即根据选定的DBMS来定义客观事物及其相互关系的实现结构

2.1 数据模型的基本概念

□ 例如:

➤ 客观事物的实现结构:

- 关系数据库: 表及其属性的定义, 如:
 - 属性的名称、数据类型、取值约束等
 - 表级的取值约束
- 面向对象数据库: 类及其属性、方法的定义

➤ 相互关系的实现结构:

- 关系数据库: 表及其外键
- 面向对象数据库: 类的继承与合成关系

2.1 数据模型的基本概念

□ 物理数据模型

- 给出了数据模型在计算机内部的真正物理结构，是一种面向计算机物理实现的模型
- 一个概念数据模型将首先被转化为某一种逻辑数据模型，并通过所选择的DBMS将其进一步转化为具体的物理数据模型，才能使其在计算机中得以物理实现

2.1 数据模型的基本概念

□ 物理数据模型 (cont.)

➤ 大都由选定的某种DBMS来负责数据库物理存储结构的选择，但也向用户提供了一些与物理存储结构和存取方法有关的定义功能，如：

— 索引 (index) 的定义

— 集簇 (cluster) 的定义

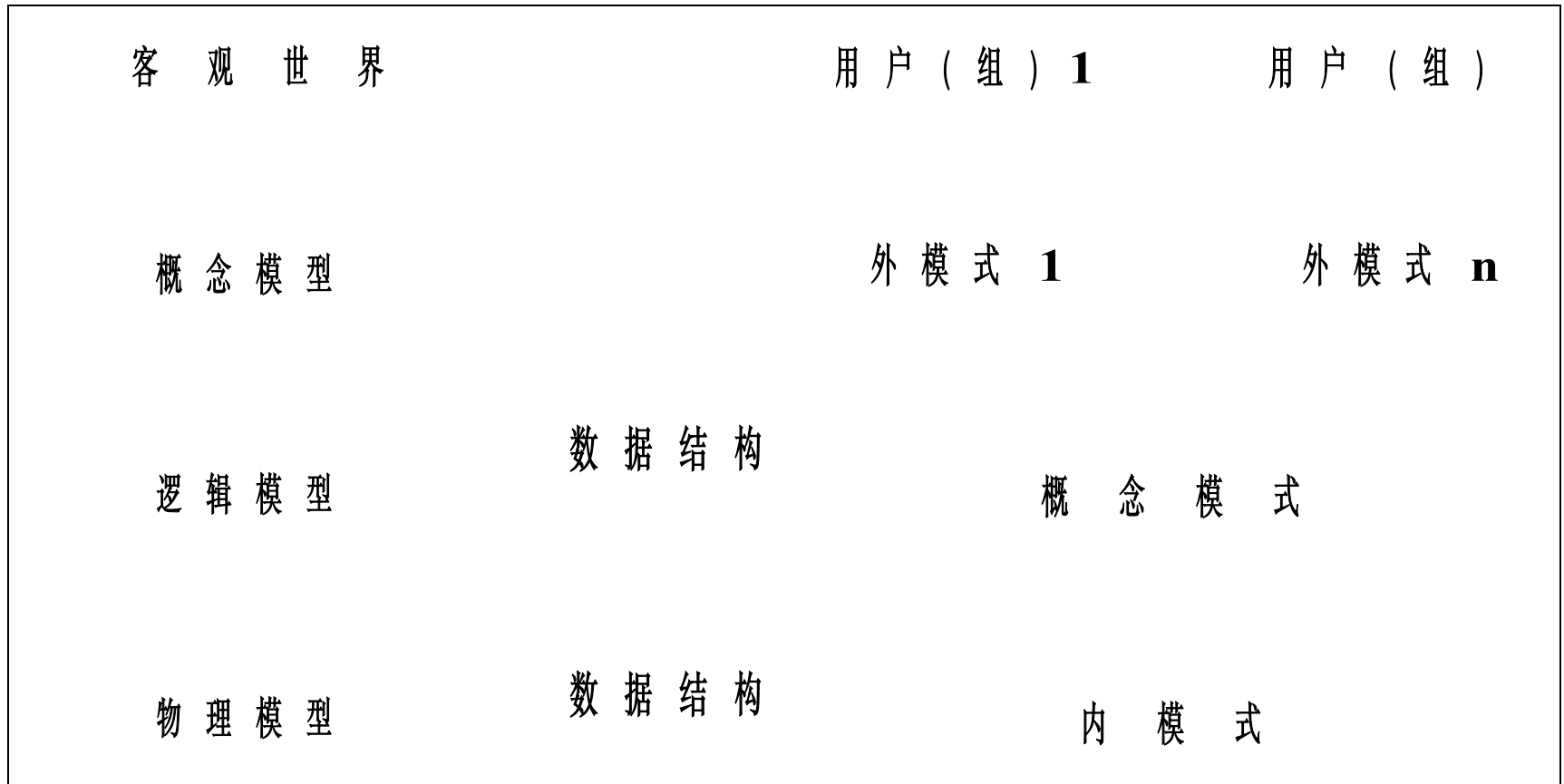
— 存储区域的选择

- 例如：Oracle用户可以选择表中数据的物理存储所使用的表空间 (tablespace) 等

2.1 数据模型的基本概念

□ 三种‘数据模型’与‘三级模式’之间的关系

- 是不同的‘模型’与‘模式’的分层方法，请注意它们之间的区别



第2章 数据模型

2.1 数据模型的基本概念

2.2 数据模型的四个世界

2.3 概念世界与概念模型

2.4 信息世界与逻辑模型

2.5 计算机世界与物理模型

2.2 数据模型的四个世界

□ 使用‘**数据模型**’概念可以将现实世界中用户的复杂要求反映到计算机数据库中的实现，这种反映是一个逐步转化的过程，它分为四个阶段，我们称其为四个世界。每一次转化都是一个加工与提高的过程

- 现实世界
- 概念世界
- 信息世界
- 计算机世界

2.2 数据模型的四个世界

□ 现实世界

- 在客观世界中根据用户的需求目标而划定边界的一个应用环境

— 用户需求

- 数据需求
 - 处理要求
- } 从而确定了数据库应支持的应用功能和应用范围

- 现实世界为整个转换过程提供了客观基础与初始启动环境

2.2 数据模型的四个世界

□ 概念世界

- 以现实世界为基础作进一步的抽象而形成的概念模型
- DB设计人员经过调查研究会从用户那里获得一组庞大复杂的需求信息，由于用户在理解要求和描述表示上的不足，因此难免会向DB设计人员提供一些多余的、错误的或者不准确的信息，因而需要DB设计人员对其作分析提高，去粗取精，去伪存真，最后形成一些基本概念与基本关系。这些基本概念与基本关系可以用我们所选择的某一种概念数据模型中所提供的术语和方法来统一表示

2.2 数据模型的四个世界

□ 概念世界中的基本术语

【例】 E-R模型： 实体， 属性， 联系

【例】 OO模型： 对象， 类， 方法， 继承，

➤ 概念世界与具体的DBMS和计算机无关

2.2 数据模型的四个世界

□ 信息世界

- 以概念世界为基础，选用特定的**DBMS**构造而成的逻辑数据模型
 - 侧重于概念数据模型的细化和在数据库系统统一级的实现，即利用特定的**DBMS**所提供的工具来定义逻辑数据模型
 - 该模型的定义与具体的**DBMS**有关

2.2 数据模型的四个世界

□ 计算机世界

➤ 基于逻辑数据模型在计算机中的物理实现而形成的物理数据模型

— 侧重于数据库物理存储结构的描述

- 存储结构的设计

- 存取路径的设计

 - 文件结构的选择：堆/直接/索引 文件等

 - 索引/集簇 的设计

- 存储空间的分配

— 是DB的最终实现结构

2.2 数据模型的四个世界

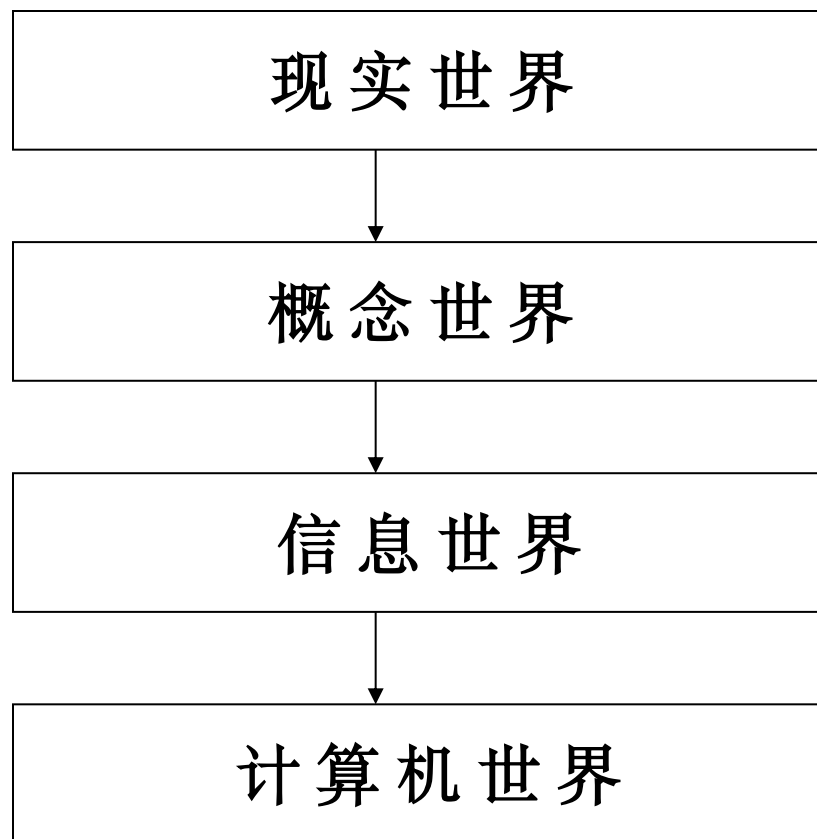


图2-1：四个世界的转化示意图

第2章 数据模型

2.1 数据模型的基本概念

2.2 数据模型的四个世界

2.3 概念世界与概念模型

2.4 信息世界与逻辑模型

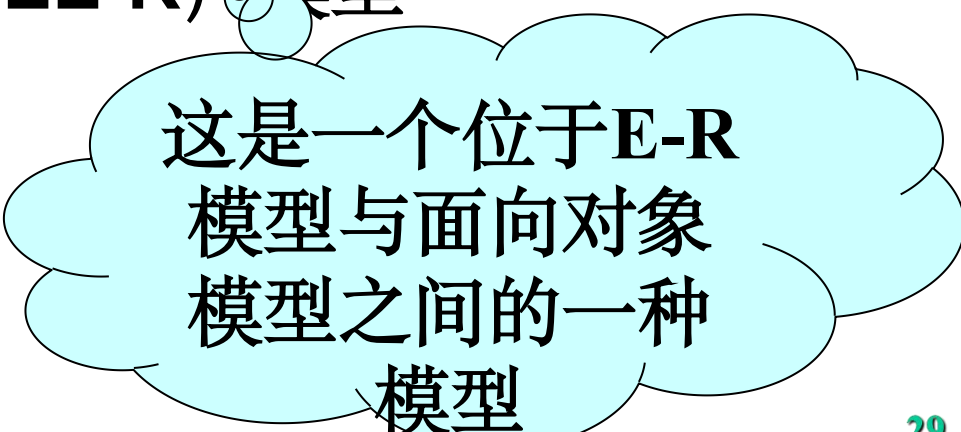
2.5 计算机世界与物理模型

2.3 概念世界与概念模型

□ 概念世界是一个较为抽象、概念化的世界，它给出了数据的概念化结构。概念世界一般用概念模型表示

□ 常用的几种概念模型

- 实体-联系 (E-R) 模型
- 扩充的实体-联系 (EE-R) 模型
- 面向对象模型
- 谓词模型



这是一个位于E-R模型与面向对象模型之间的一种模型

2.3.1 实体-联系模型

□ 实体-联系模型

- Entity-Relationship model, 简称E-R模型
- 这是一种概念化的模型, 它将现实世界的要求转化成实体、联系、属性等基本概念及它们之间的两种基本关系, 并且用一种较为简单的图表示, 称**E-R图** (Entity-Relationship diagram)

– E-R模型中的基本概念

- 实体
 - 属性
 - 联系
- } 以及它们之间的连接关系

2.3.1 实体-联系模型

□ 实体(Entity)

➤ 客观存在且又能相互区别的事物

— 是对现实世界中的客观事物的抽象，是概念世界中的基本单位

➤ 实体集

— 由具有共性的实体所构成的集合

2.3.1 实体-联系模型

□ 属性 (Attribute)

- 实体所具有的某种特性或特征
 - 属性可以有值
 - 一个属性可以取的值的集合，被称为该属性的**值域** (value domain)
- 一个实体可以有多个属性
 - 所谓具有共性的实体是指这些实体含有相同的属性组成
 - 不同的实体在这些属性上的取值则存在着区别

2.3.1 实体-联系模型

□联系 (Relationship)

- 一个实体集中的实体与另一个实体集中的实体之间的对应关系。在概念世界中，我们用两个实体集的联系来反映它们之间的这种关系
- 联系的种类（与联系相关的实体集的个数）
 - 两个实体集间的联系 (二元联系)
 - 多个实体集间的联系 (多元联系)
 - 单个实体集内部的联系

2.3.1 实体-联系模型

□ 联系的例子

➤ 两个实体集之间的联系

- 隶属关系 (教师, 系别)
- 学习关系 (学生, 课程)
- 借阅关系 (学生, 图书)

选手甲和选手乙都是同一个实体集‘**运动员**’中的实体

➤ 多个实体集之间的联系

- 供应关系 (工厂, 产品, 用户)

➤ 单个实体集内部的联系

- 围棋比赛 (黑方(选手甲), 白方(选手乙))

2.3.1 实体-联系模型

□相同实体集之间的多种联系

➤在同一组实体集之间可以存在多种联系

➤例如:

—职工之间

- 管理关系（上下级关系）
- 同事关系

—教师与研究生之间

- 教学关系
- 指导关系

2.3.1 实体-联系模型

□联系的函数对应关系 (图2-2)

- 一一对应 (one to one)
- 一多对应 (one to many)
 - 或: 多一对应 (many to one)
- 多多对应 (many to many)

2.3.1 实体-联系模型

□联系的函数对应关系

➤ 一一对应 (1 : 1)

实体集 A

实体集 B

—例如： ‘学校’ 与 ‘校长’
 ‘居民’ 与 ‘身份证’

2.3.1 实体-联系模型

□联系的函数对应关系

➤一多对应 ($1 : m$)

实体集 A

实体集 B

—例如： ‘学生’ 与 ‘宿舍’

2.3.1 实体-联系模型

□联系的函数对应关系

➤ 多多对应 ($m : n$)

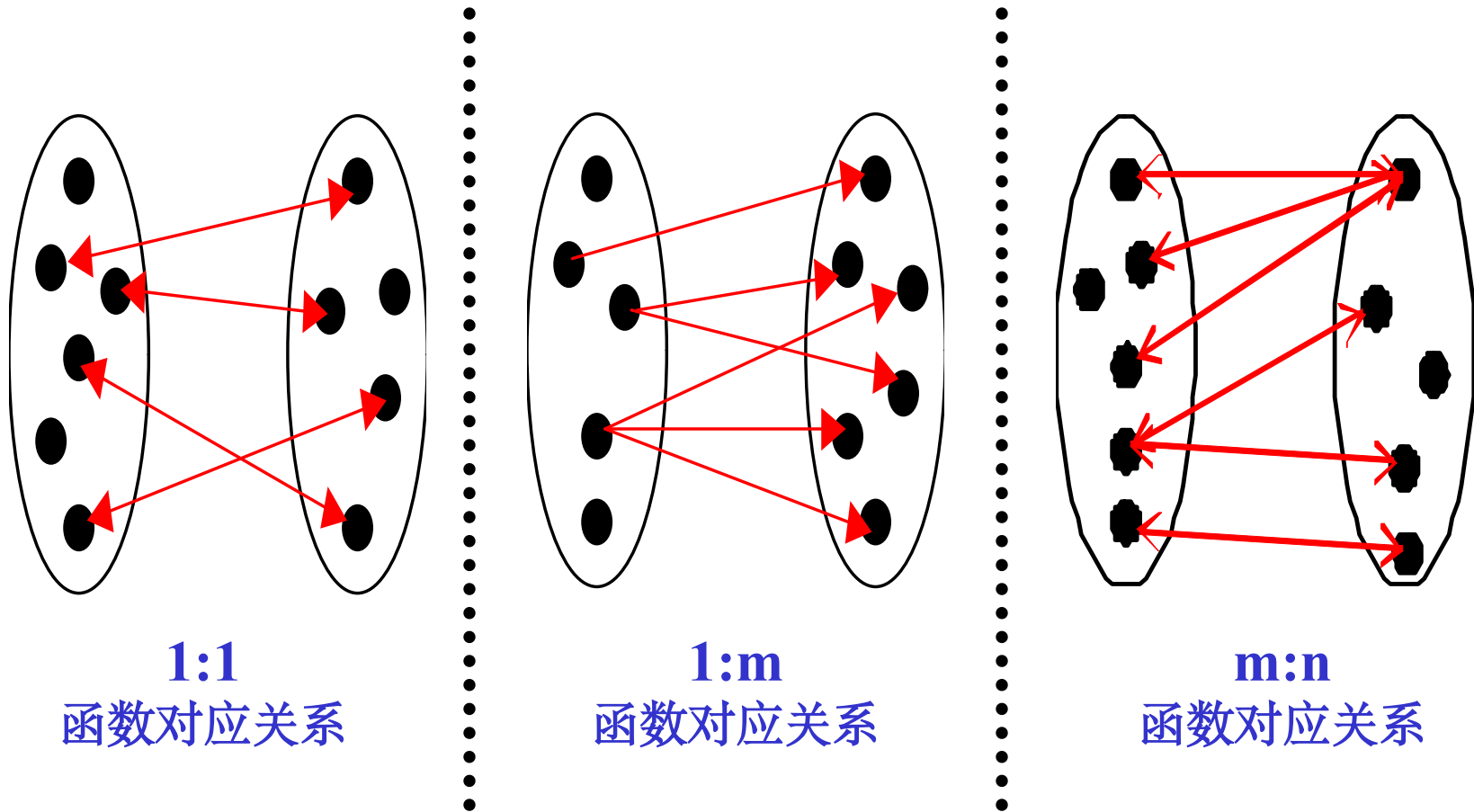
实体集 A

实体集 B

—例如： ‘学生’ 与 ‘课程’

2.3.1 实体-联系模型

□ 联系的函数对应关系



2.3.1 实体-联系模型

□联系所具有的特性

- 因联系的发生而产生的特性可以通过联系上的属性来表示

□例如:

- 学习关系 (学生, 课程, 成绩)
- 借阅关系 (学生, 图书, 借阅日期, 归还如期)
- 比赛 (甲方, 乙方, 比赛结果)

2.3.1 实体-联系模型

□ 基本概念之间的连接关系

- 1) 实体集（联系）与属性间的连接关系
- 2) 实体集与联系间的连接关系

□ 如何描述实体、属性、联系以及它们之间的连接关系？

2.3.1 实体-联系模型

□ 实体（集）、属性及其连接关系的描述

➤ 属性的描述：属性名，属性域

➤ 实体的描述：

— 实体名

— 实体型：实体名 + 一组属性名

- 用于描述实体的组成结构信息

— 实体值

- 实体中的每个属性都可以取值，由一个实体的所有属性的取值所构成的属性值的集合被称为该实体的实体值

- 在关系模型中，‘实体值’又被称为‘元组’
(tuple)

2.3.1 实体-联系模型

□ 实体（集）、属性及其连接关系的描述 (cont.)

➤ 实体集的描述

— 由具有相同实体型的实体所构成的集合被称为实体集

— 实体集的描述:

属性集合 + 关键字(key)

2.3.1 实体-联系模型

□ 实体（集）描述的例子，带下划线的属性为各实体集的关键字属性

实 体	属 性
学 生	<u>学 号</u> ， 姓 名， 系， 专 业， 性 别， 入 学 年 份
教 师	姓 名， 性 别， 出 生 日 期， 职 称， 学 历， 参 加 工 作 时 间， <u>工 作 证 编 号</u> ， <u>身 份 证 号</u>
运 动 员	<u>运 动 员 证 编 号</u> ， 姓 名， 性 别， 出 生 日 期， 从 事 运 动 项 目， 身 高， 体 重， <u>身 份 证 号</u>
教 练 员	姓 名， 性 别， 出 生 日 期， 职 称， 从 事 运 动 项 目， 参 加 工 作 时 间， <u>工 作 证 编 号</u> ， <u>身 份 证 号</u>

2.3.1 实体-联系模型

□联系及其与实体集之间的连接关系的描述

➤ **联系名**：每个联系有一个名字

➤ **属 性**：联系也可以有属性

— 由一个‘联系名’ + 与该联系相关的‘实体集的名称’，以及联系上的属性，从而构成联系及其与实体集之间的连接关系的描述

➤ **函数对应关系**

□ 例如：

选课联系（学生，课程，成绩）

2.3.1 实体-联系模型

- 如果是单个实体集内部的联系，那么在描述它们之间的连接关系时，需要描述清楚参与联系的双方（或多方）在该联系中所担当的角色
- 例如：
 - 职工的上下级联系（上级职工，下级职工）
 - 围棋比赛联系（黑方，白方）

2.3.1 实体-联系模型

□ E-R模型的图示法: E-R图

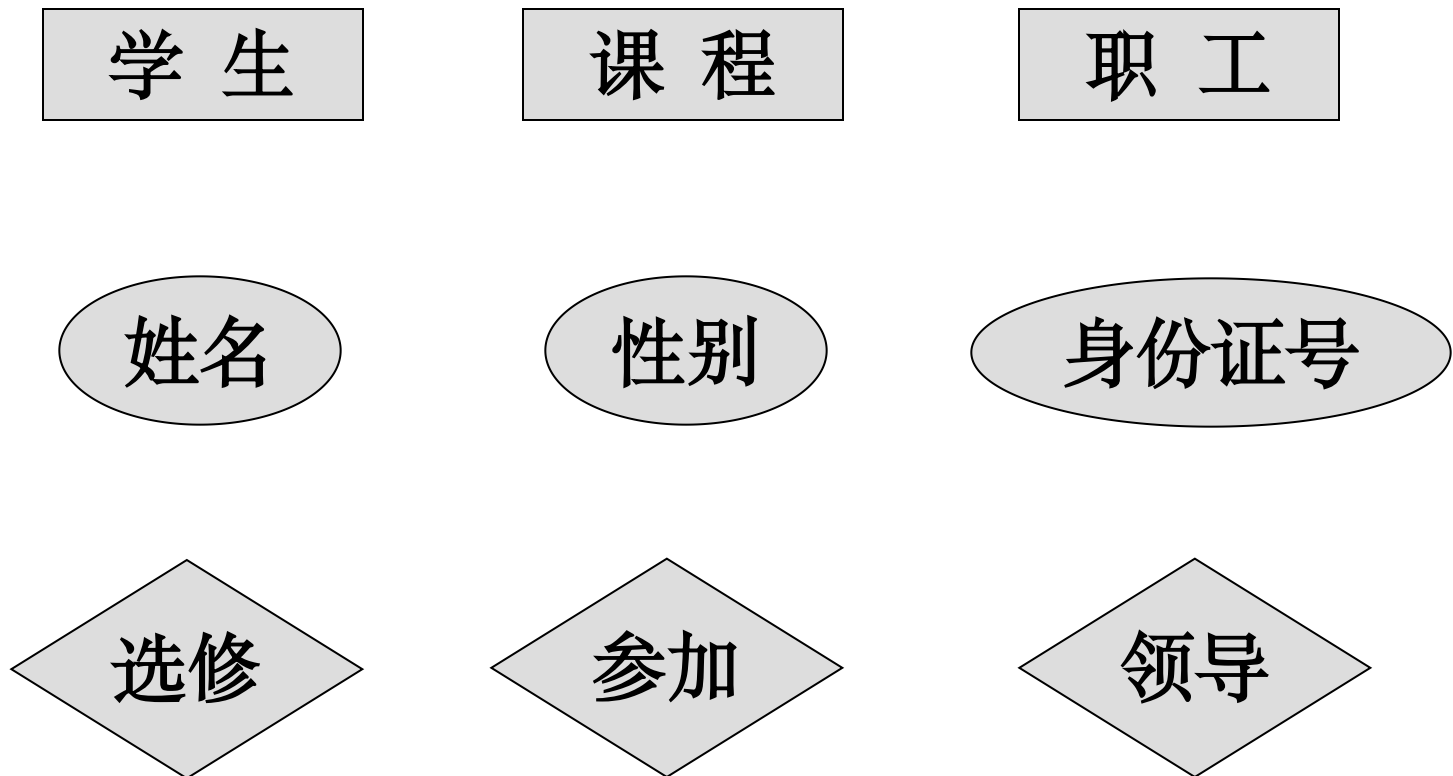
➤ 使用一些简单的图形符号来表示概念数据模型

➤ 基本概念的表示



2.3.1 实体-联系模型

□ 基本概念的名称应该写在相应的图形符号的方框范围内，例如：



2.3.1 实体-联系模型

➤ 连接关系的表示

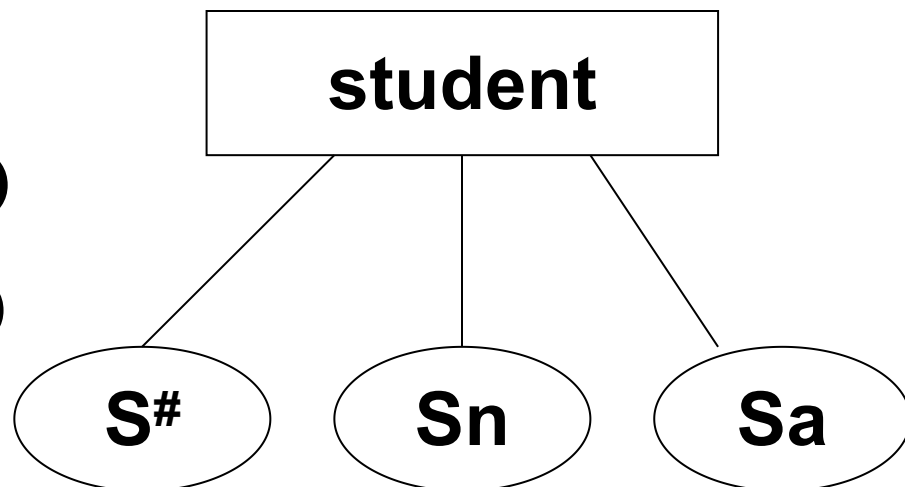
— 实体集与属性间的连接关系

- 实体集: **student**

- 属性: **S[#]** (学号)

Sn (学生姓名)

Sa (学生年龄)

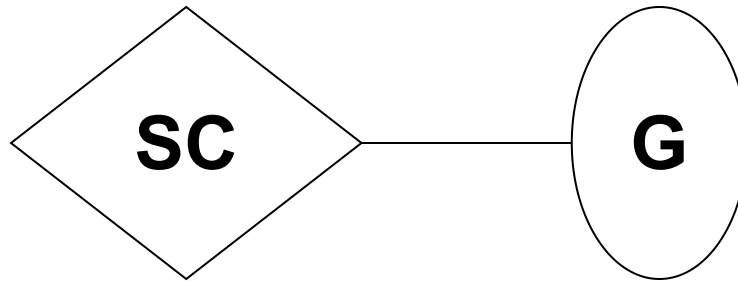


E-R图表示法

2.3.1 实体-联系模型

— 联系与属性的连接关系

- 联系: SC
- 属性: G (学生的课程成绩)

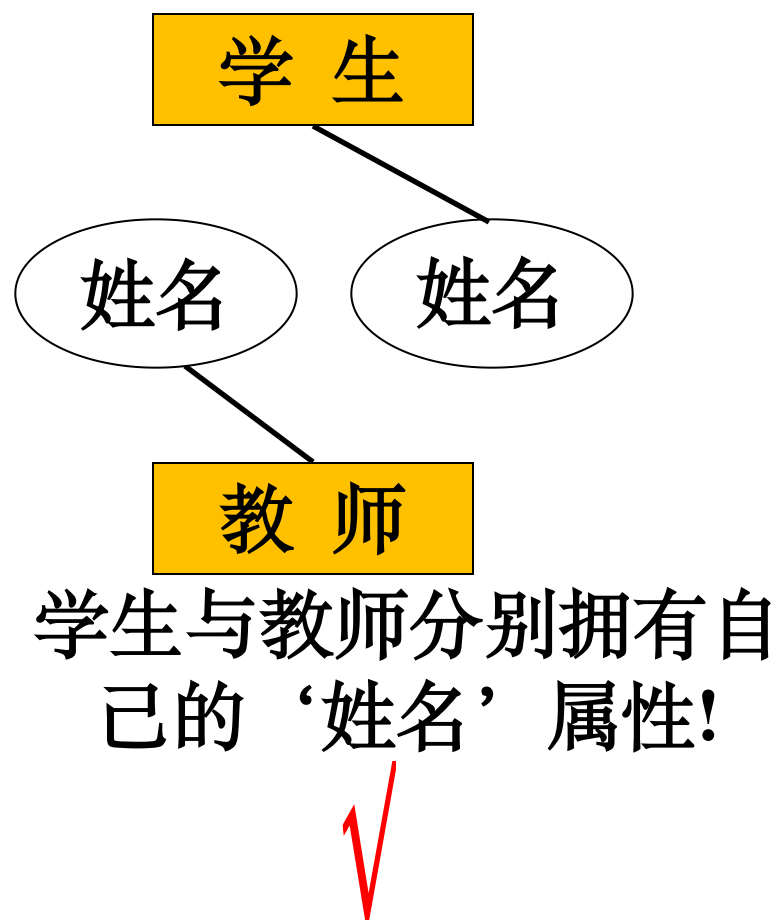
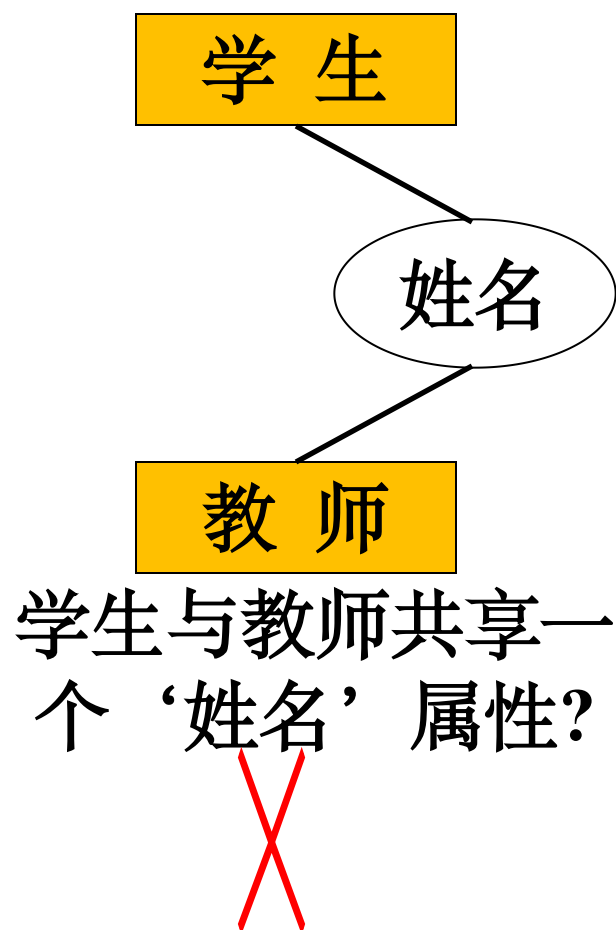


E-R图表示法

2.3.1 实体-联系模型

□ 连接关系的表示 (cont.)

- 每个实体集 (联系) 可以有多个属性, 但每个属性只能隶属于一个实体集 (联系)

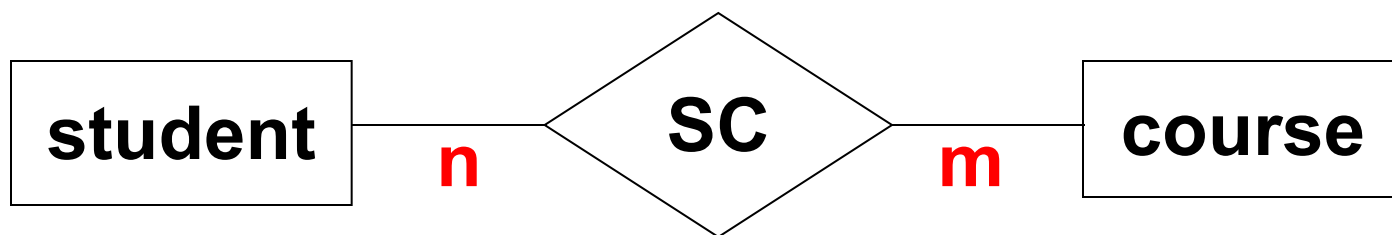


2.3.1 实体-联系模型

➤ 连接关系的表示 (cont.)

— 实体集与联系间的连接关系E-R图

- 实体集student与联系SC
- 实体集course与联系SC

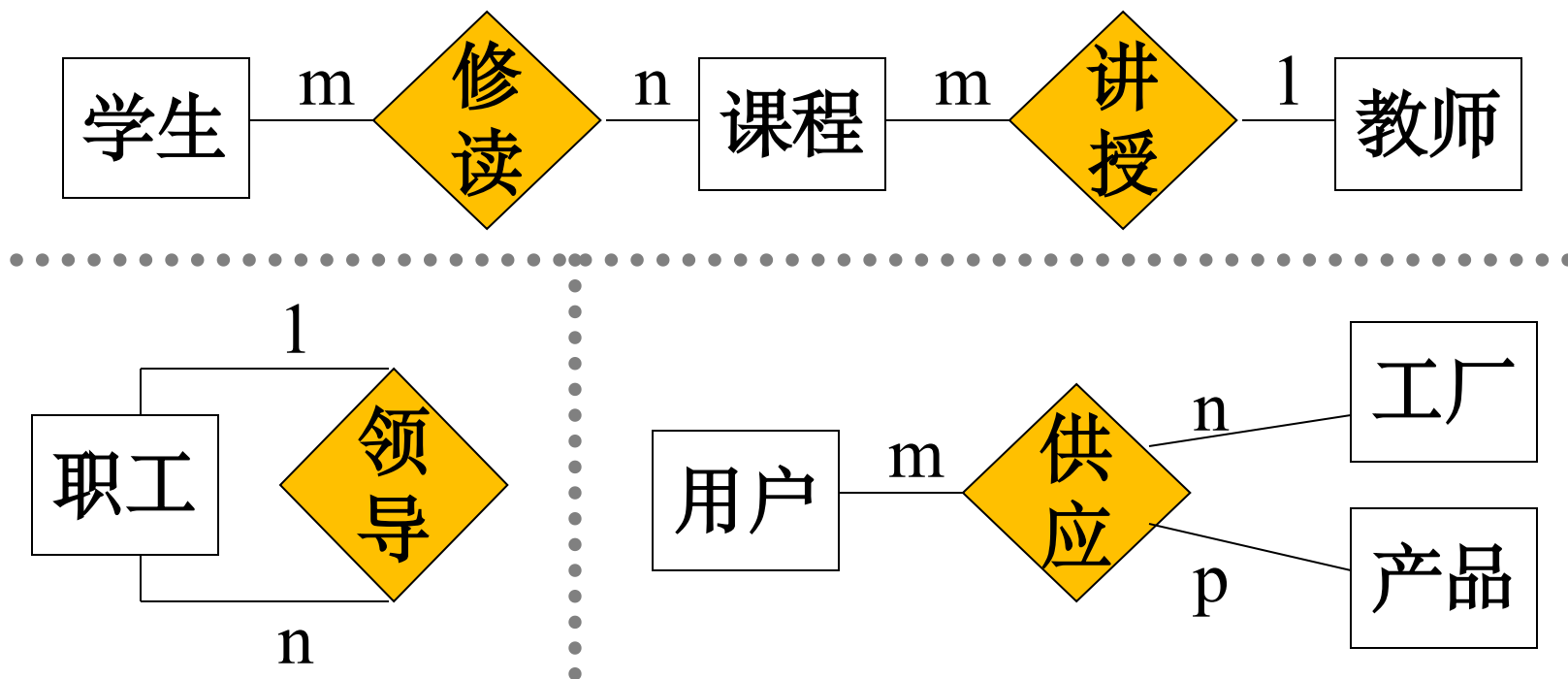


【注】为了刻画实体间的函数对应关系，必须在线段边上用1:1（一对一），1:n（一对多），n:m（多对多）等注明

2.3.1 实体-联系模型

□ 连接关系的表示 (cont.)

- 每个联系可以与一个或多个实体集相关，每个实体集也可以与一个或多个联系相关
- 例如（省略了实体集及联系上的属性）



2.3.1 实体-联系模型

□ 例2.1

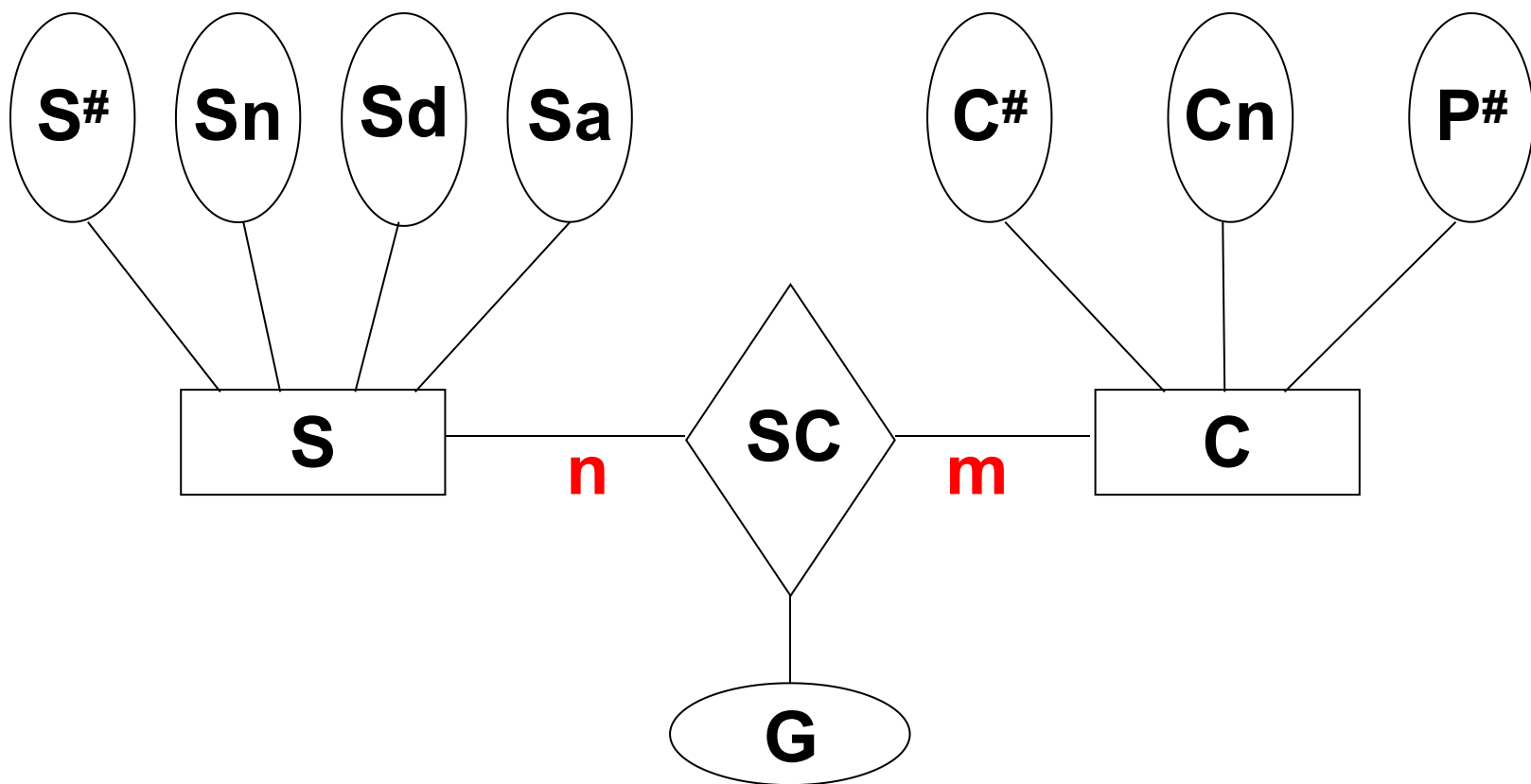


图2-12 E-R图的一个实例

2.3.1 实体-联系模型

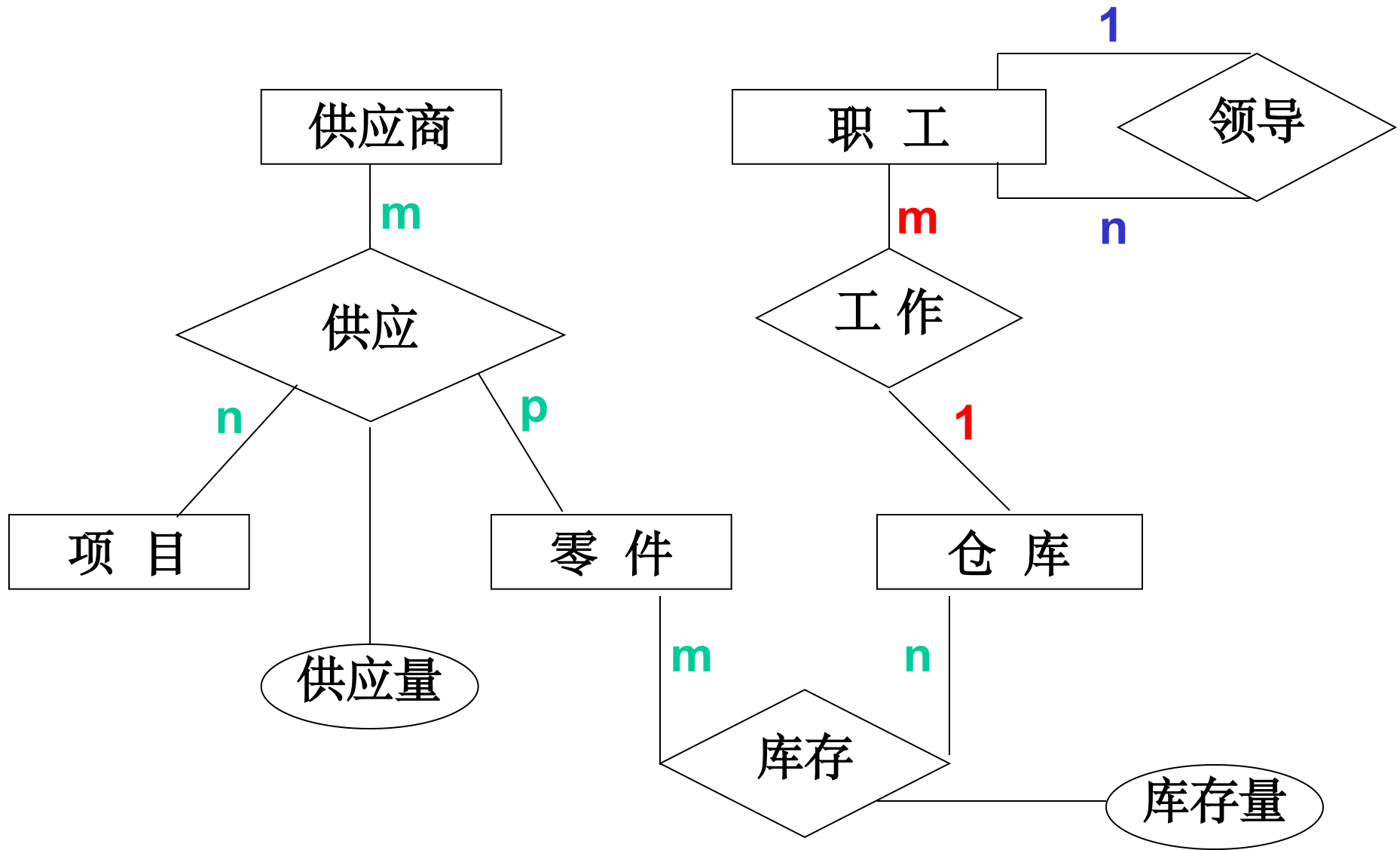
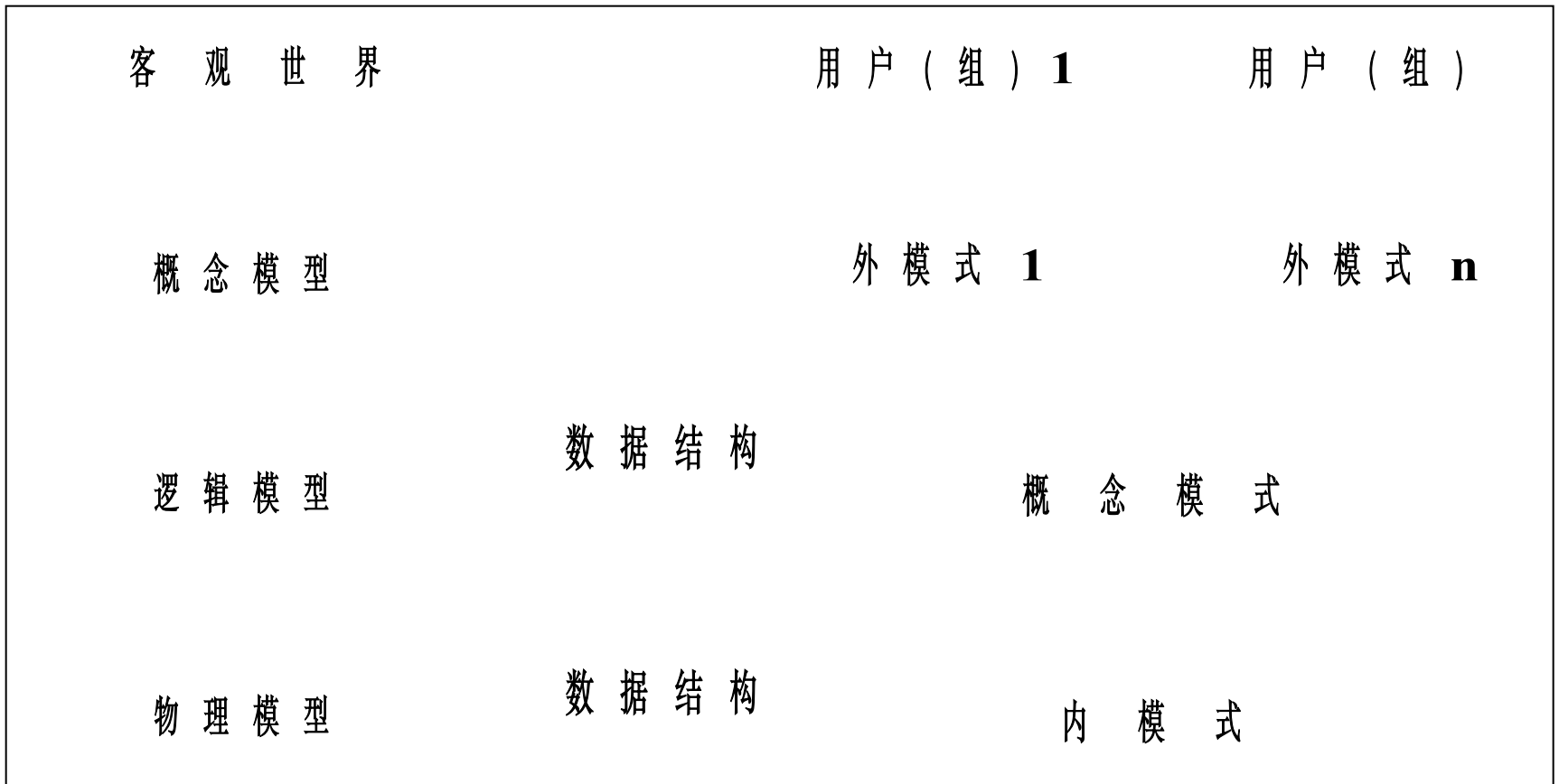


图2-13 某工厂的物资管理E-R图 (省略了属性)

回顾

- 数据模型：数据结构、数据操作、数据约束
- ‘数据模型’与‘三级模式’之间的关系



- E-R模型：实体、属性、联系（函数对应关系）

2.3.1 实体-联系模型

□ E-R模型的设计

- 标识系统中的 实体、属性与 联系
- E-R图表示

□ E-R模型的设计选择

- 实体 or 属性？
- 实体 or 联系？
- 二元联系 or 三（多）元联系？
- 联系的函数对应关系？
- 属性的依附对象：实体 or 联系？

2.3.1 实体-联系模型

□ 实体 or 属性 ?

- 实体: 需要进一步多方面的描述信息
- 属性: 单一的描述值 (非结构化的单值信息)

□ 实体 or 联系 ?

- 实体: 可以独立存在的持久对象
- 联系: 因为某种需要或某件事情的发生而产生的信息, 通常与现实世界中的多个对象有关

2.3.1 实体-联系模型

□二元联系 or 三(多)元联系？

- 基于一个联系的语义描述需要，以及因此而涉及到的实体的个数
- 在三(多)元联系中，在下述情况下也可以考虑采用若干个二元联系来实现
 - 用户只需要使用它们之间的两两联系
 - 不会出现二义性（歧义性）

□联系的函数对应关系？

- 基于系统的语义约束来定义

2.3.1 实体-联系模型

□ 属性的依附对象：实体 or 联系？

➤ 实体(集)中的属性

— 是该实体的内在特征，不会因为某些联系的出现而产生改变或消亡

➤ 联系上的属性

— 用于描述因联系的发生而需要记录、存储下来的信息

— 其值会随着联系的产生而出现，也会随着联系的消亡而消亡

模型设计实例

□ 图书借阅系统: Case One

□ 篮球联赛系统: Case Two

第2章 数据模型

2.1 数据模型的基本概念

2.2 数据模型的四个世界

2.3 概念世界与概念模型

2.3.1 E-R模型

2.3.2 **EE-R模型**

2.4 信息世界与逻辑模型

2.5 计算机世界与物理模型

2.3.2 扩充E-R模型

□ E-R模型在表示概念世界中使用较为普遍，但其在表示能力上尚有欠缺，如复杂的语义表达能力。因此很多人对E-R模型进行了扩充，构成了扩充的实体-联系模型，简称为EE-R模型

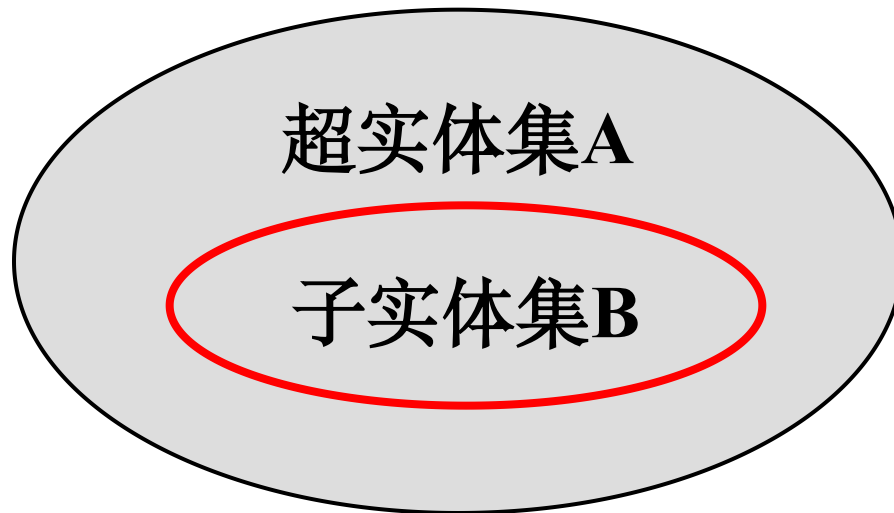
□ EE-R模型

- 扩充的E-R模型版本有很多，但其对E-R模型的扩充成分主要是“IS-A联系”
- EE-R模型也可以用一些基本的图形符号来表示，我们称其为EE-R图

2.3.2 扩充E-R模型

□ IS-A联系

- 如果实体集B是实体集A的一个子集，且具有比实体集A更多的属性，则我们称在实体集A与实体集B之间存在着一种特殊的‘IS-A联系’。其中：
 - 实体集A被称为实体集B的 超(实体)集
 - 实体集B被称为实体集A的 子(实体)集



□ 子集B可以通过IS-A联系继承超集A中的所有属性

2.3.2 扩充E-R模型

【例2.3和例2.4】 ‘学生’ 与 ‘研究生’

Student(学号S[#], 姓名Sn, 系别Sd, 年龄Sa)

G_student(S[#], Sn, Sd, Sa, **导师姓名**, **研究方向**)



- ‘研究生’是‘学生’的一个子集，每个研究生同时也是一个学生，也拥有学生所具有的特性（属性）
- 因此，我们在它们之间建立一种特殊的IS-A联系，使得‘研究生’实体集能够从‘学生’实体集那里继承得到它们的共性，而将注意力集中于‘研究生’自己的特性定义上

2.3.2 扩充E-R模型

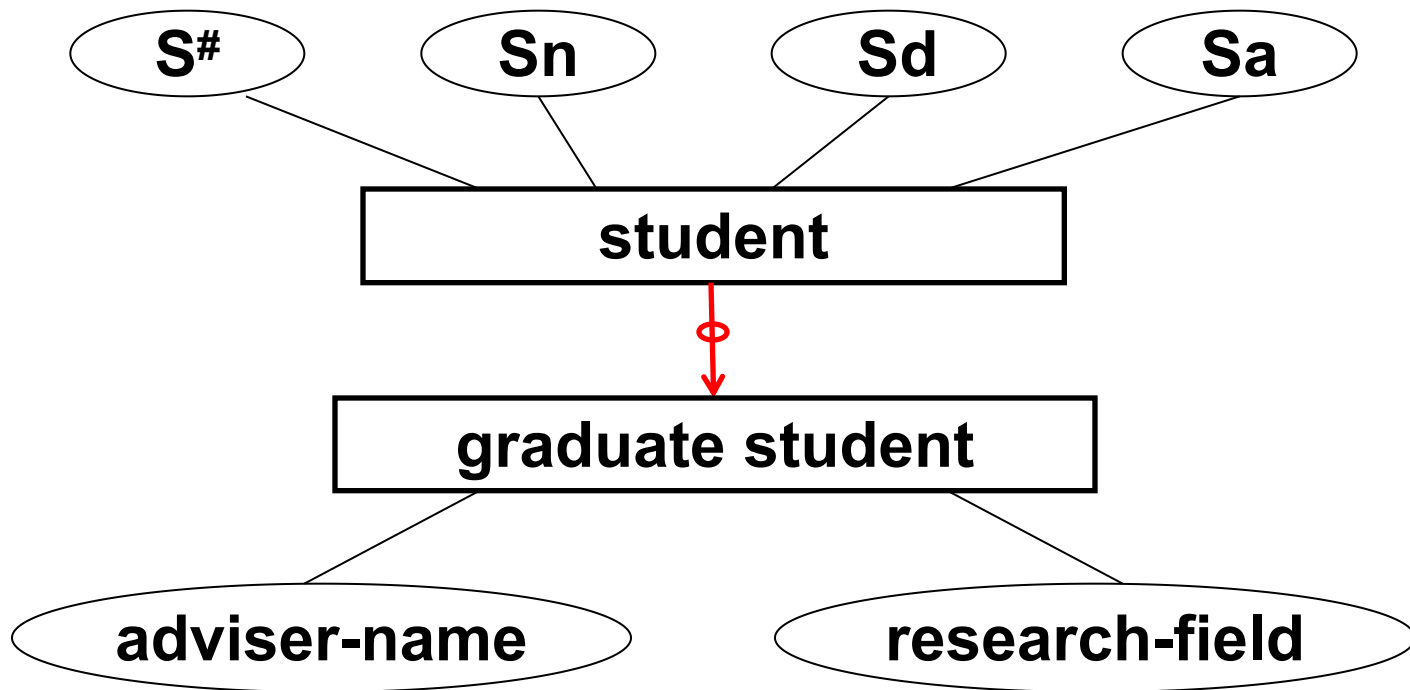
□ EE-R图

- 实体集、属性以及联系的图形表示方法与E-R图一样，这里只需要增加扩充部分的图形表示

2.3.2 扩充E-R模型

□ IS-A 联系的表示方法

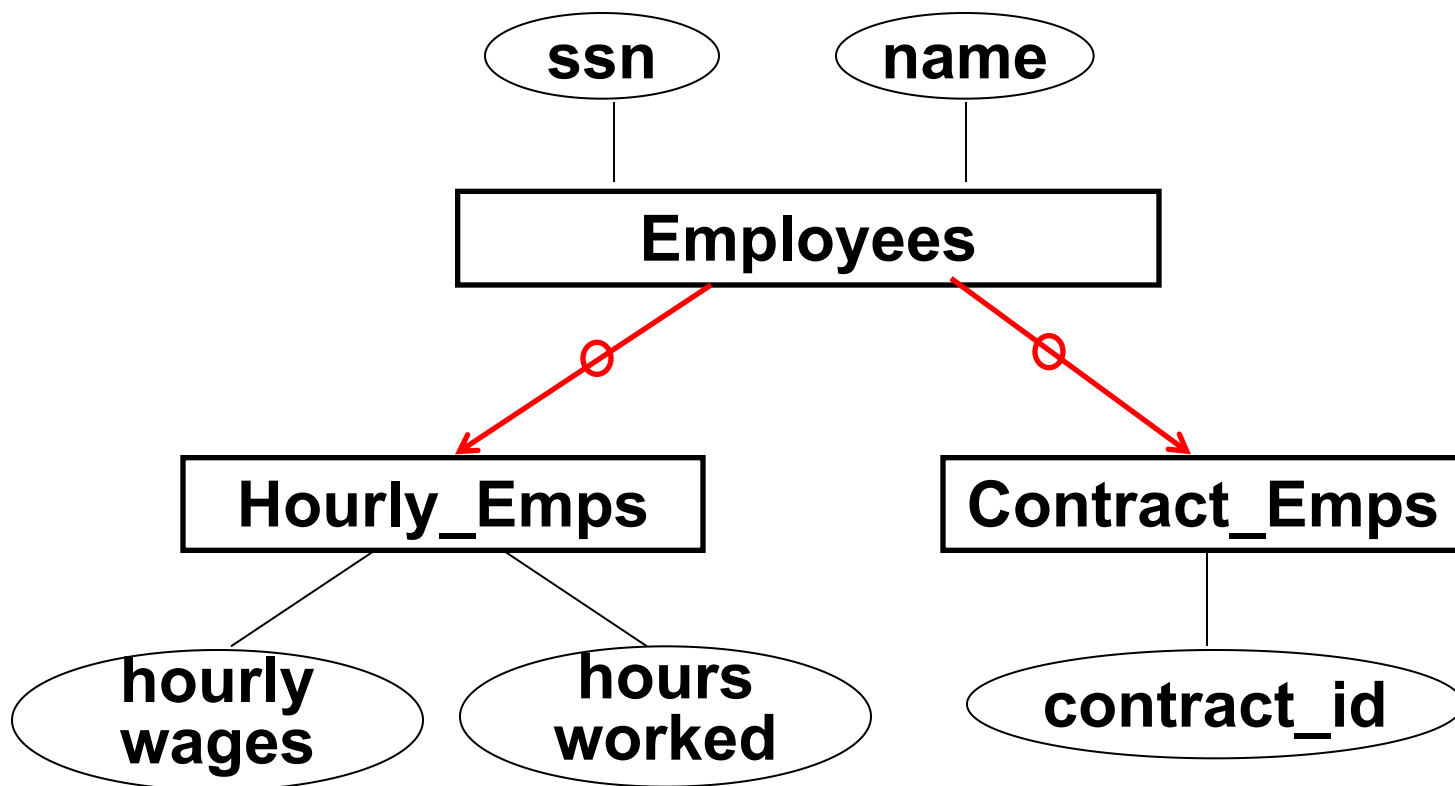
【例2.4】 图2.14



2.3.2 扩充E-R模型

□ IS-A联系的表示方法

【例】多个子实体集的例子



2.3.2 扩充E-R模型

□除了上述的扩充成分之外，还有一种比较有用的扩充成分被称为‘弱实体’

□弱实体 (Weak Entity)

➤如果一个实体A的存在需要依赖于其他实体集中的某个实体的存在，那么实体A被称为弱实体。例如：

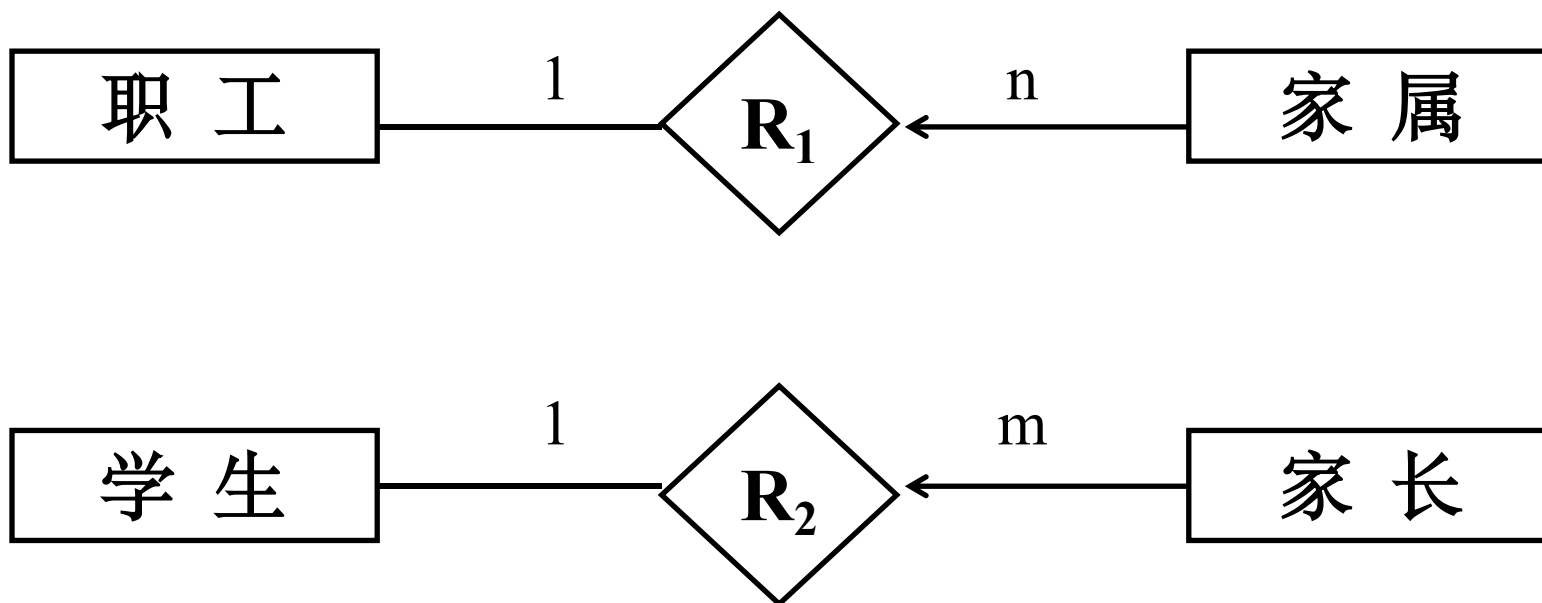
- 职工 vs 家属
- 学生 vs 家长

➤弱实体(集)与所依赖的实体(集)之间的函数对应关系应该是“多对一”的关系

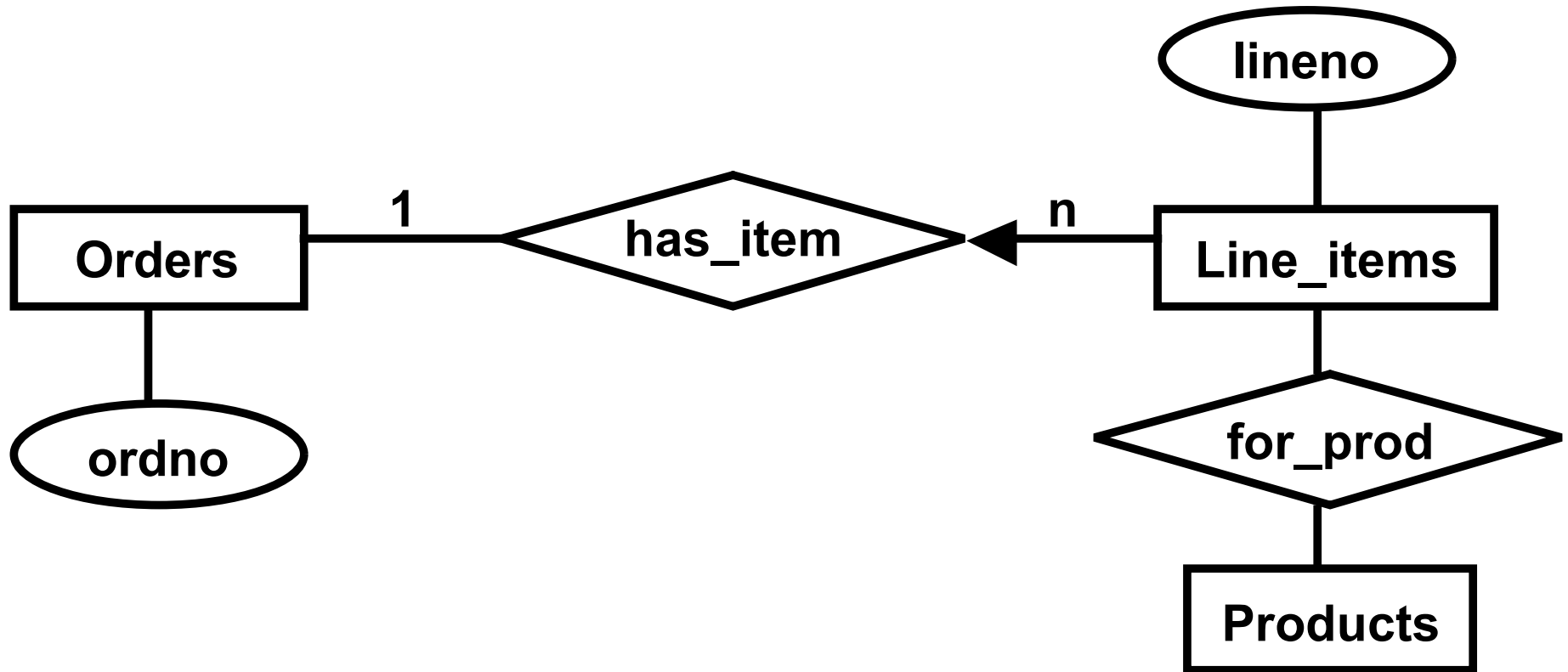
2.3.2 扩充E-R模型

□ 弱实体的表示

➤ 从弱实体到联系的有向箭头



弱实体的例子



订单、订单项与所订购的商品之间的EE-R图

第2章 数据模型

2.1 数据模型的基本概念

2.2 数据模型的四个世界

2.3 概念世界与概念模型

2.4 信息世界与逻辑模型

2.5 计算机世界与物理模型

2.4 信息世界与逻辑模型

□ 信息世界是数据库的世界，它着重于数据模型在数据库系统一级的构造与操作

➤ 信息世界用逻辑数据模型来进行描述

□ 逻辑模型的分类

➤ 由于在不同时期以及不同的数据库厂商之间采用的实现手段和方法的不同，因此逻辑模型的种类很多

➤ 在数据库技术的发展历程中，具有比较重要的影响和历史地位的逻辑模型有：

— 层次模型和网状模型

— 关系模型和对象关系模型

— 面向对象模型

— 谓词模型

2.4 信息世界与逻辑模型

□ The '60s: 数据库技术的萌芽阶段

➤ 1961: IDS (Integrated Data Store)

– Designed by Bachman

– 奠定了网络数据模型(Network Data Model)的基础

- Conference on Data Systems Languages Database Task Group (CODASYL DBTG)

2.4 信息世界与逻辑模型

- 1965-1970: IMS (Information Management System)
 - Developed by IBM
 - 奠定了层次数据模型(Hierarchical Data Model)的基础
 - 允许对数据的多用户存取

2.4 信息世界与逻辑模型

□ The '70s: 数据库技术飞速发展阶段

- 1970: 关系数据模型 (The Relational Model)
– Developed by E. F. Codd
- 1971: CODASYL Database Task Group Report

2.4 信息世界与逻辑模型

- 1975: 著名的国际会议
 - ACM Special Interest Group on Management Of Data organized first SIGMOD international conference
 - Very Large Data Base Foundation organized first VLDB international conference
- 1976: Entity-Relationship (ER) model

2.4 信息世界与逻辑模型

□ 有影响的研究工作

- System R (IBM)
- INGRES (University of California, Berkeley)
- System 2000 (University of Texas, Austin)
- Socrate Project (University of Grenoble, France)
- ADABAS (Technical University of Darmstadt, W. Germany)

□ 数据库语言

- SQUARE, SEQUEL (SQL), QBE, QUEL

2.4 信息世界与逻辑模型

□ The '80s: 商用关系数据库管理系统的兴起

➤ DBMS在PC平台上的实现

DBASE (Foxbase, Foxpro等) ,
PARADOX

➤ 1983: 商用关系数据库管理系统的出现与流行

DB2, ORACLE, SYBASE, INFORMIX,

➤ 1985:

发布最初的SQL标准

2.4 信息世界与逻辑模型

➤ 其它方向:

- Expert Database Systems
- Object-oriented DBMSs

2.4 信息世界与逻辑模型

□ The '90s

- 专用数据库系统

spatial, temporal, multimedia
data,

- 商用面向对象数据库管理系统

- 对象关系数据库管理系统

□ 新世纪以来

- 数据仓库

- 安全数据库

- XML数据库

- 嵌入式、移动、实时、.....

2.4 信息世界与逻辑模型

□ NoSQL (= Not only SQL)

➤ 泛指非关系型的数据库

– 键值存储数据库

– 列存储数据库

– 文档型数据库

– 图数据库

□ NewSQL

□ 是对各种新的可扩展/高性能数据库的简称

2.4 信息世界与逻辑模型

□ 概念模型与逻辑模型对应关系表

概念模型	E – R模型			EE – R模型	面向对象模型	谓词模型
逻辑模型	层次模型	网状模型	关系模型	对象关系模型	面向对象模型	谓词模型

2.4.1 关系模型与关系模型数据库系统

□ 关系模型 (**Relational model**) 是完全不同于层次模型和网状模型的一种新的逻辑模型

➤ 关系模型的基本数据结构

— 二维表，简称‘表’ (**Table**)

➤ 关系模型的数据操纵

— 是建立在二维表上的操作，它包括对一张表及多张表间的查询，以及对一张表的删除，插入及修改等操作

2.4.1 关系模型与关系模型数据库系统

□ 二维表 (Table)

- 二维表由表框架与元组所组成，表框架由若干个属性组成
- 存放于框架内的每‘一行数据’都被称为‘一个元组’ (Tuple)，或称‘行’ (Row)
- 一张二维表是由一个有 n 个属性的框架及 m 个元组组成

2.4.1 关系模型与关系模型数据库系统

□ 关系

➤ 由行和列组成的二维表格

➤ 关系的约束

1) 同一表中的属性名各不相同

2) 表中的属性与属性的排放次序无关

3) 表中的元组均不相同

4) 表中的元组与元组的排列次序无关

5) 表中的每一分量必须是一个不可分割的基本数据项

2.4.1 关系模型与关系模型数据库系统

学生

学号	姓名	系别	年龄
S1001	张曼英	计算机	19
S1002	李 红	数学	20
S1003	丁一珉	中文	18
S1004	王爱民	计算机	20

课程

课程编号	课程名	主讲教师
C101	C++	T02
C102	OS	T01
C103	DB	T02

2.4.1 关系模型与关系模型数据库系统

□ 关系中的基本概念

- 关系模式
- 关系数据库模式
- 元组
- 关键字（或简称为‘键’-key）
 - 主关键字
 - 外关键字

2.4.1 关系模型与关系模型数据库系统

□ 关系模式

- 一个关系的关系名及其属性名的集合构成该关系的关系模式

□ 关系数据库模式

- 该关系数据库中所有关系的关系模式的集合

□ 元组

- 关系中的每一行

2.4.1 关系模型与关系模型数据库系统

□ 关键字

- 关系中的一个属性集的值能唯一标识关系中的一个元组，且又不含多余的属性值，则称该属性集为该关系的关键字
- 每一个关系都有关键字
- 一个关系也可以有多个关键字，所以关键字也被称为‘**候选关键字**’

2.4.1 关系模型与关系模型数据库系统

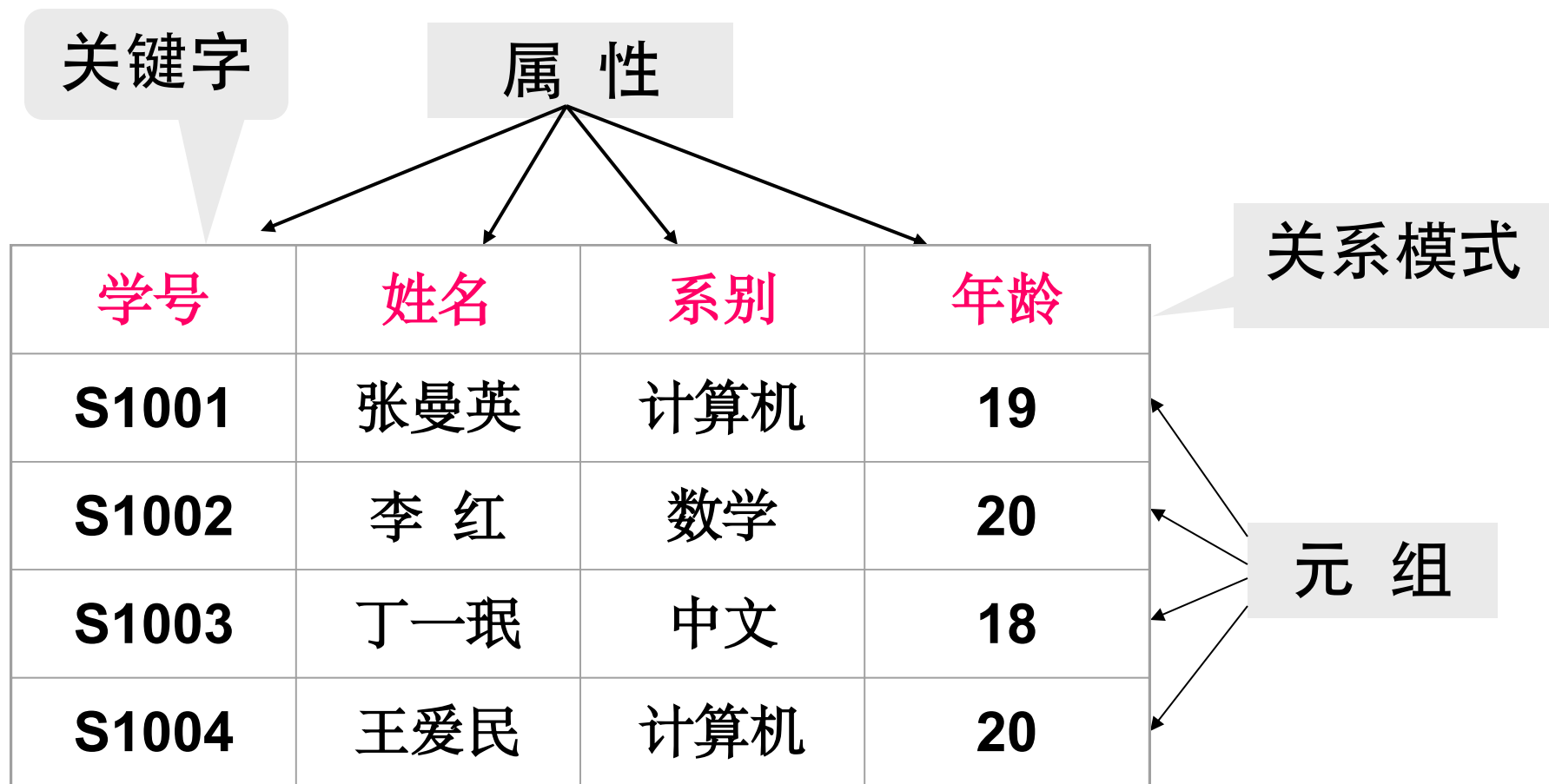
□主关键字

- 可以从关系的候选关键字中选取一个作为该关系的主关键字

□外关键字

- 设关系R中的属性集F，其取值来自于关系S中的主关键字K，则称属性集F是关系R的外关键字
- 关系R和关系S可以是同一个关系

2.4.1 关系模型与关系模型数据库系统



2.4.1 关系模型与关系模型数据库系统

外 关 键 字



学 号	课程编号	成 绩
S1001	C101	85
S1001	C102	91
S1001	C103	95
S1003	C101	90
S1002	C101	76
S1003	C102	87
S1004	C101	88

2.4.1 关系模型与关系模型数据库系统

□ 关系模型上的数据操作

- 关系模型数据操作的对象是‘关系’
- 关系模型数据操作的结果也是一个‘关系’
- 关系模型的五种基本操作：
 - 属性指定
 - 元组选择
 - 关系的合并
 - 元组插入
 - 元组删除

第2章 数据模型

2.1 数据模型的基本概念

2.2 数据模型的四个世界

2.3 概念世界与概念模型

2.4 信息世界与逻辑模型

2.5 计算机世界与物理模型

2.5 计算机世界与物理模型

- 物理模型是面向计算机的模型，它构造数据库系统的物理实现
 - 主要涉及操作系统级文件组织，有时还会涉及到硬件级数据组织

2.5 计算机世界与物理模型

□ 文件系统的组成

- **项** (Item) : 文件系统中最小基本单位, 项内符号是不能继续分割的
- **记录** (Record) : 由若干项组成, 记录内的各项间有内在语义联系
 - 记录有型与值的区别
- **文件** (file) : 记录的集合
 - 一般讲, 一个文件所包含的记录都是同型的
- **文件集** (file set) : 由若干个文件构成

2.5 计算机世界与物理模型

□ 提高文件读写操作效率的方法

- 索引 (Index)
- Hash法
- 集簇 (Cluster)

2.5 计算机世界与物理模型

□ 索引 (Index)

- 将文件中的记录与其物理地址（即磁盘块）间建立一张对应关系表以便于快速查找，这就是索引
- 索引一般也是一个文件。当数据文件中的记录数很大时，索引文件本身也还需要建立索引，这叫二级索引
- 依此类推，可以建立多级索引

2.5 计算机世界与物理模型

□ Hash法

- 一种函数转换法，其主要思想是：通过一个hash函数将要查找的记录转换成该记录所在的物理地址，然后可以直接进行记录的定位读取操作

□ 集簇 (Cluster)

- 在记录查找中往往需要按某项的项值查找，将具有相同或相邻项值的记录聚集在相同磁盘块内或圆柱体内以减少读盘次数，提高查找速度，这被称为集簇

本章小结

- 本章讨论数据模型，它是数据库系统的核心
- 数据模型
 - 基本概念
 - 三个抽象层次上的数据模型
- 概念数据模型
 - 四种概念模型
 - E-R模型，E-R图
 - EE-R模型：IS-A联系，弱实体
- 逻辑数据模型
 - 关系模型