

第5章数据库设计



第5章数据库设计

5.1 数据库设计

--什么是数据库设计？本章的目标

--数据库设计中的抽象

--数据模型与概念模型

5.2 利用E-R模型/IDEF1X模型进行数据库设计

5.3 数据库设计正确性分析

5.4 数据库设计方法与设计过程

5.1 数据库设计

—什么是数据库设计？

➤ 回顾学生选课数据库SCT:

• 教师(教师编号T#, 教师名TName, 所属院系D#, 工资Salary)

Teacher (T# char(3), Tname char(10), D# char(2),
Salary float(2))

• 选课(学生号S#, 课程号C#, 成绩Score)

SC (S# char(8), C# char(3), Score float(1))

• • • • •

✓ 如何设计出来的呢？

✓ 变化一下有无问题呢？

数据库系统

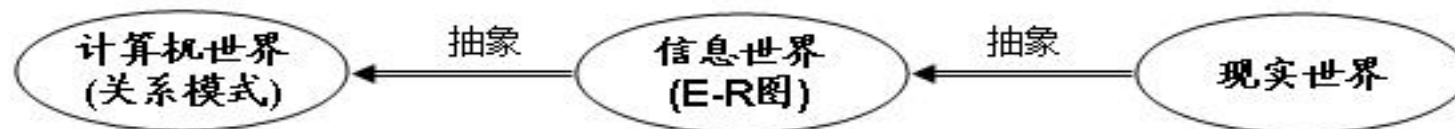
现实世界



数据库系统

数据库设计

数据建模：信息模型



例如定义图书表: Book

```

Create Table Book ( ISBN# char(25) not null, Bname char(40),
Publisher char(40), Version float(4), PubDate datetime );
    
```

数据库表

T4: 图书

| 书号 | 书名 | 出版社 | 版本 | 出版日期 |
|-----------------------|---------|---------|-----|------------|
| 0000000-1-00000-101-0 | 计算机导论 | 清华大学出版社 | 2.0 | 2009-04-01 |
| 0000000-1-00000-102-1 | 计算机组成原理 | 清华大学出版社 | 1.0 | 2009-01-01 |
| 0000000-1-00000-103-0 | 数据库系统 | 科学出版社 | 1.0 | 2009-04-10 |
| 0000000-1-00000-104-0 | 数据库学习教程 | 科学出版社 | 1.0 | 2009-01-01 |
| 0000000-1-00000-105-0 | 数据库设计 | 清华大学出版社 | 1.0 | 2009-01-01 |
| 0000000-1-00000-106-0 | 大学物理 | 机械工业出版社 | 2.0 | 2004-01-01 |
| 0000000-1-00000-107-4 | 大学英语 | 外文出版社 | 2.0 | 2002-06-01 |
| 0000000-1-00000-108-1 | 数据库系统 | 清华大学出版社 | 1.0 | 2009-04-01 |

SQL: 数据库语言

```

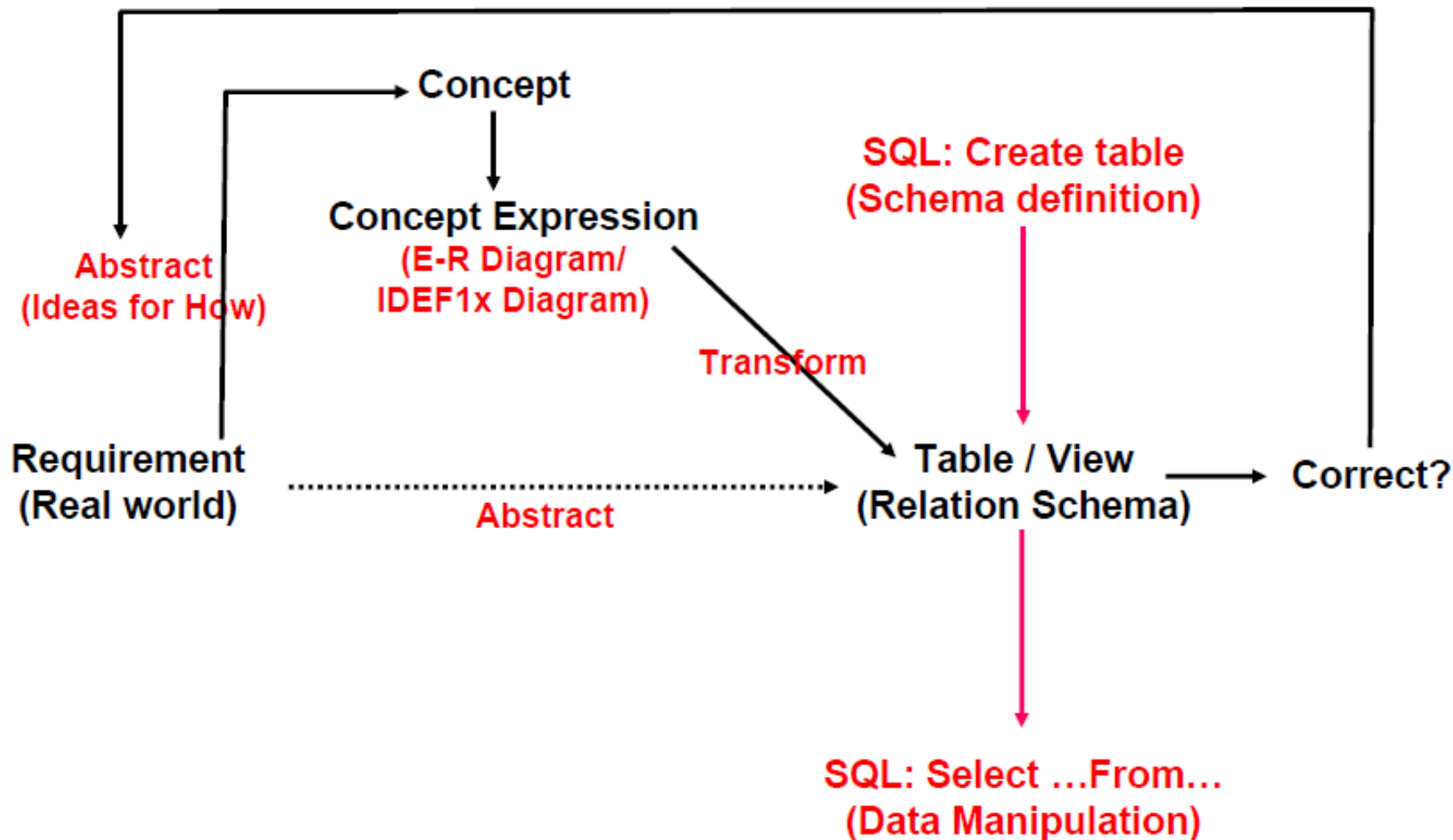
Select Bname From Book
Where Publisher = '哈尔滨出版社' and
Pubdate > '2007-01-01'
Order By PubDate DESC;
    
```

数据库维护与控制

数据库应用程序

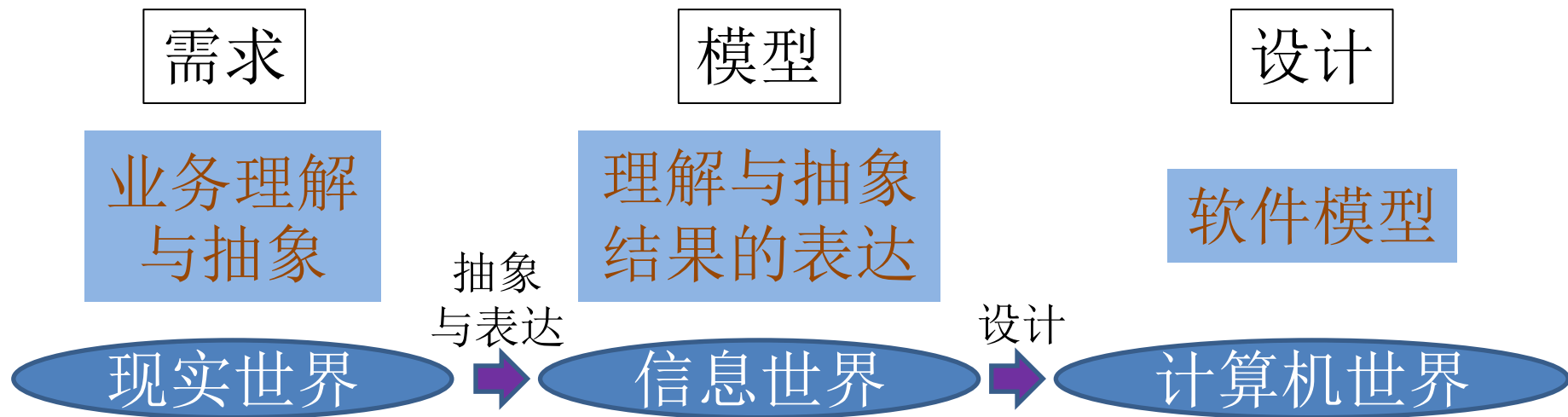


5.1 数据库设计



5.1 数据库设计

- 现实世界的理解与抽象;
- 用元模型的概念表达理解与抽象的结果;
- 所建“模型”是否符合需求“(正确性判断);
- 怎样保证所建”模型“符合需求(过程与方法);



5.1 数据库设计

需求：折扣政策管理

- 不同客户有不同的折扣；
- 不同产品有不同的折扣；
- 不同客户购买不同产品有不同的折扣



客户折扣表

| 客户 | 折扣率 |
|----|-----|
| | |
| | |

产品折扣表

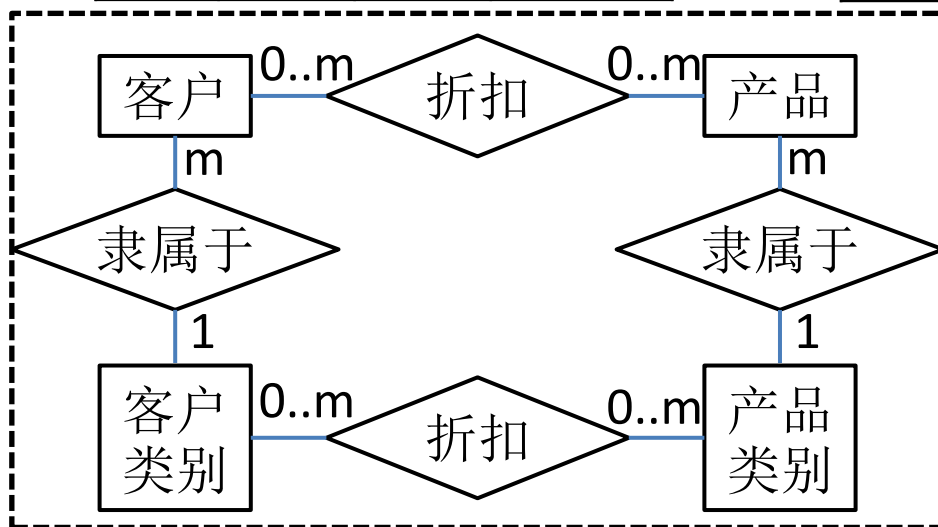
| 产品 | 折扣率 |
|----|-----|
| | |
| | |

客户-产品折扣表

| 序号 | 客户 | 产品 | 折扣率 |
|----|----|----|-----|
| | | | |
| | | | |

客户-产品折扣表

| 客户 | 产品 | 折扣率 |
|----|----|-----|
| | | |
| | | |



5.1 数据库设计

需求：折扣
政策管理

- 不同客户有不同的折扣；
- 不同产品有不同的折扣；
- 不同客户购买不同产品有不同的折扣



客户-产品折扣表

| 序号 | 客户 | 产品 | 折扣率 |
|----|----|----|-----|
| | | | |
| | | | |



客户折扣表

| 客户 | 折扣率 |
|----|-----|
| | |
| | |

产品折扣表

| 产品 | 折扣率 |
|----|-----|
| | |
| | |

客户-产品折扣表

| 客户 | 产品 | 折扣率 |
|----|----|-----|
| | | |
| | | |

客户-产品折扣表

| 序号 | 客户类别 | 产品类别 | 折扣率 |
|----|------|------|-----|
| | | | |
| | | | |

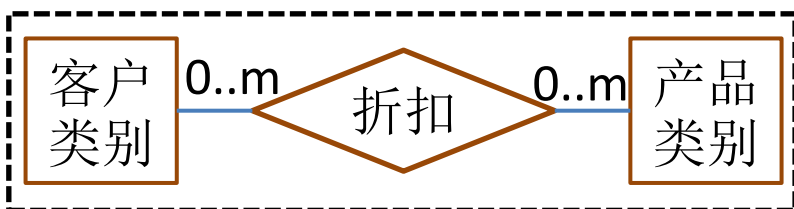


5.1 数据库设计

需求：折扣政策管理

- 不同客户有不同的折扣；
- 不同产品有不同的折扣；
- 不同客户购买不同产品有不同的折扣

需求



模型

客户折扣表

| 客户 | 折扣率 |
|----|-----|
| | |
| | |

产品折扣表

| 产品 | 折扣率 |
|----|-----|
| | |
| | |

客户-产品折扣表

| 客户 | 产品 | 折扣率 |
|----|----|-----|
| | | |
| | | |

客户-产品折扣表

| 序号 | 客户类别 | 产品类别 | 折扣率 |
|----|------|------|-----|
| | | | |
| | | | |

设计



5.1 数据库设计

—什么是数据库设计？

客户信息表

| 客户姓名 | 客户地址 | 电话 | 联系人 |
|--------|------|----|-----|
| 客户已购产品 | | | |
| 客户现有项目 | 项目名称 | | |
| | 项目名称 | | |
| 跟踪座谈情况 | 第一次 | | |
| | 第二次 | | |

产品/E5

| |
|------|
| 产品编码 |
| |

客户已购产品/E2

| |
|----------|
| 客户代码(fk) |
| 产品编号 |
| 产品编码(fk) |

客户/E1

| |
|------|
| 客户代码 |
| |

客户跟踪/E4

| |
|------------|
| 客户代码(fk) |
| 客户跟踪号 |
| 客户项目编号(fk) |

客户项目/E3

| |
|----------|
| 客户代码(fk) |
| 客户项目编号 |
| 产品编码(fk) |



5.1 数据库设计

—什么是数据库设计？

出库单

出库类别____ 出库日期____ 外协单位____
领料单编号____ 领料部门____ 领料人____
库房____ 库管人____ 备注____
限额审核____ 部门制单____ 部门审核____

| 序号 | 工程号 | 领料条码 | 领料名称 | 领料规格 | 申领数量 | 实发数量 | 单位 | 单价 | 金额 |
|----|-----|------|------|------|------|------|----|----|----|
| | | | | | | | | | |
| | | | | | | | | | |

出库单/E1

| |
|------|
| 出库编号 |
| |



出库明细/E2

| |
|----------|
| 出库编号(fk) |
| 条目序号 |
| 物料编码(fk) |

5.1 数据库设计

——什么是数据库设计？

➤ 通过对现实世界的分析，设计出数据库的模式

- ❑ 现实中的卡片、单据、表格、报表……
- ❑ 用各种各样形式管理的数据，如何存储成数据库，由计算机管理？
- ❑ 信息之间有哪些本质联系，如何分析清楚？
- ❑ 如何更准确地更正确地反映和描述现实世界(关系设计的一些理论)
- ❑ 如何用简单易懂的表示方法，表征出复杂的信息联系，使所有相关人员都清楚(E-R图/IDEF1X模型)
- ❑ 本质上，是信息的描述与刻画……

5.1 数据库设计

——本章的目标

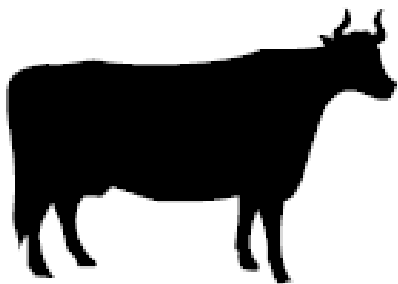
- 理解数据库设计
- 扎实地理解并掌握E-R模型/IDEF1X模型：数据库设计的重要工具
 - ▣ 能够绘制E-R图/IDEF1X图, 而且绘制得正确
- 能够用E-R/IDEF1X模型准确理解现实世界并进行数据库设计
 - ▣ 理解现实世界并进行抽象的能力, 理解并抽象得正确
- 能够分析数据库设计的正确性

5.1 数据库设计

—数据库设计中的抽象

➤ 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)
- ❑ 也可以通过对现实进行抽象, 形成信息(抽象)



“牛”



5.1 数据库设计

—数据库设计中的抽象

➤ 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
- “公牛”而不是母牛
- 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往往体现在信息(及其联系)的正确分析上，体现在理解现实世界能力的高低

5.1 数据库设计

—数据库设计中的抽象

- 现实世界 ==> 信息世界 ==> 计算机世界
- 现实(客观存在) ==> 抽象/描述(概念/观念)
==> 计算机中(用计算机实现)



描述为 ==> “牛”_(观念) 描述为 ==> 0100 1001_(计算机)



5.1 数据库设计

—数据库设计中的抽象

➤ 基本的抽象示例：“型”与“值”的抽象

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

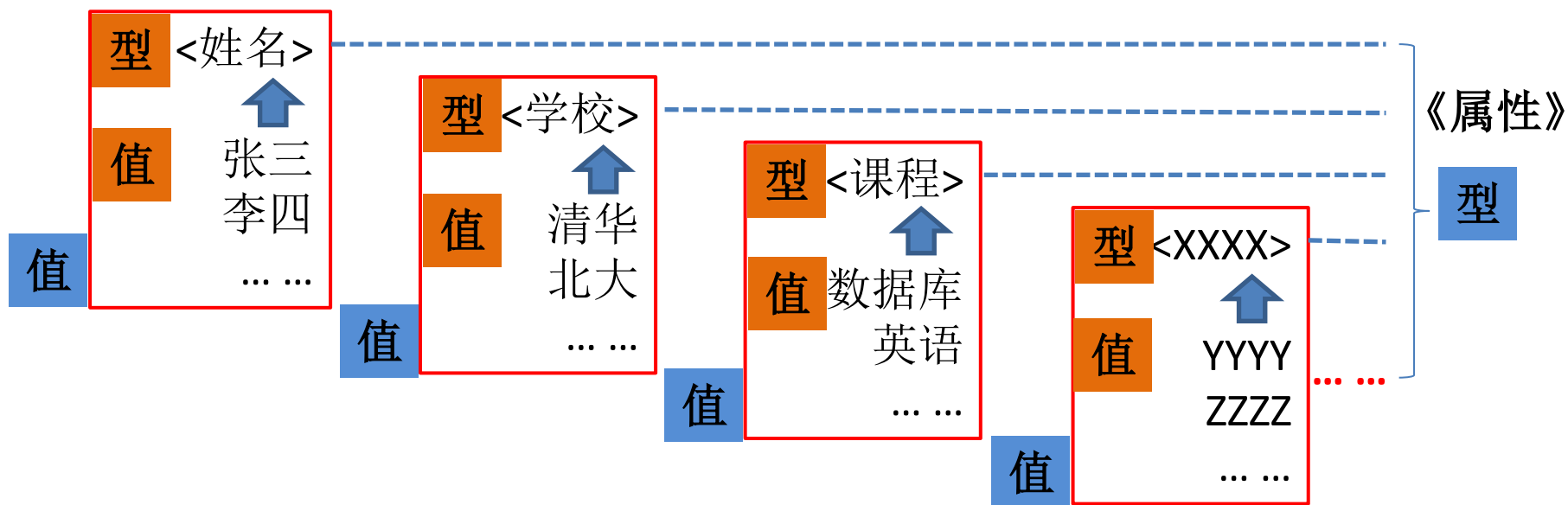
型

5.1 数据库设计

—数据库设计中的抽象

➤ 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



5.1 数据库设计

—数据库设计中的抽象

➤ 现实层(客观存在)

==> 抽象层(观念世界/信息世界, 描述现实世界的一种观点)

==> ... (信息世界的若干层抽象) ...

==> 逻辑层(计算机世界: 独立于物理设备)

==> 物理层(计算机世界: 不同物理设备的具体实现)

5.1 数据库设计

—数据库设计中的抽象

- 越抽象，语义信息越少，概括性越高，越反映共性信息，表征的范围越大
- 检验抽象正确性的方法：能够依据现实抽象出来（抽象化），同时也能够依据抽象的信息和抽象规则还原为被抽象对象（具体化）

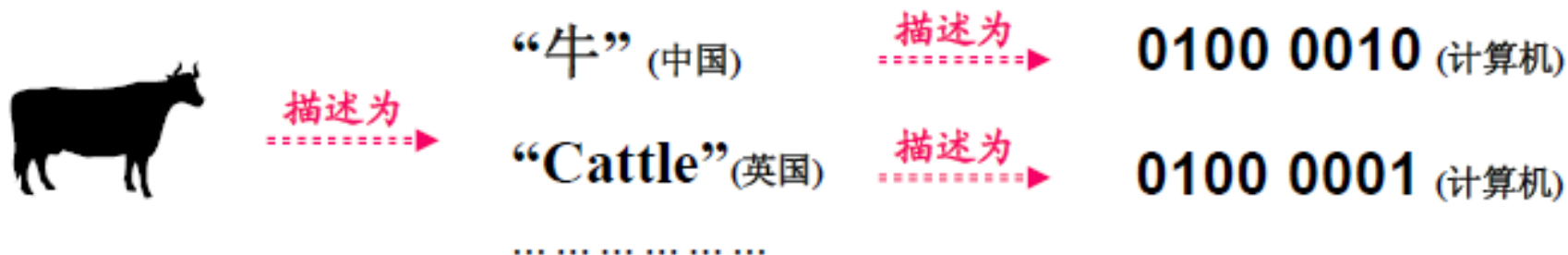
现实 $\xrightarrow[A]{\text{抽象}}$ 抽象的信息 $\xrightarrow[B]{\text{还原}}$ 的规则还原成的现实 A' 用抽象的信息按抽象

- 理论上： $A = A'$ ；
- 现实上： $A \neq A'$ ，差距在哪？

5.1 数据库设计

—数据库设计中的抽象

- 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

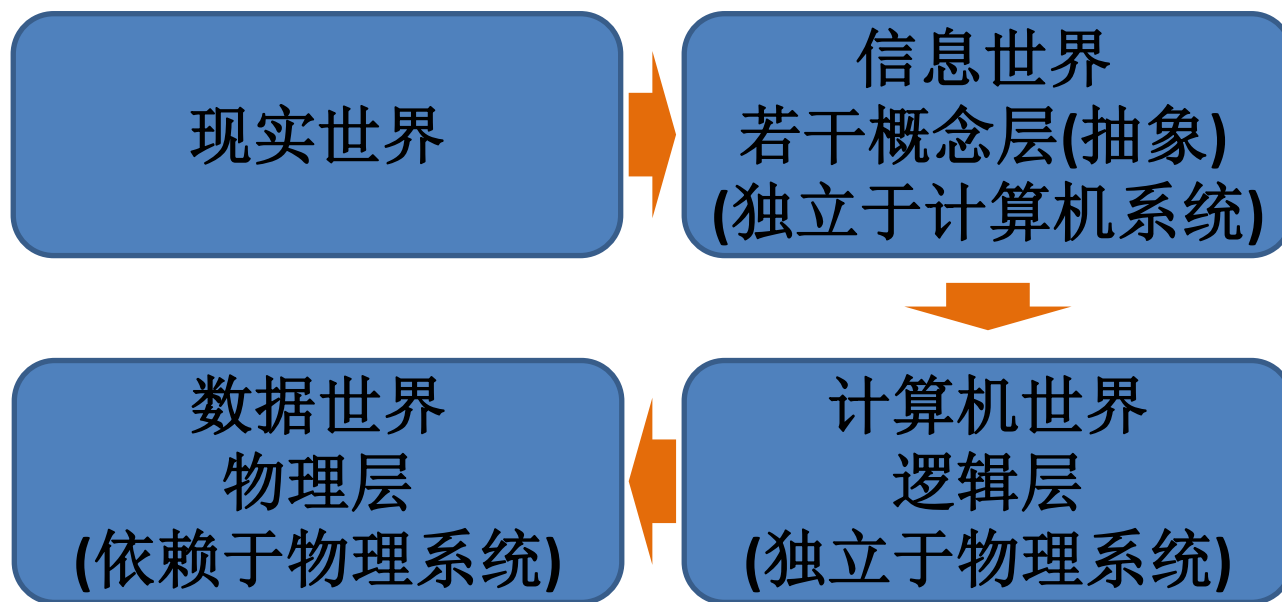


- 现实的抽象与描述需要遵循统一的数据模型：统一的概念与统一的表示方法
- 统一是为了信息交流、信息共享

5.1 数据库设计

—数据库设计中的抽象

- 数据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、计算机世界、数据世界的模型。

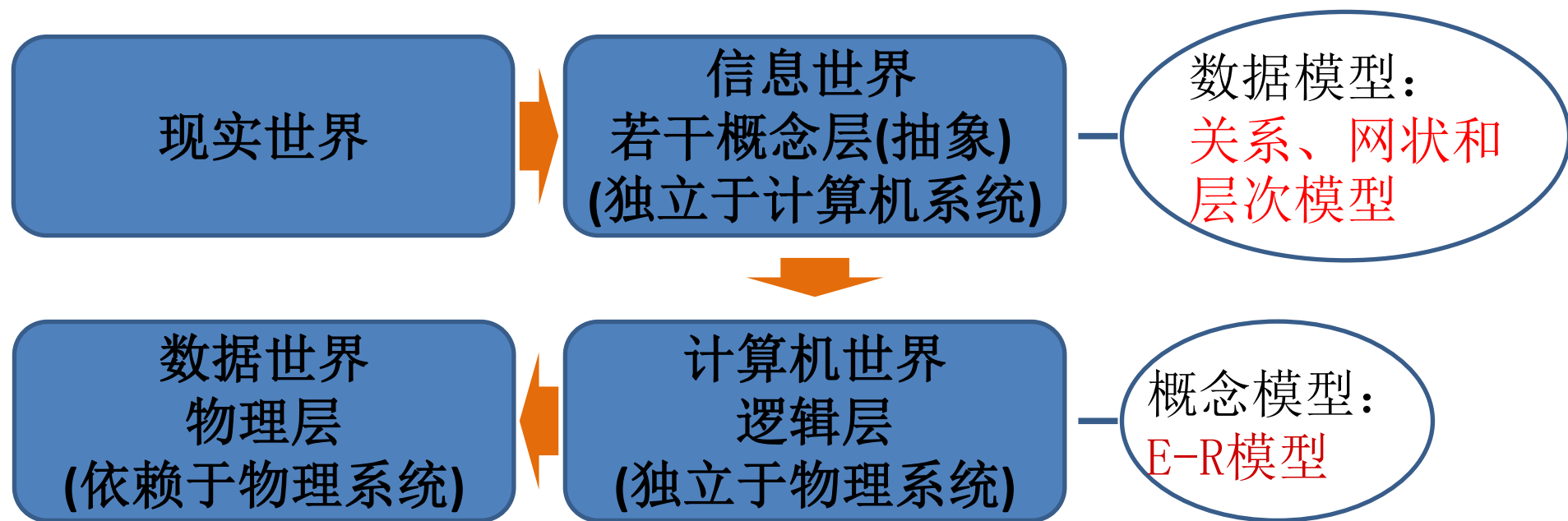


5.1 数据库设计

—数据库设计中的抽象

➤ 数据模型分不同的层次：

- ▣ 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**



第5章数据库设计

5.1 数据库设计

5.2 利用E-R模型/IDEF1X模型进行数据库设计

- E-R模型的基本概念

- E-R模型的几种图示方法

- 数据库设计实践

- E-R图向关系模型的转换

5.3 数据库设计正确性分析

5.4 数据库设计方法与设计过程

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

➤ 历史

- E-R模型: Entity-Relationship Model (实体-联系模型)
- 1976年, P. P. S. Chen提出E-R模型, 用E-R图来描述概念模型

➤ 观点

- 世界是由一组称作实体的基本对象和这些对象之间的联系构成的

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

➤ E-R图模型给出了一组概念，用这组概念可以描述信息世界

• 抽象的概念(语义较少)

□ 实体 □ 属性 □ 联系 □ 关键字/码

• 对应问题领域的概念(给抽象概念赋予一定语义的新的概念)

□ 学生(学号,姓名,...)

□ 课程(课程号,课程名,...)

□ 教师(教师编号, 教师名,...)

□ 任课(教师编号,课程号,...)

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

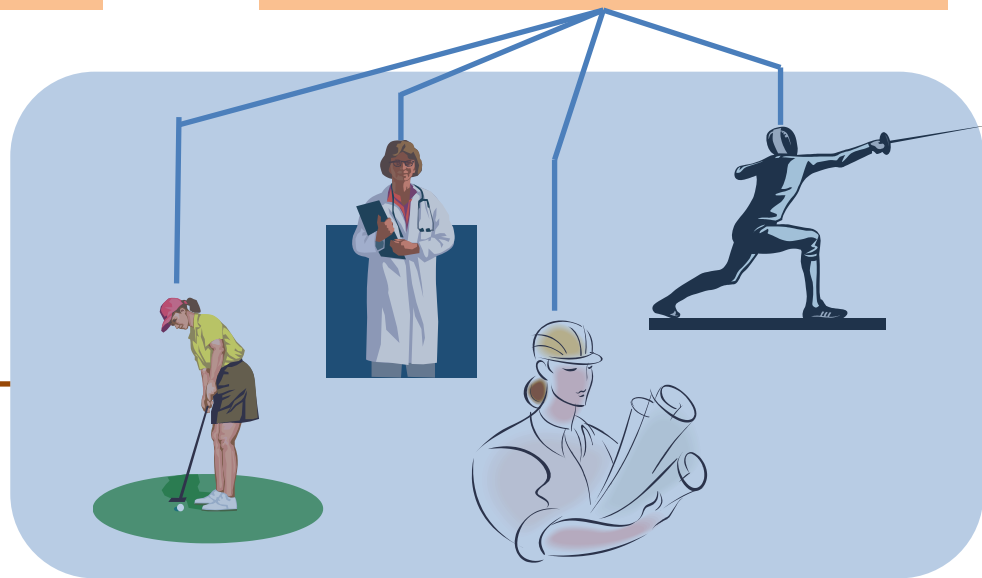
- 实体：客观存在并可相互区分的事物
- 实体有类(实体)和个体(实体的实例)的概念

一类实体(实体集--简称实体)

一个个个体(称为实例)

一类实体具有相同或相似的特性

女士



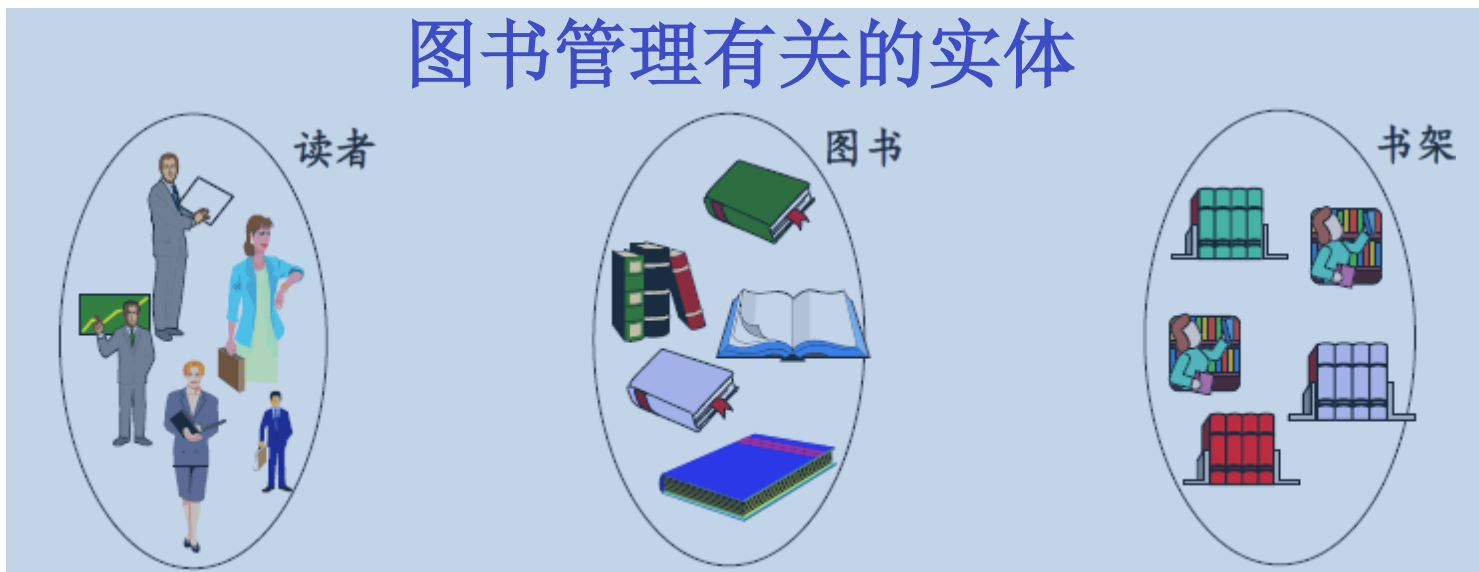
5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

➤ E-R图设计首先要找出问题领域的实体

即找出有哪些类/实体能够用一个一个、一件件、一串串等重叠量词形容的，而不是一个、一件…

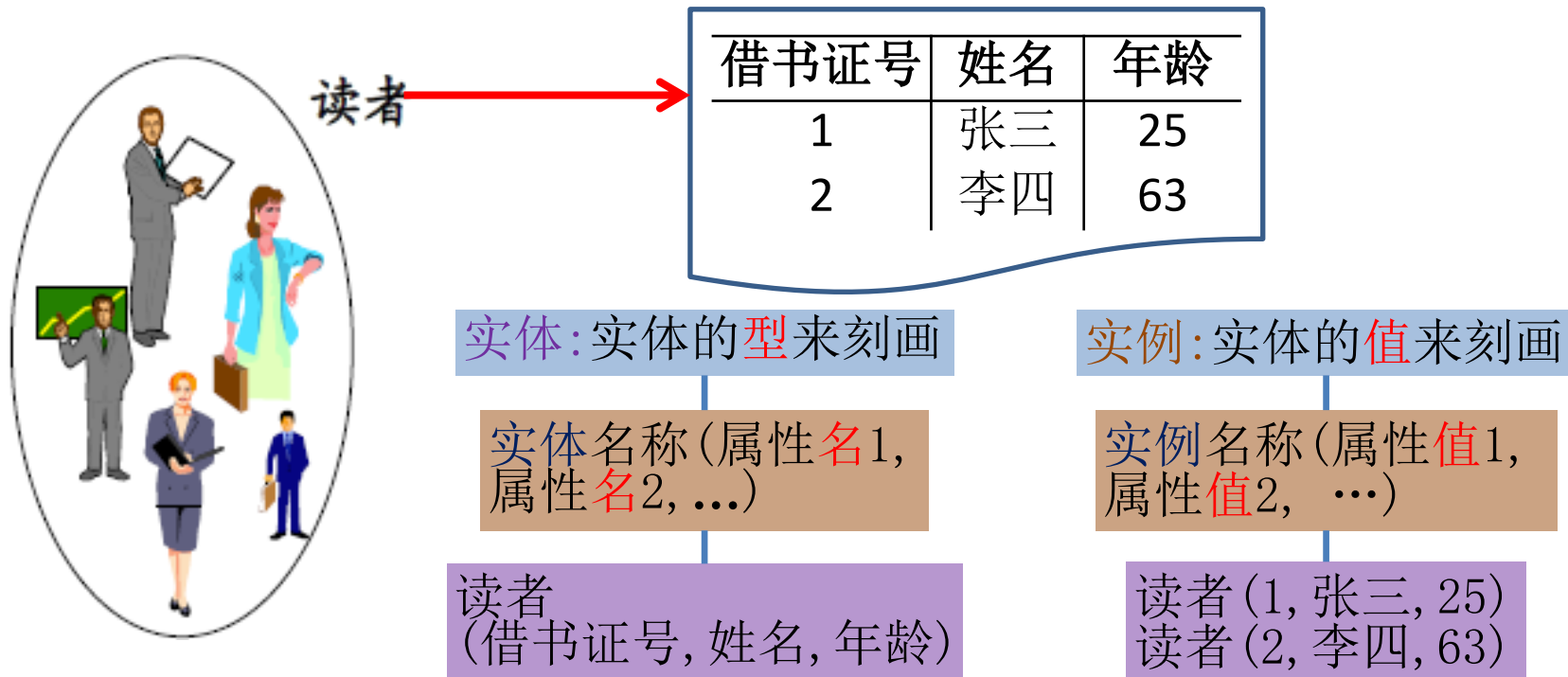
图书管理有关的实体



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

- 实体用属性来描述
- 属性，实体所具有的某一方面特性



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

➤ 实体中的每一实例如何区分？

特殊并关键的属性：关键字

➤ 关键字/码，实体中能够用其值唯一区分开每一实例的属性或属性组合



读者

| 借书证号 | 姓名 | 年龄 |
|------|----|----|
| 1 | 张三 | 25 |
| 2 | 李四 | 63 |

借书证号是唯一的，其他属性均有重复的，所以借书证号是关键字，其他不是... ..

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

➤ 属性还有很多类型，注意区分：

□ 单一属性与复合属性，

✓ 复合属性示例：家庭住址包含省份，详细住址

✓ 在关系模型中，复合属性一定要转化为单一属性(关系的第1范式)

□ 单值属性和多值属性：每个实例的该属性值是一个还是多个

✓ 多值属性示例：电话号码，一个人可能有多个电话号码

✓ 在关系模型中，多值属性一定要转化为单值属性(关系的第1范式)

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

➤ 属性还有很多类型，注意区分：

- ▣ 可空值属性和非空值属性：每个实例的该属性值可以是或不能是空值
- ▣ 导出属性
 - ✓ 由其他属性计算而得
 - ✓ 例如由“出生年份” 可以得出“年龄”

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

➤ 实体描述的另外的例子



图书

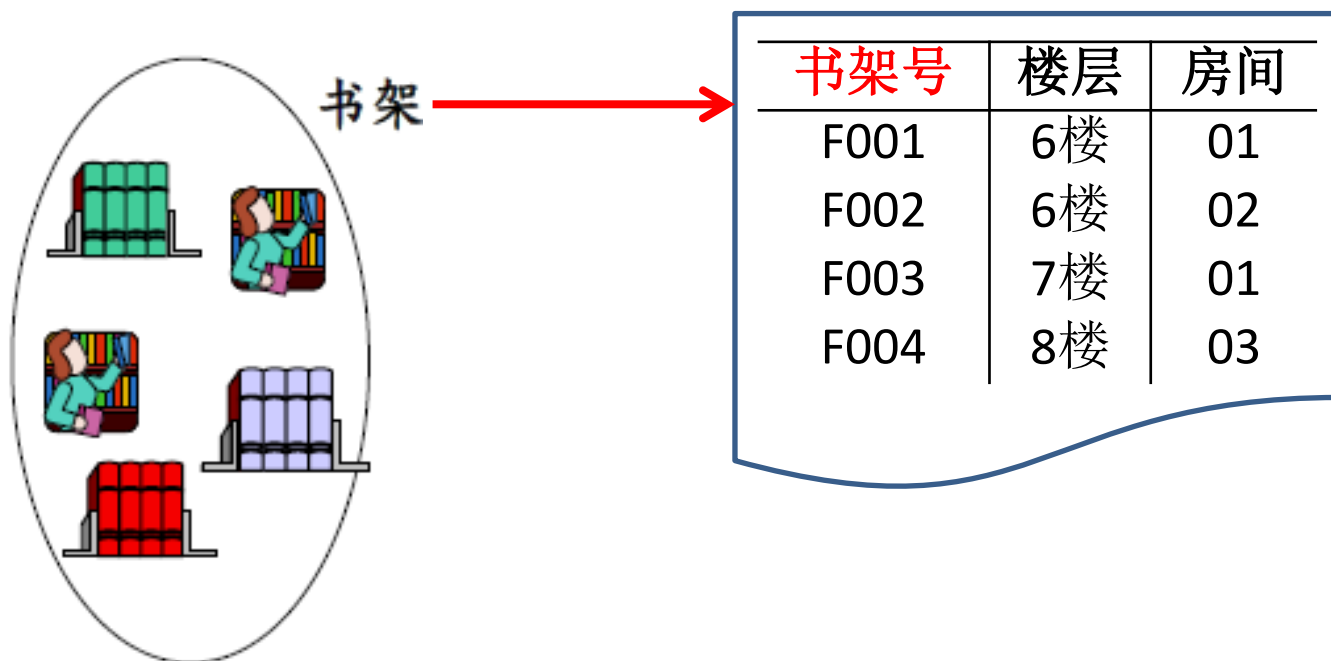


| 书号 | 书名 | 出版社 |
|------|-----|-----|
| A001 | 数据库 | 机械 |
| A002 | 代数 | 清华 |
| A003 | 英语 | 人大 |
| A004 | 计算机 | 吉林 |

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

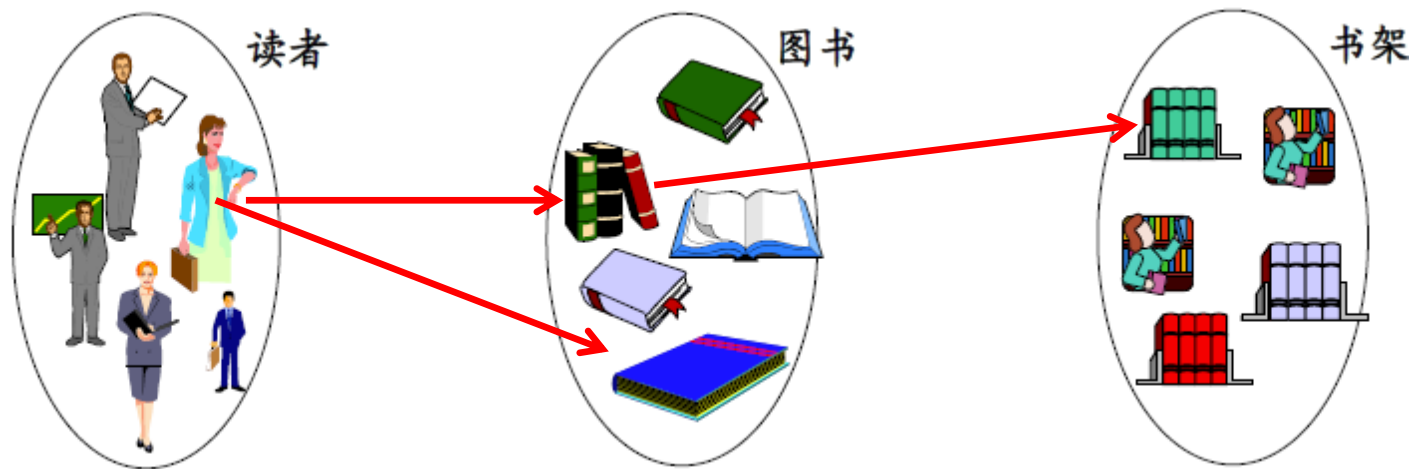
➤ 实体描述的另外的例子



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

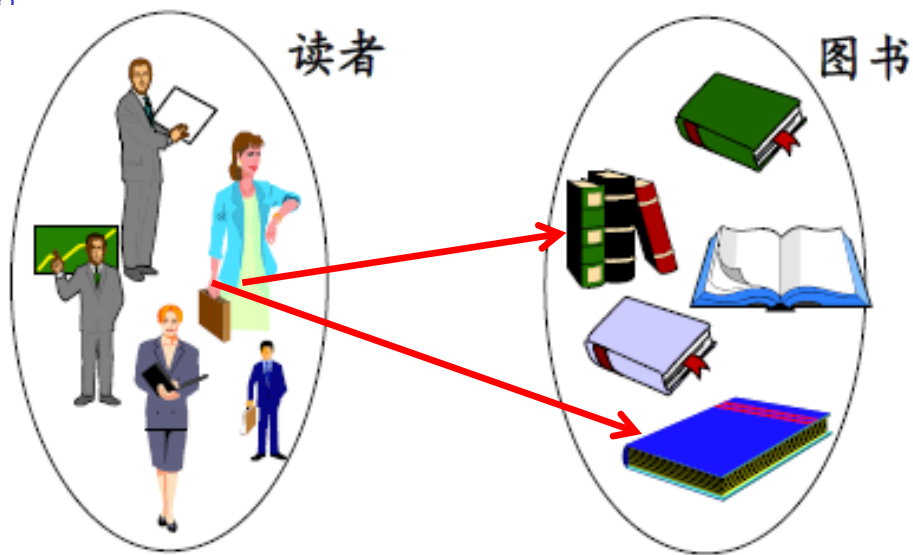
- 实体之间是有联系的
- 联系，是指一个实体中的实例和其他实体中实例之间所可能发生的联系
 - 例如，哪位《读者》借阅了哪本《图书》，
 - 哪本《图书》放在哪个《书架》上



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

- 参与发生联系的实体的数目，称为联系的度或元。
- 联系有一元联系、二元联系和多元联系。
- 现实世界中大多数联系是二元的，即两个不同实体之间的联系。



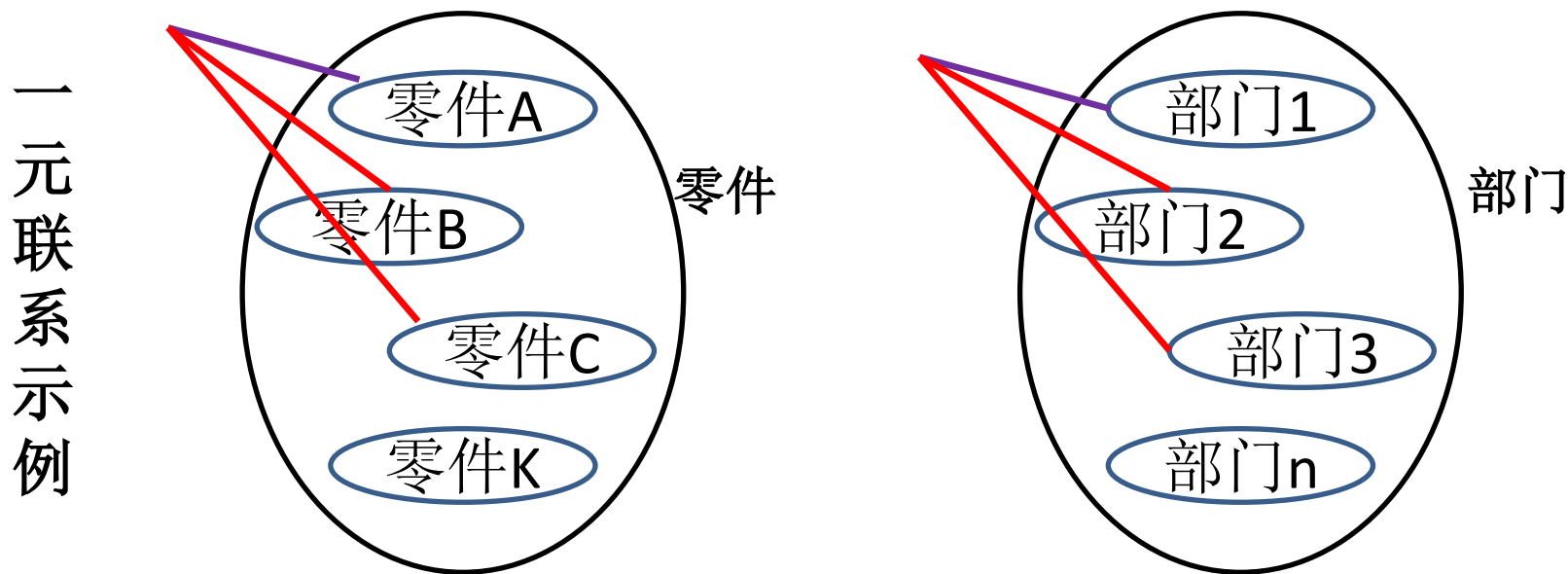
5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

➤ 一元联系示例

□ 零件A由零件B和零件C装配构成

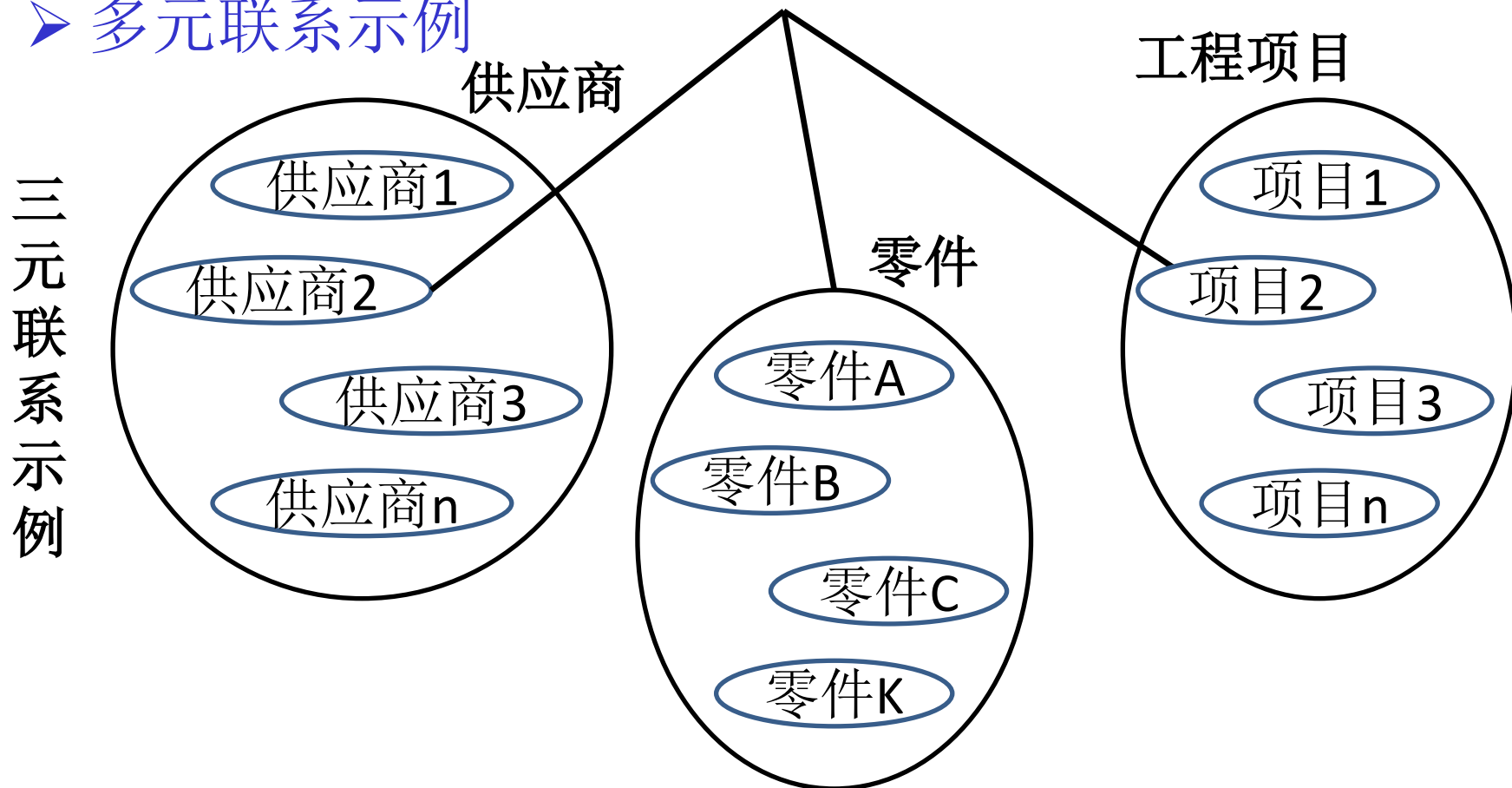
□ 部门1下设两个子部门部门2和部门3



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

➤ 多元联系示例

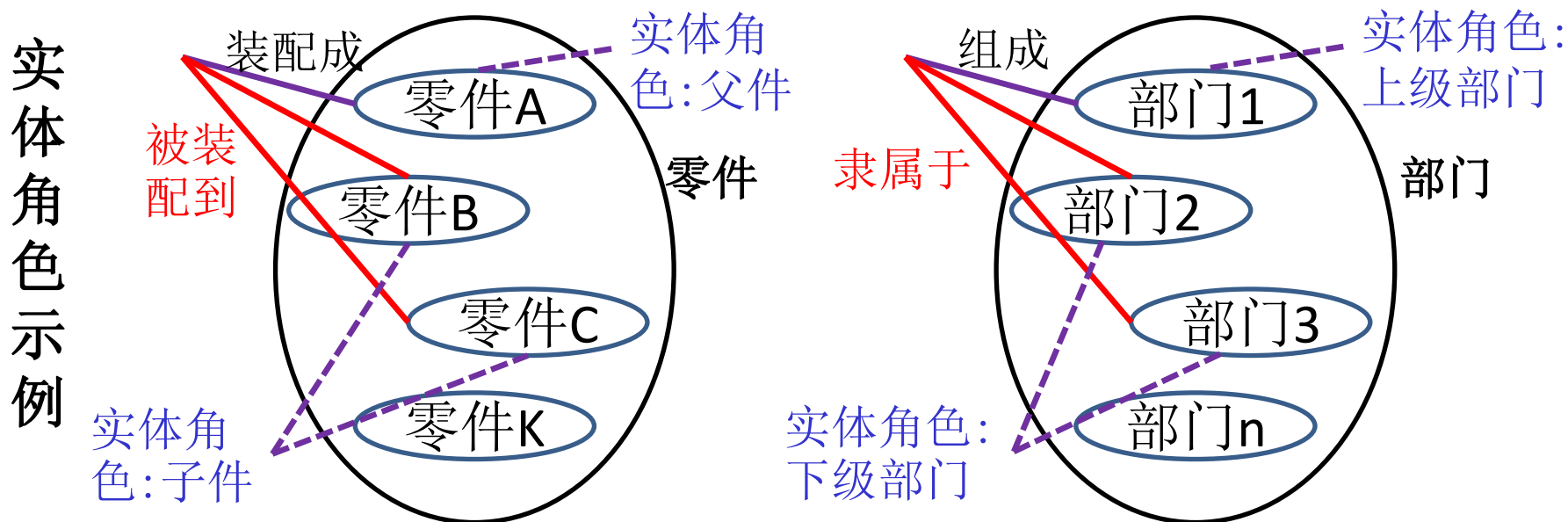


5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

► 角色 (Role)

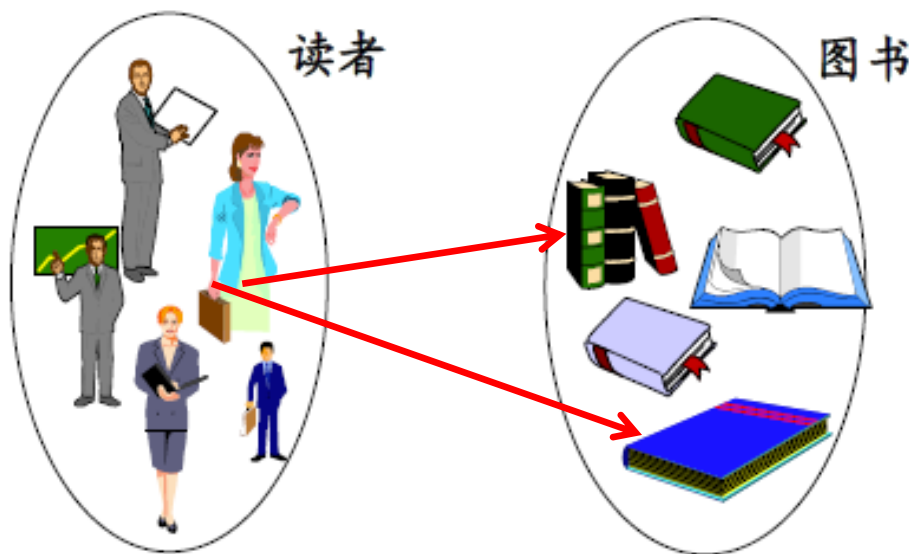
- 实体在联系中的作用称为实体的角色
- 当同一个实体不止一次参与一个联系时，为区别各实体的参与联系的方式，需要显式指明其角色



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

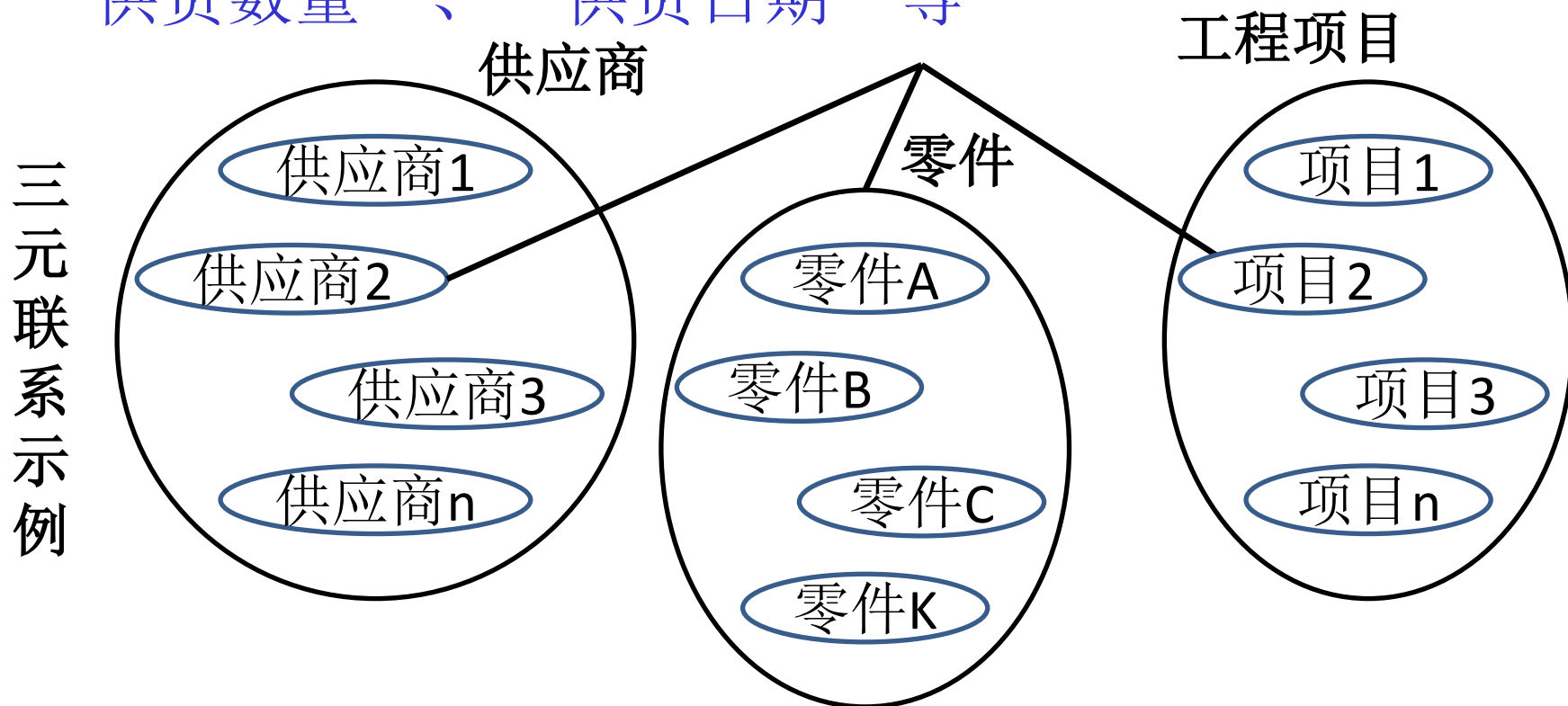
- 联系的特性也可以用属性来刻画
- 例如，下图二元联系的属性可以有：
“借阅时间”、“归还时间”等



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

- 再如，下图三元联系的属性可以有：
“供货数量”、“供货日期”等



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

- 实体之间的联系有很多种类型
- 我们首先要清楚二元联系的：一对一、一对多和多对多联系
 - 一对一联系(1:1)
 - 一个“经理”只管理一个“商店”，
 - 一个“商店”只能有一个“经理”
 - 一对多联系(1:m和m:1)
 - 一个“画家”可以绘制多幅“作品”，
 - 一幅“作品”只能由一个“画家”来完成
 - 多对多联系(m:n)
 - 一位同学可以选学多门课程，一门课程可由多个人来选学

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

➤ 联系的基数:

实体实例之间的联系的数量，即一个实体的实例通过一个联系能与另一实体中相关联的实例的数目。

➤ 常见的映射基数如上，

有一对一的（1:1），

一对多的（1:m），

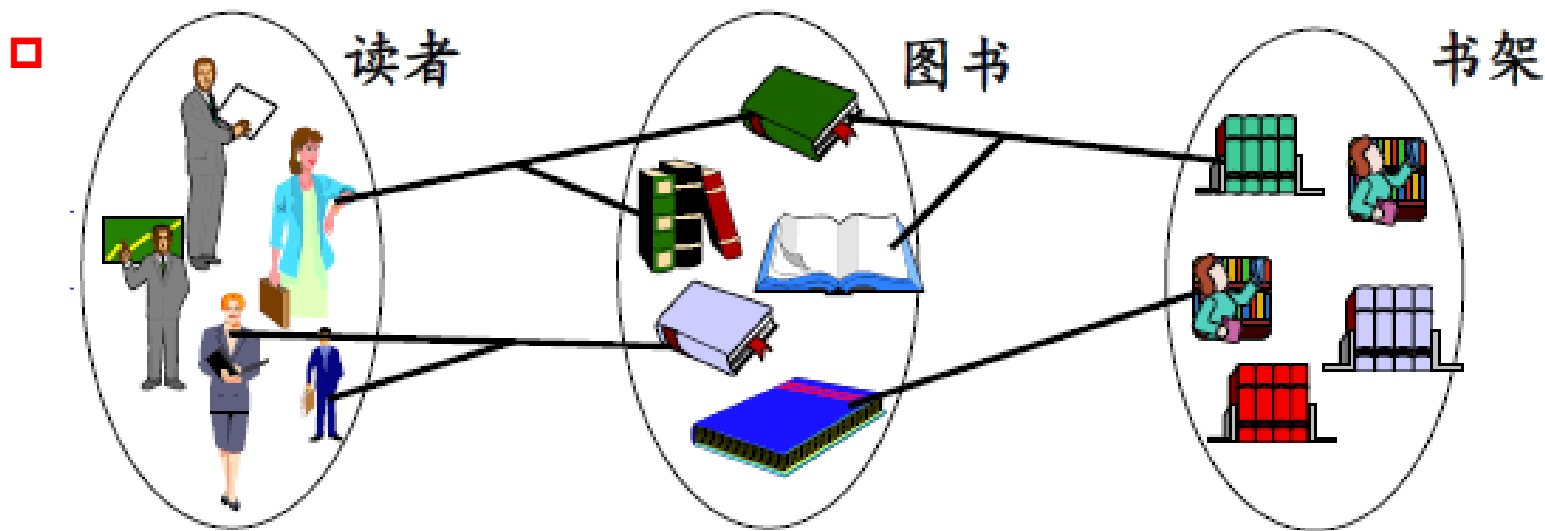
多对多的（m:n）

几种情况

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

- 进一步，联系的基数还要区分是0个、1个、不定数目的多个还是固定数目的多个(即，对每个实体的实例而言是否必须存在)
- 通常以实体参与联系的最小基数和最大基数来标记



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

- 进一步，联系的基数还要区分是0个、1个、不定数目的多个还是固定数目的多个(即，对每个实体的实例而言是否必须存在)
- 通常以实体参与联系的最小基数和最大基数来标记
 - ▣ “书架”参与“存放图书”的联系基数为(0..m)，而“图书”参与此联系的基数为(1..1)
一个“书架”可以存放0或多本“图书”，一本“图书”只能存放在1个“书架”

5.2 利用E-R模型/IDEF1X模型进行数据库设计

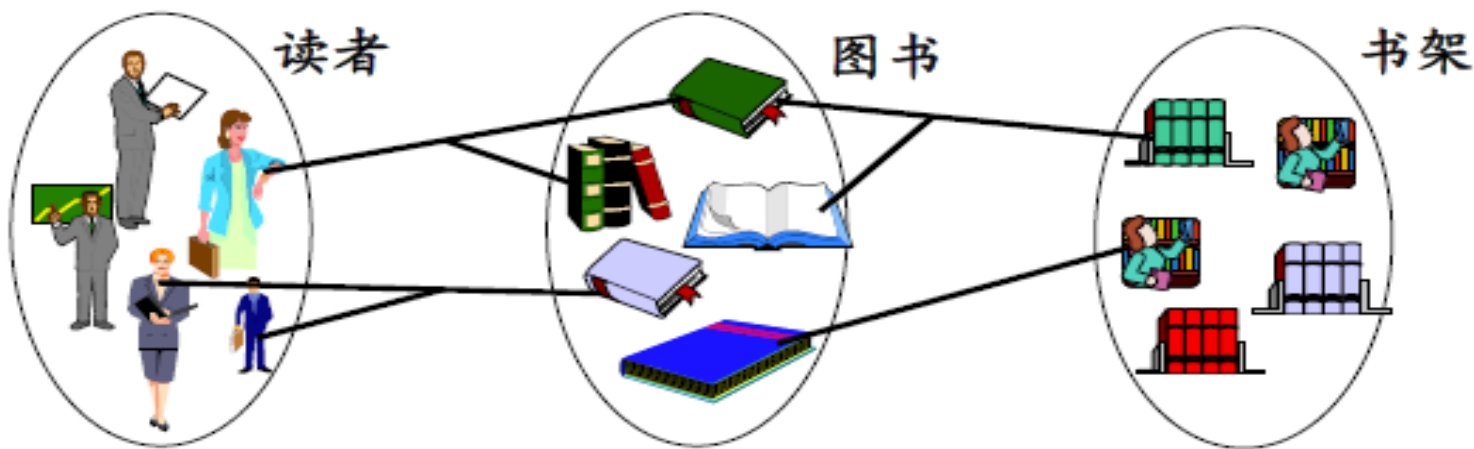
--E-R模型的基本概念

➤ 完全参与联系：

即该端实例至少有一个参与到联系中，最小基数为1(1..m)。

➤ 部分参与联系：

即该端实例可以不参与联系，最小基数为0(0..m)



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的基本概念

➤ 为什么需要区分完全参与联系和部分参与联系？

➤ 例如：

- 学生-选课信息
- 职工-子女信息

➤ 父子实体间是伴随关系

选课记录和职工子女的信息参与联系时就是“完全参与”。

✓ 学生被开除, 其选课记录也应该一同删除。

✓ 人事管理系统中, 职工子女的信息就是以职工的存在为前提的。

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法

- Chen 方法
- Crow's Foot方法
- IDEF1X方法

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：Chen方法

- 矩形框：表示实体
- 菱形框：表示联系
- 椭圆：表示属性

▣ 双线椭圆：表示多值属性

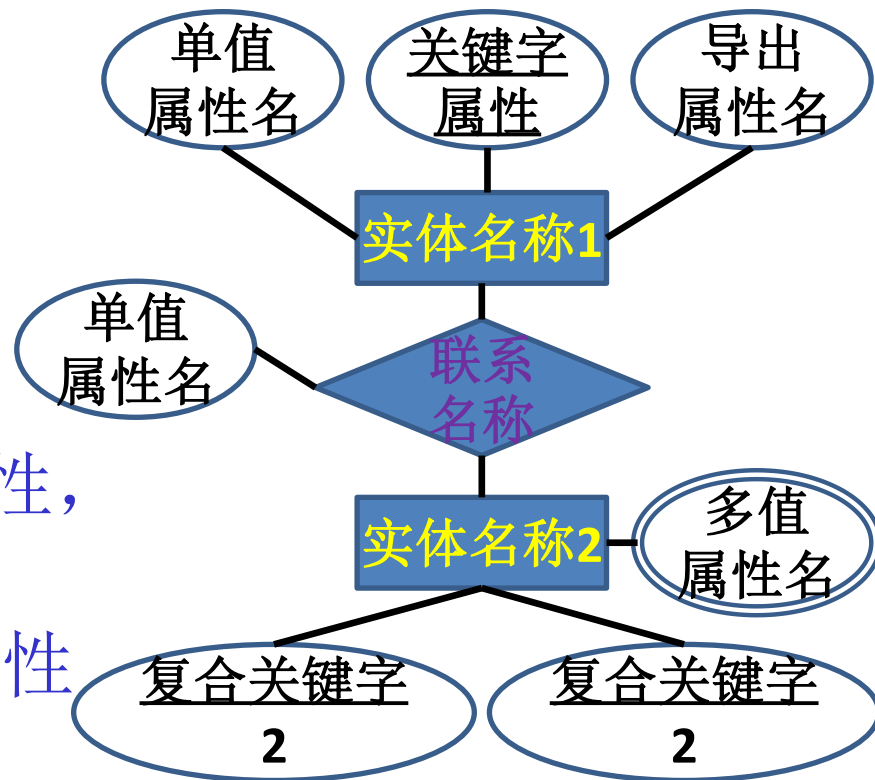
▣ 虚线椭圆：表示导出属性

- 直线：用于连接实体和属性，以及实体与联系

- 下划线：标识关键字/码属性

- 下划线下标有相同数字：

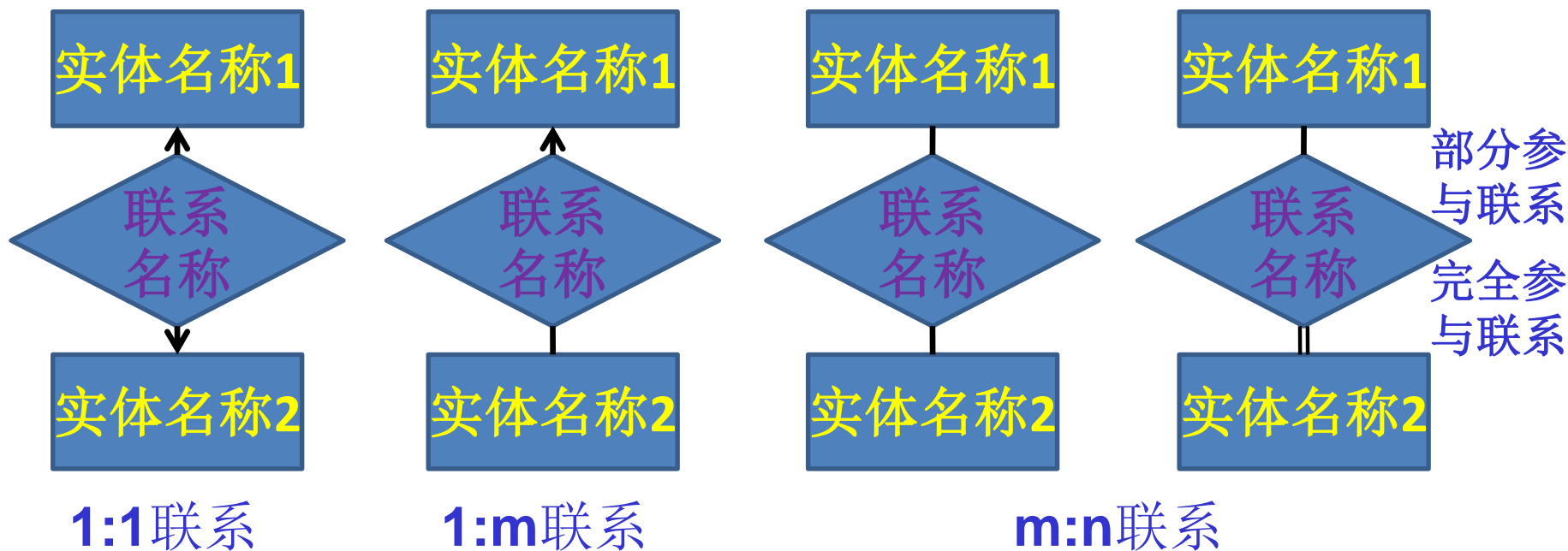
标识一个关键字是由多个属性来构成的



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：Chen方法

- 箭头直线：由联系指向1端
- 无箭头直线：由联系指向多端
- 单直线：部分参与联系；双直线：完全参与联系；

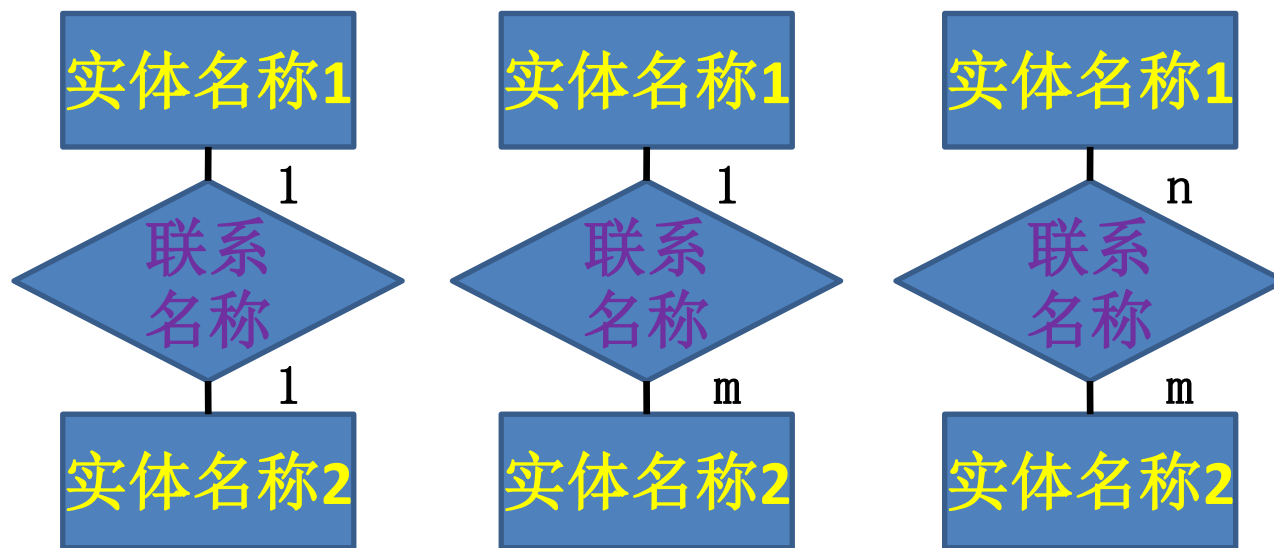


5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：Chen方法

或者采用：

- 直线旁标1：由联系指向1端
- 直线旁标m或n：由联系指向多端

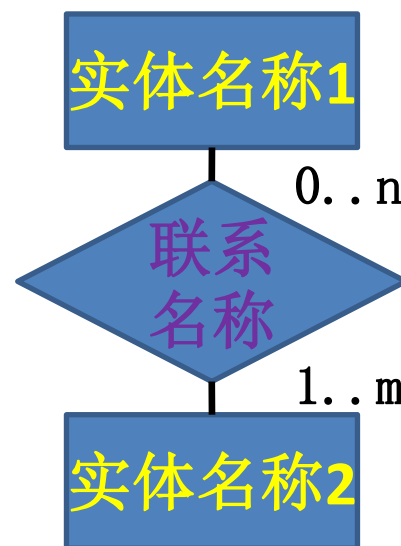


5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：Chen方法

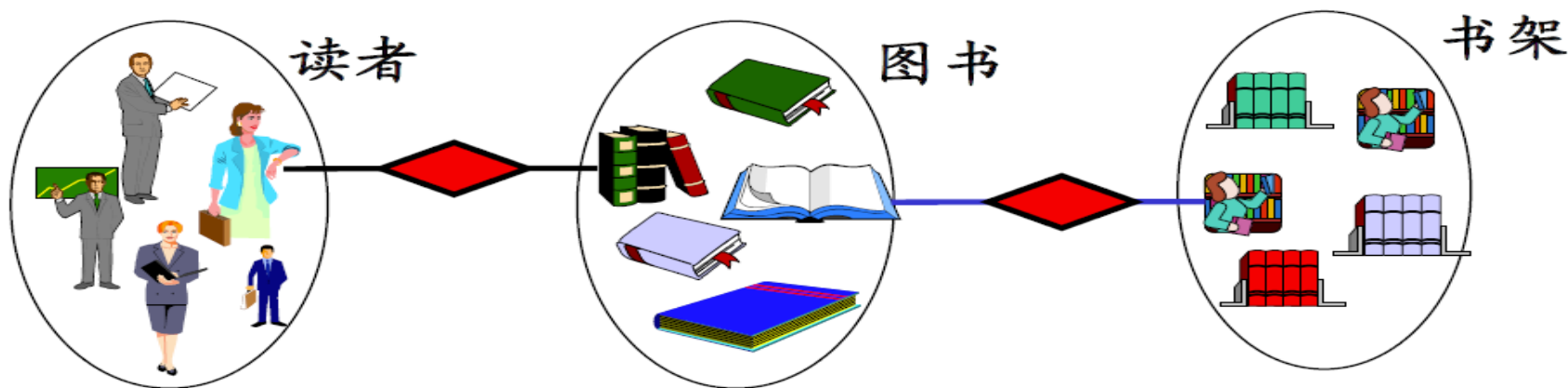
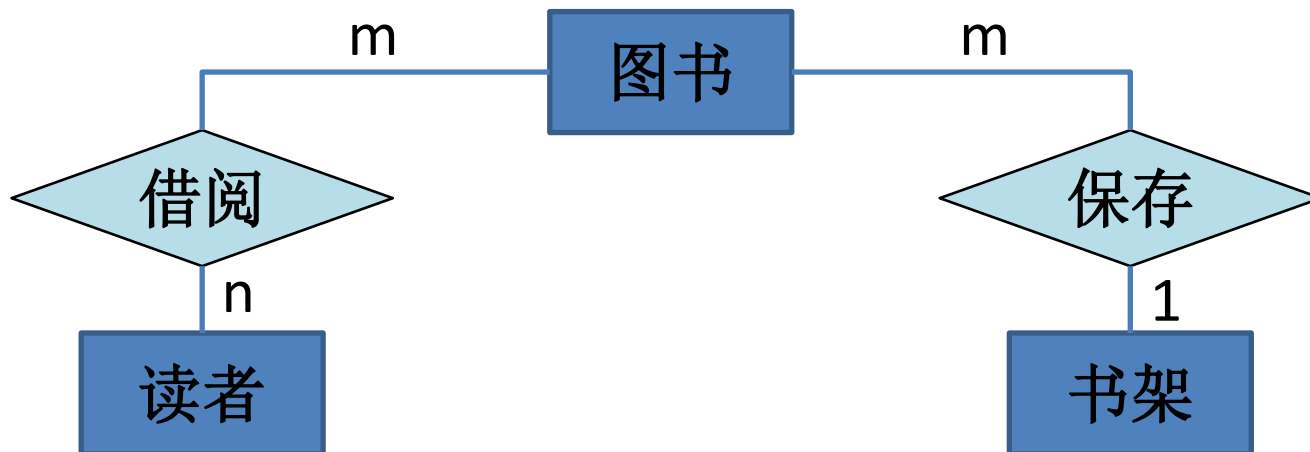
或者采用：

- 直线旁标1..1, 0..1, 1..m, 0..m
完全参与联系还是部分参与联系。
- 0开始的，为部分参与联系，
即该端实例可以不参与联系。
- 1开始的，为完全联系，
即该端实例至少有一个参与到联系中；



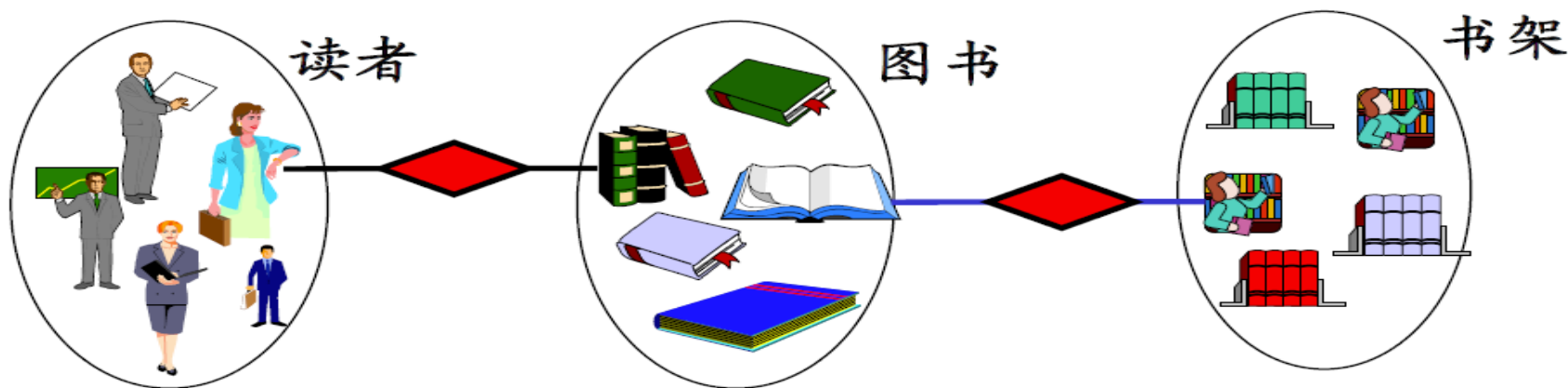
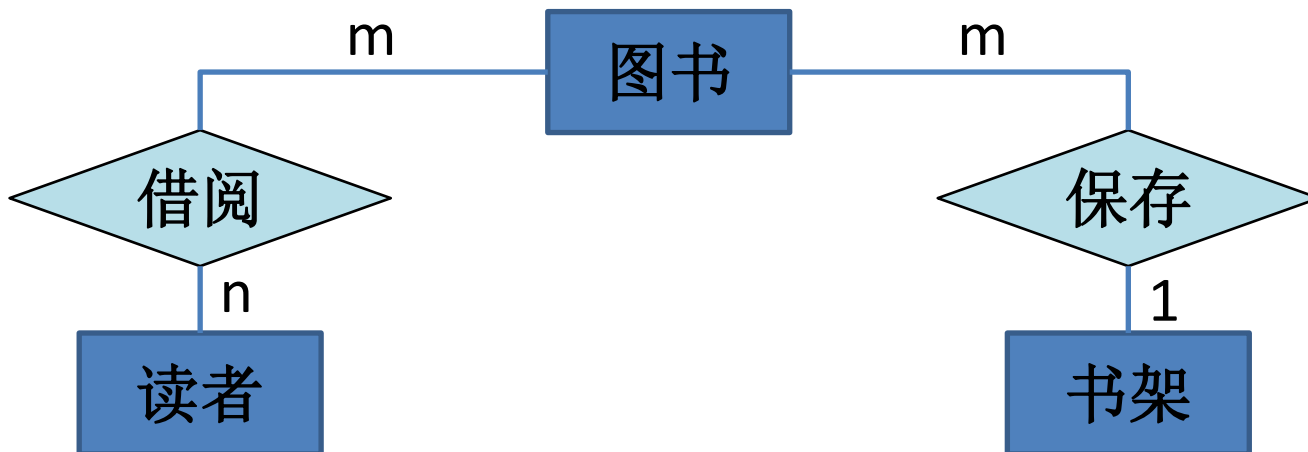
5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：Chen方法示例



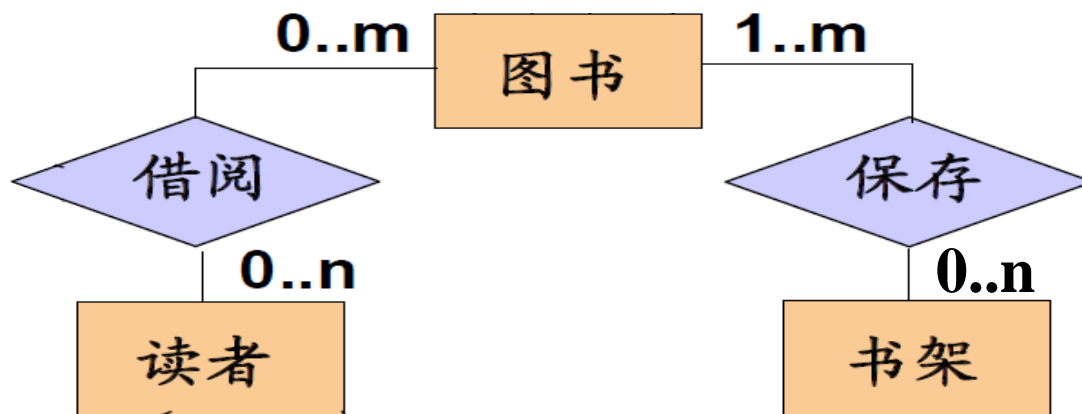
5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法: Chen方法示例



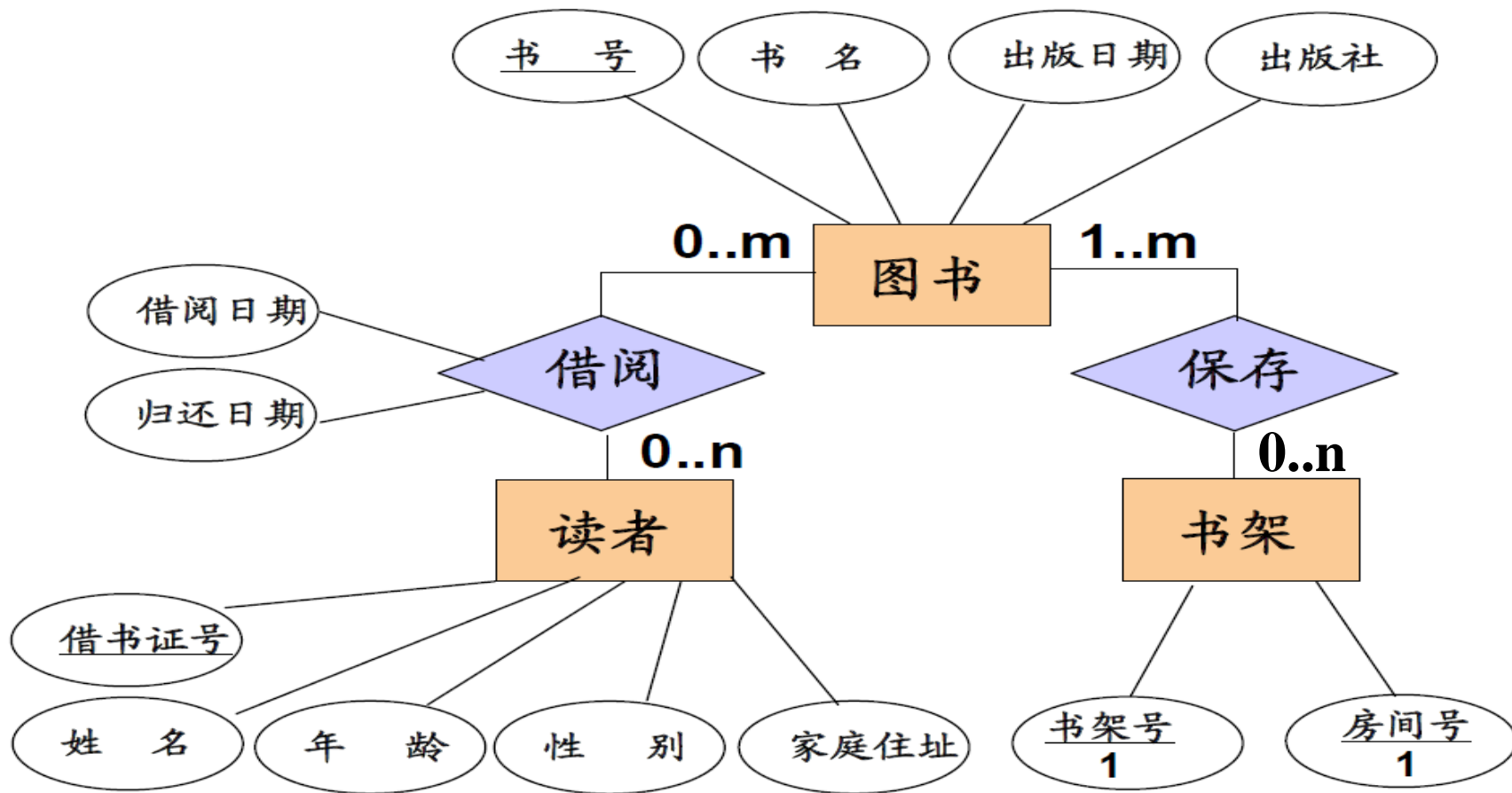
5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：Chen方法示例



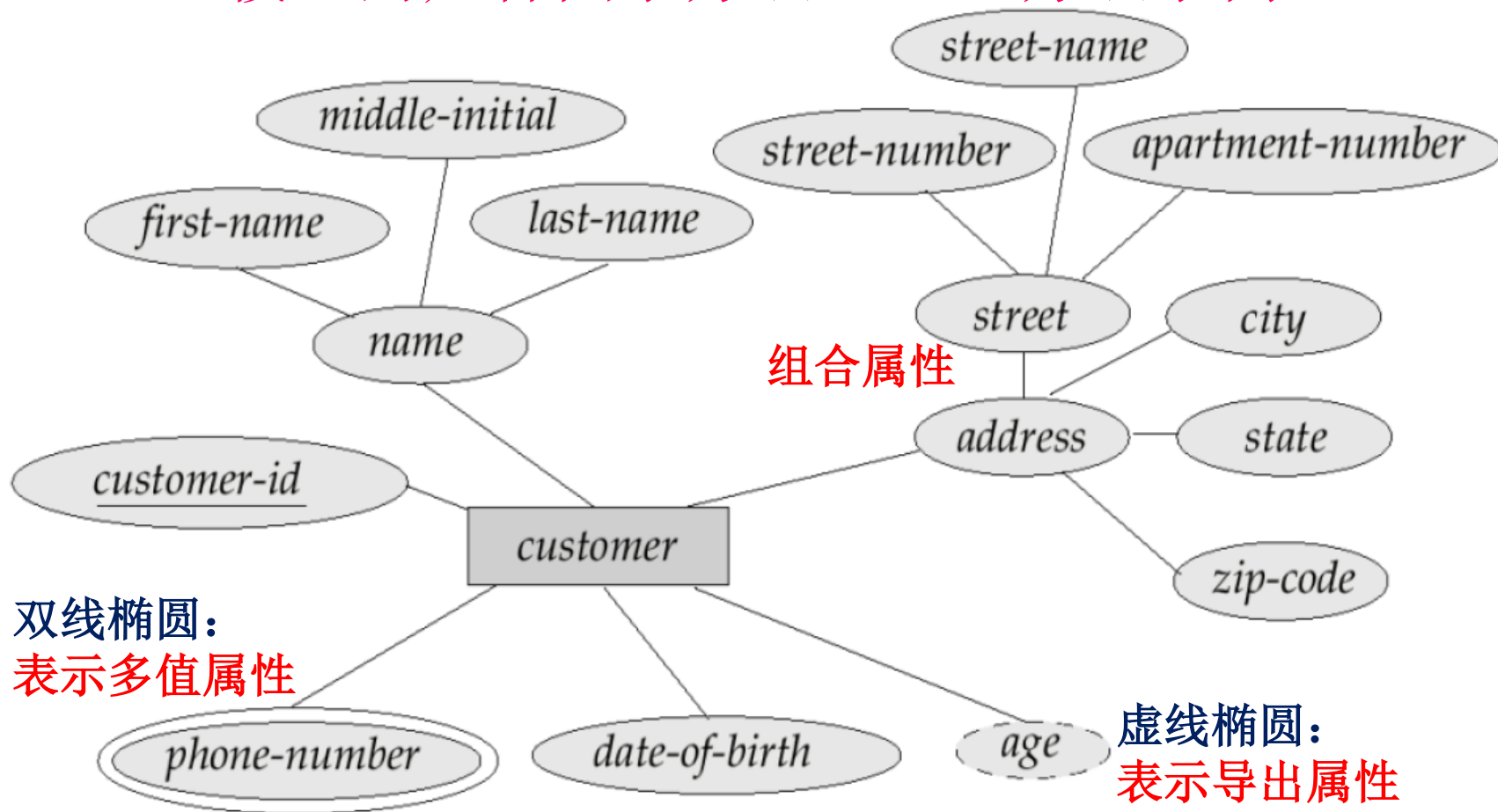
5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：Chen方法示例



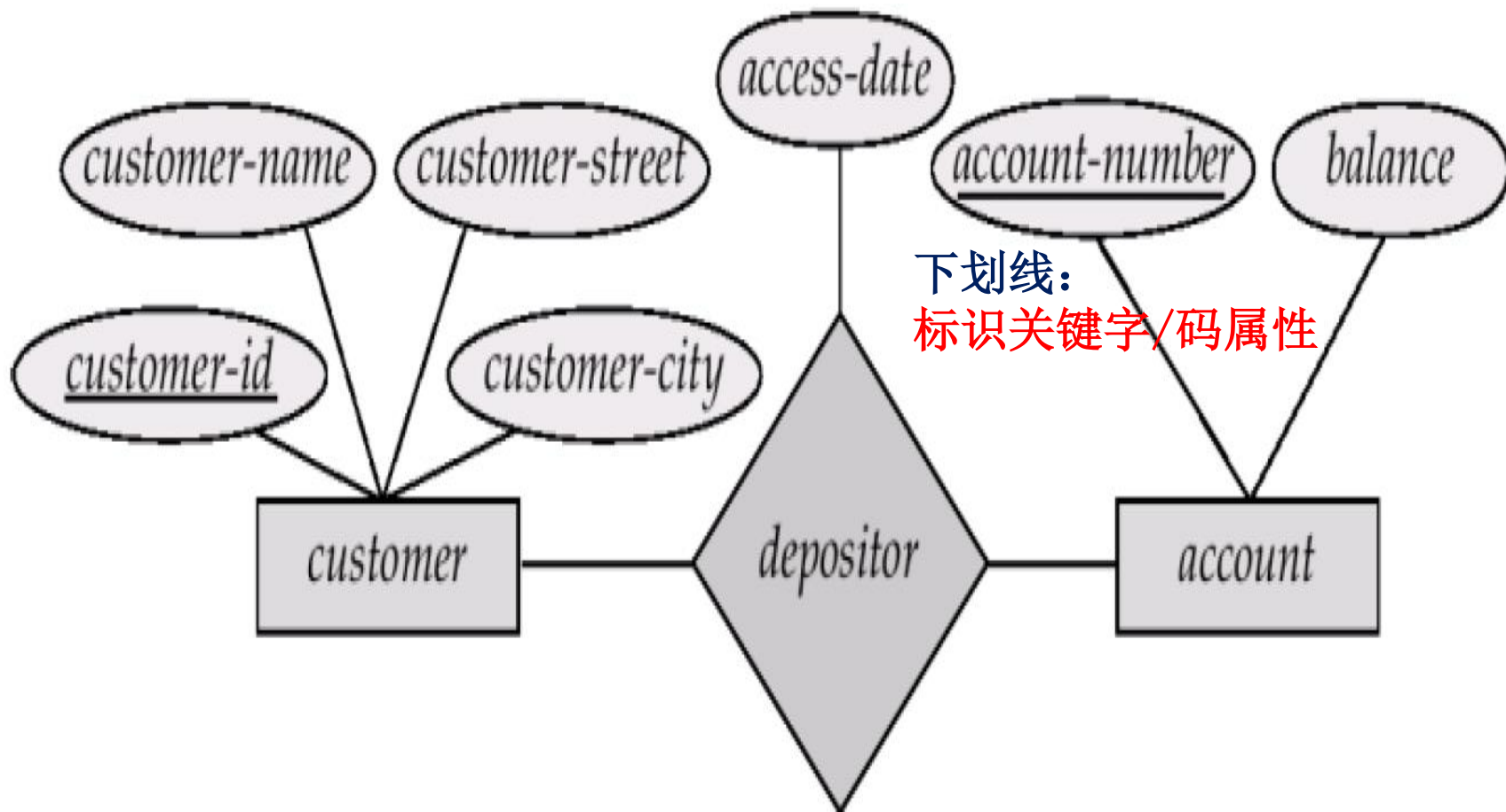
5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：Chen方法示例



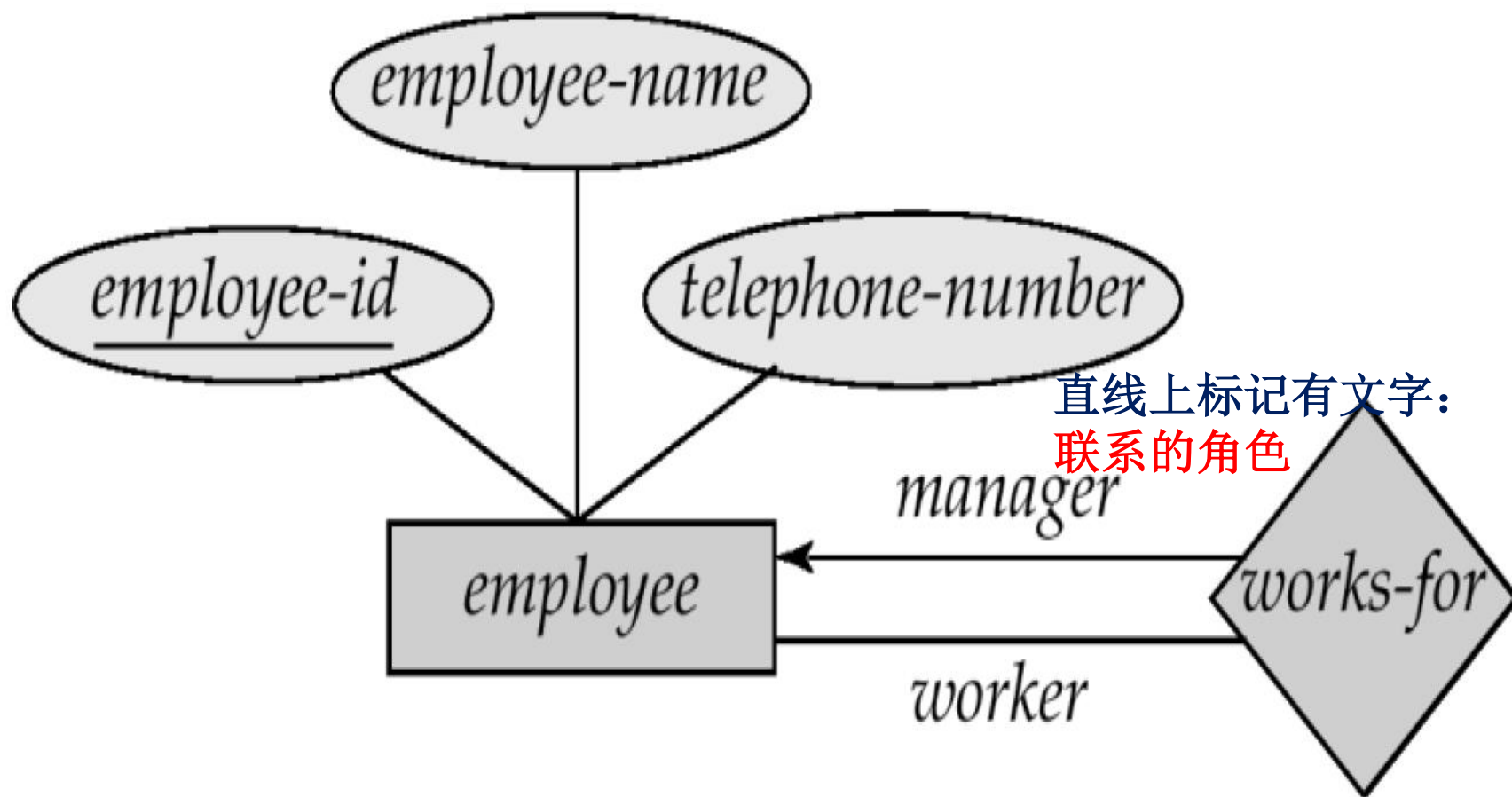
5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：Chen方法示例



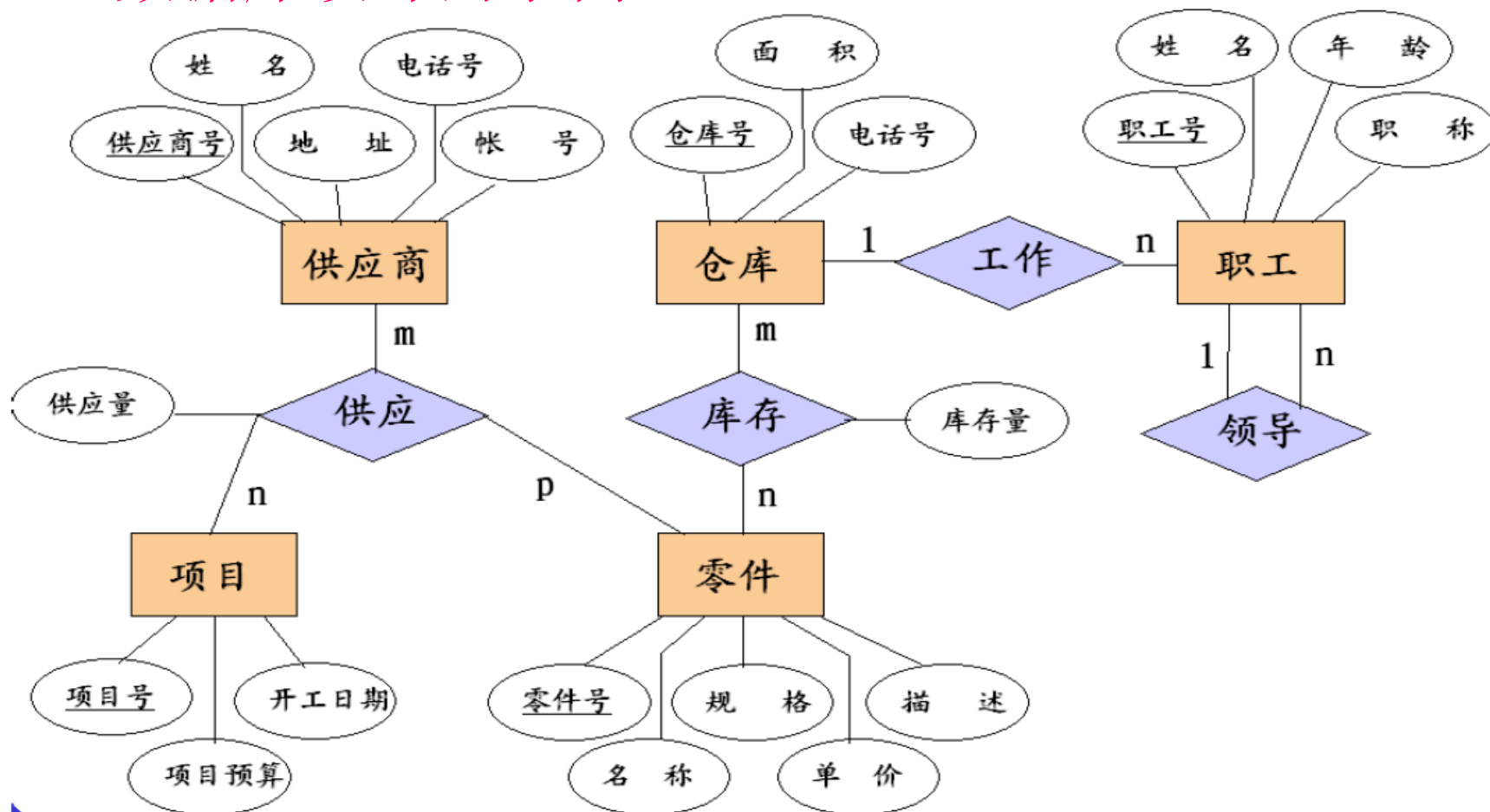
5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：Chen方法示例



5.2 利用E-R模型/IDEF1X模型进行数据库设计

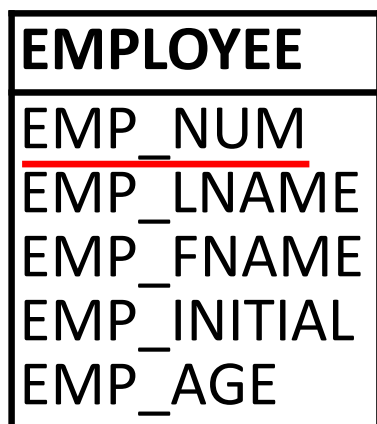
--数据库设计的示例



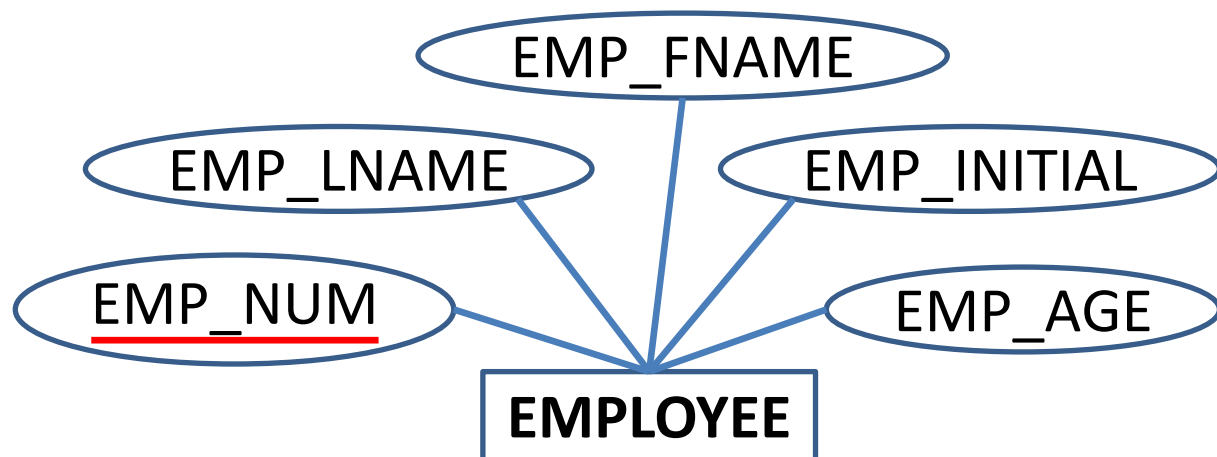
5.2 利用E-R模型/IDEF1X模型进行数据库设计

--简单熟悉Crow's foot图示方法

- 矩形框表示实体。
- 实体名称写在横线上面，实体属性写在横线的下面。
- 实体的关键字属性下加下划线。



crow's foot方法



chen方法

5.2 利用E-R模型/IDEF1X模型进行数据库设计

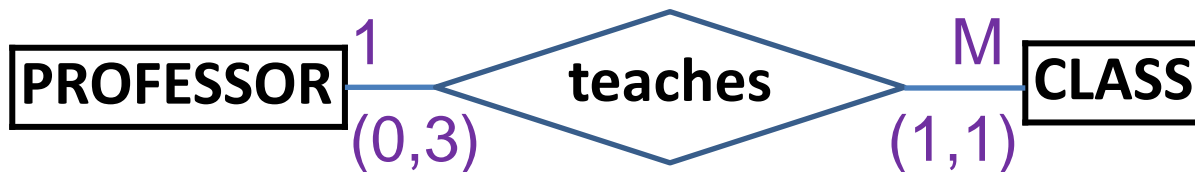
--简单熟悉Crow's foot图示方法

➤ 菱形框表示联系，

也可以将菱形框省略而直接以联系名来替代。



crow's foot
方法



chen方法

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--简单熟悉Crow's foot图示方法

➤ 联系的基数表示方法



1



1 或多(many)



0 或多(many)



0 或 1



多(many)

crow's foot
方法



chen方法

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--简单熟悉Crow's foot图示方法

➤ Crow's foot示例



1:m(one-to-many)情况

业务规则：一个**painter**可以绘制多幅**painting**，而一幅**painting**只能由一个**painter**来绘制



m:n(many-to-many)情况

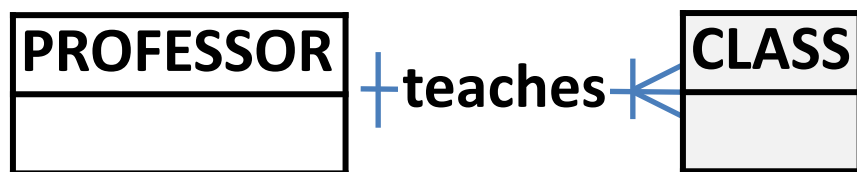
业务规则：一个**student**可以选择多个**class**，而一个**class**又可由多个**student**来组成



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--简单熟悉Crow's foot图示方法

➤ Crow's foot示例



1:m(one-to-many)

完全参与联系

业务规则：一个**professor**至少要教一个**class(>=1)**，而一个**class**只能由一个**professor**来教



1:m(one-to-many)

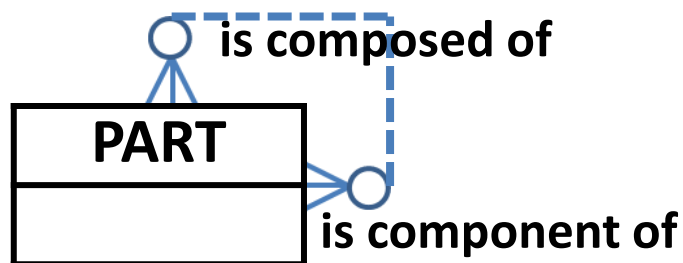
部分参与联系

业务规则：一个**professor**可以教多个**class**，但也可以不教；一个**class**只能由一个**professor**来教

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--简单熟悉Crow's foot图示方法

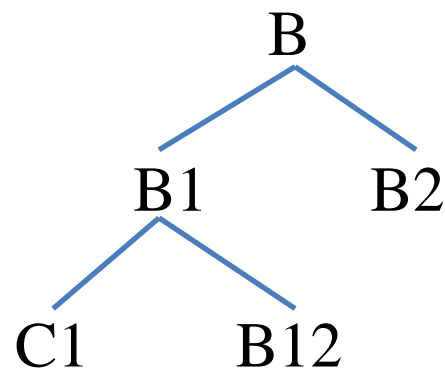
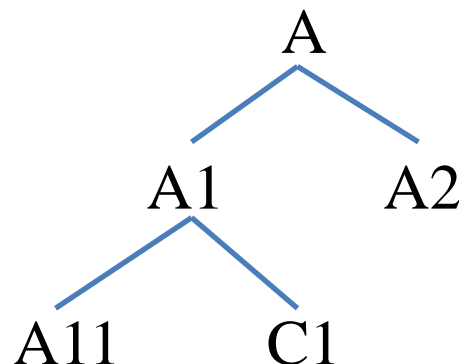
➤ Crow's foot示例



m:n(many-to-many)

部分参与联系

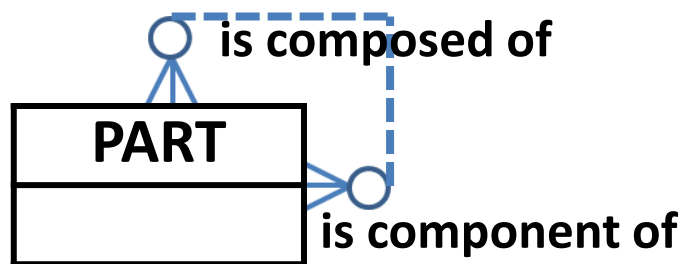
业务规则: 一个 **part** 可以由多个其他的 **part** 构成, 而一个 **part** 也可以构成多个其他的 **part** (但都可以不参与)



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--简单熟悉Crow's foot图示方法

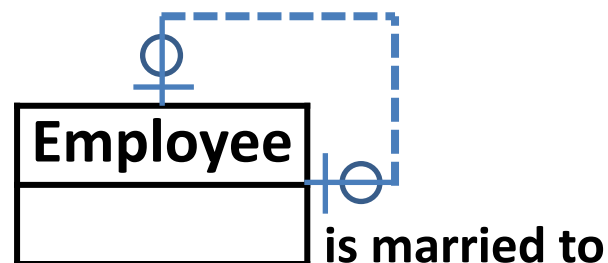
➤ Crow's foot示例



m:n(many-to-many)

部分参与联系

业务规则:一个**part**可以由多个其他的**part**构成, 而一个**part**也可以构成多个其他的**part**(但都可以不参与)



1:1(one-to-one)

部分参与联系

业务规则:一个**employee**可以与另一个且只能一个**employee**结婚, 但其可以不结婚

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ IDEF1X是IDEF标准之一

- ❑ IDEF (ICAM DEFinition method) 是70年代美国空军ICAM项目建立的，是用于描述企业内部运作的一套建模方法。
- ❑ 共有16套方法，每套方法都是通过建模程序来获取某个特定类型的信息。
- ❑ 最常使用的是IDEF0～IDEF4，包括：
 - ✓ IDEF0，功能建模 “function model”
 - ✓ IDEF1，信息建模 “information model”
 - ✓ IDEF2，仿真建模 “dynamics model”
 - ✓ IDEF3：过程描述获取 “Process Description Capture”
 - ✓ IDEF4：面向对象设计 “Object-Oriented Design”
- ❑ 1983年，IDEF1被升级为IDEF1X
IDEF1X：数据建模 (Data Modeling)

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

- IDEF1X将E-R模型扩充语义含义而形成的，或者说，IDEF1X是E-R图的细化…
- IDEF1X是一种进行数据库设计的工程化的方法

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 实体(Entity)

- ❑ 独立标识符实体/独立实体(Identifier-Independent Entity)：强实体
- ❑ 从属标识符实体/从属实体(Identifier-dependent Entity)：弱实体

➤ 联系(Relationship)

- ❑ 可标定连接联系(Identifying Connection Relationship)
- ❑ 非标定连接联系(Non-Identifying Connection Relationship)
- ❑ 分类联系(Categorization Relationship)
- ❑ 非确定联系(Non-Specific Relationship)

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 属性/关键字(Attribute/Key)

- 属性(Attribute)
- 主关键字/主码/主属性(Primary Keys)
- 次关键字/次码/次属性(Alternate Keys)
- 外来关键字/外来码/外来属性(Foreign Keys)

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 实体(Entity):

一个“实体”表示一个现实和抽象事物的集合，这些事物必须具有相同的属性和特征。这个集合的一个元素就是该实体的一个实例。

➤ 实体被区分为独立实体和从属实体。

➤ 独立实体在扩展E-R图中又称强实体，从属实体在扩展E-R图中又称弱实体。

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

- 独立实体：一个实体的实例都被唯一的标识而不决定于它与其他实体的联系

| 合同/E1 |
|-------|
| 合同号 |
| |

| 合同号 | 合同名称 | 签约日期 | 签约人 |
|-------|-------|----------|-----|
| HT001 | A采购合同 | 2012.1.3 | 张三 |
| HT002 | B采购合同 | 2012.2.5 | 李四 |
| HT003 | C采购合同 | 2012.3.3 | 王五 |
| HT004 | D采购合同 | 2012.4.2 | 赵六 |

IDEF1X独立实体描述方法

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

- 从属实体：一个实体的实例的唯一标识需要依赖于该实体与其他实体的联系

一份份合同

| | |
|-------|----------|
| HT001 | A采购合同 |
| HT002 | B采购合同 |
| HT003 | C采购合同 |
| 电脑 | 3000 x 2 |
| 打印机 | 4500 x 1 |

一份合同中的一
项项条款

合
同

合
同
条
目

| 合同号 | 合同名称 | 签约日期 | 签约人 |
|-------|-------|----------|-----|
| HT001 | A采购合同 | 2012.1.3 | 张三 |
| HT002 | B采购合同 | 2012.2.5 | 李四 |
| HT003 | C采购合同 | 2012.3.3 | 王五 |

| | 合同条目号 | 产品 | 价格 | 数量 |
|--|-------|-----|------|----|
| | item1 | 电脑 | 3000 | 2 |
| | item2 | 打印机 | 4500 | 1 |

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

- 从属实体需要从其他实体继承属性作为关键字的一部分。主关键字包含了外来属性的实体为从属实体。

通过继承合同号属性
可将合同实体与合同
条目实体联系起来。

合同条目/E2

合同号(Fk)
合同条目号

IDEF1X从属
实体描述方法

合同

| 合同号 | 合同名称 | 签约日期 | 签约人 |
|-------|-------|----------|-----|
| HT001 | A采购合同 | 2012.1.3 | 张三 |
| HT002 | B采购合同 | 2012.2.5 | 李四 |
| HT003 | C采购合同 | 2012.3.3 | 王五 |

合同
条目

| 合同号 | 合同条目号 | 产品 | 价格 | 数量 |
|-------|-------|-----|------|----|
| HT003 | item1 | 电脑 | 3000 | 2 |
| HT003 | item2 | 打印机 | 4500 | 1 |



5.2 利用E-R模型/IDEF1X模型进行数据库设计

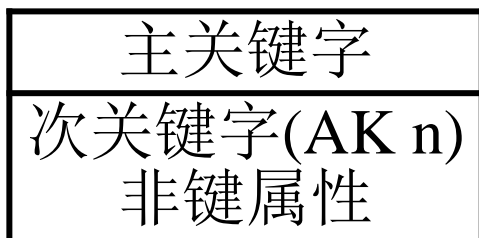
--E-R模型的几种图示方法：IDEF1X方法

➤ 关于实体的规则：工程化的要求

- 独立实体用直角方形框，从属实体用圆角方形框表示；
- 实体用实体名/实体号标识；
- 从属实体的实例依赖于独立实体实例存在而存在；
- 每一个实体必须使用唯一的实体名，相同的含义总是用于同一实体名，相同的含义不能用于不同的实体名；

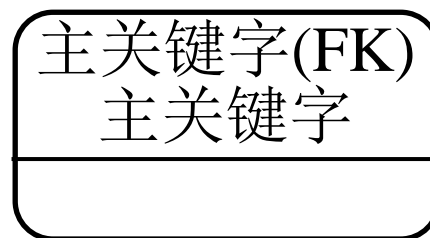
独立实体的图示示例

实体名/实体号



从属实体的图示示例

实体名/实体号



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 关于实体的规则：工程化的要求

- 一个实体可以有一个或多个属性，这些属性可以是其自身所具有的，也可以是通过一个联系而继承得到的；
- 一个实体应有一个或多个能唯一标识实体每一个实例的属性，即应有一个主关键字及若干次关键字(0或多个)；
- 任意实体都可与模型中任意其他的实体有任何联系；
- 如果一个完全外来关键字是一个实体主关键字的全部或部分，那么该实体就是从属实体。相反，如果仅一部分或根本没有外来关键字属性用作一个实体的主关键字，那么，这个实体就是独立实体。

5.2 利用E-R模型/IDEF1X模型进行数据库设计

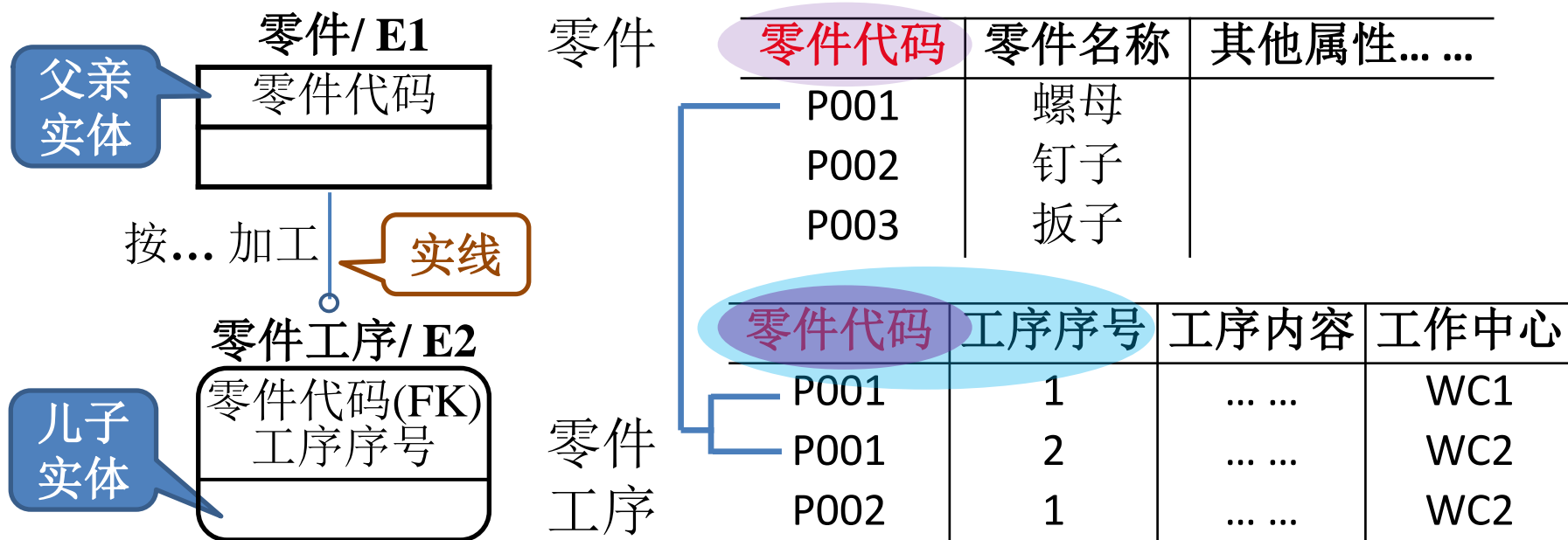
--E-R模型的几种图示方法：IDEF1X方法

- 联系(Relationship):
是实体之间的一种连接关系
- 联系有连接联系、分类联系和不确定性联系
- 连接联系，又称父子联系或依存联系。可进一步区分为标定联系、非标定联系。
 - ◆ 标定联系
 - ◆ 非标定联系
 - ◆ 分类联系
 - ◆ 不确定联系

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

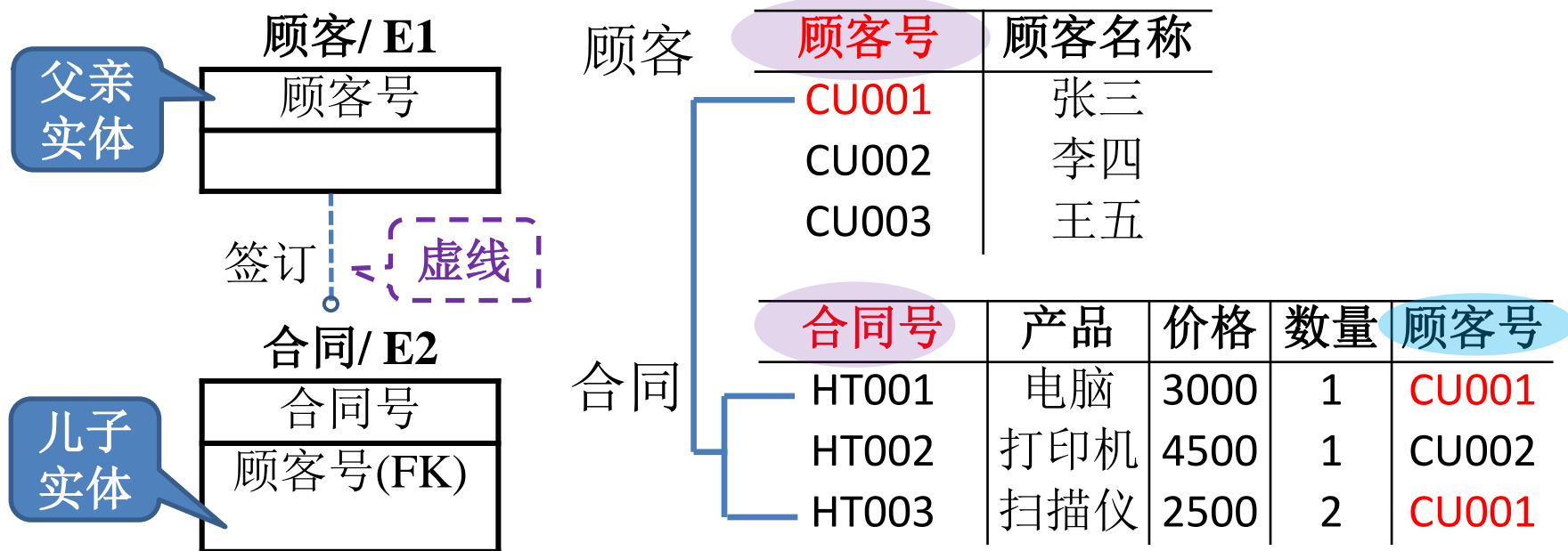
- 标定联系：儿子实体的实例都是由它与父亲实体的联系而确定。父亲实体的主关键字是儿子实体主关键字的一部分



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

- 非标定联系：儿子实体的实例能够被唯一标识而无需依赖与其他实体的联系。父亲实体的主关键字不是儿子实体的主关键字。

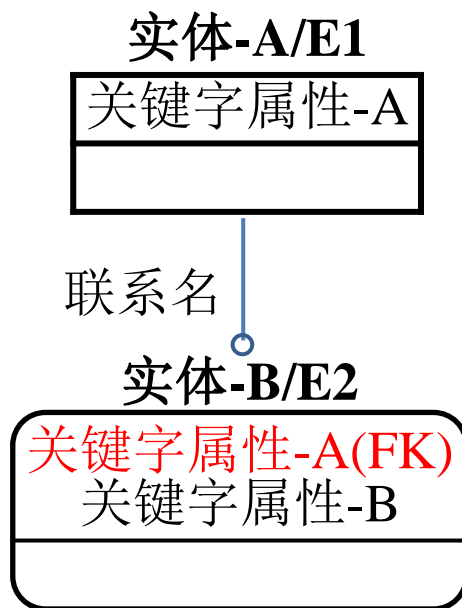
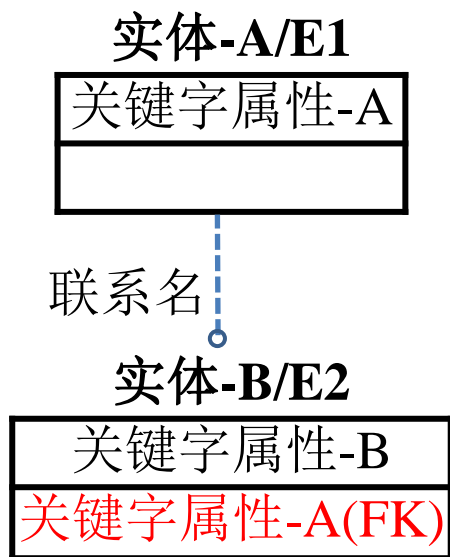


5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 关于连接联系的规则：工程化的要求

- 标定联系用实直线表示，非标定联系用虚直线表示
- 在子实体一侧有圆圈，联系名标注在直线旁



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 关于连接联系的规则：工程化的要求

- 一个确定性连接联系总是存在于两个实体之间，一个作为父亲实体，另一个作为儿子实体；
- 儿子实体的一个实例必须且总是恰好地与父亲实体的一个实例相联系；
- 父亲实体一个实例可与儿子实体的0个、1个或多个实例相联系，具体情况由特定基数而定。在儿子端标注 $P(1 \text{或大于} 1) / Z(0 \text{或} 1) / n(\text{确定数目}) / \langle \text{省略} \rangle (0, 1 \text{或大于} 1)$ ；
- 在标定联系中的儿子实体总是一个从属标识符实体；
- 一个实体可以与任意多个其他实体相联系，可以在不同的联系中充当不同的角色，如在一些联系中充当父亲实体，而在另外一些联系中充当儿子实体。

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 分类联系：

一个实体实例是由一个一般实体实例及一个分类实体实例构成的

➤ 一个一般实体是若干具体实体(分类实体)的类

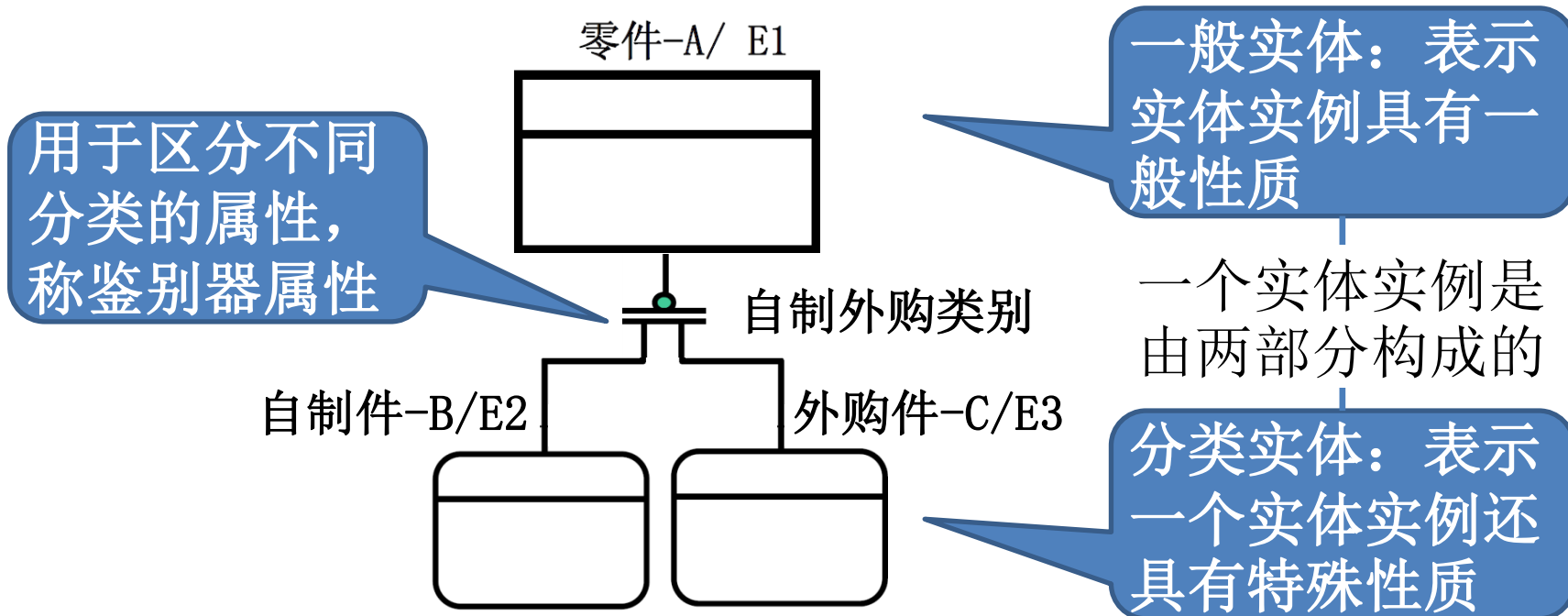
➤ 分类实体与一般实体具有相同的标识符

➤ 不同分类实体除具有一般实体特征外，各自还可能具有不同的属性特征

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

- 例如：零件 { 自制件 外购件 } 自制件和外购件不仅具有相同的属性，而且还各自具有不同的属性，如自制件需有工艺信息等。



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--分类联系的进一步理解

➤ 特殊化(Specialization)

- 实体的实例集中，某些实例子集具有区别于该实例集内其它实例的特性，可以根据这些差异特性对该实例集进行分组/分类，这一分组/分类的过程称作特殊化
- 若两个子集 S_i 与 S_j 没有交集，则S称为E的不相交特殊化，否则称为重叠特殊化。
 - ✓ 自顶向下、逐步求精
 - ✓ 父类--子类
 - ✓ 子类 == 特例 == 更小的实例集合 == 更多的属性

例如：一个银行帐号可以有存款帐号、贷款帐号

学生可以有研究生、本科生

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--分类联系的进一步理解

➤ 概括 (Generalization)

- 若干个实体根据共有的性质，可以合成一个较高层的实体。
概括是一个高层实体与若干个低层实体之间的包含关系
- 自底向上、逐步合成

➤ 概括 Vs 特殊化

- 概括与特殊化是个互逆的过程
- 特殊化强调同一实体不同实例之间的差异，概括强调不同实体之间的相似性
- 反映了数据库设计或数据库抽象的不同思路或方法

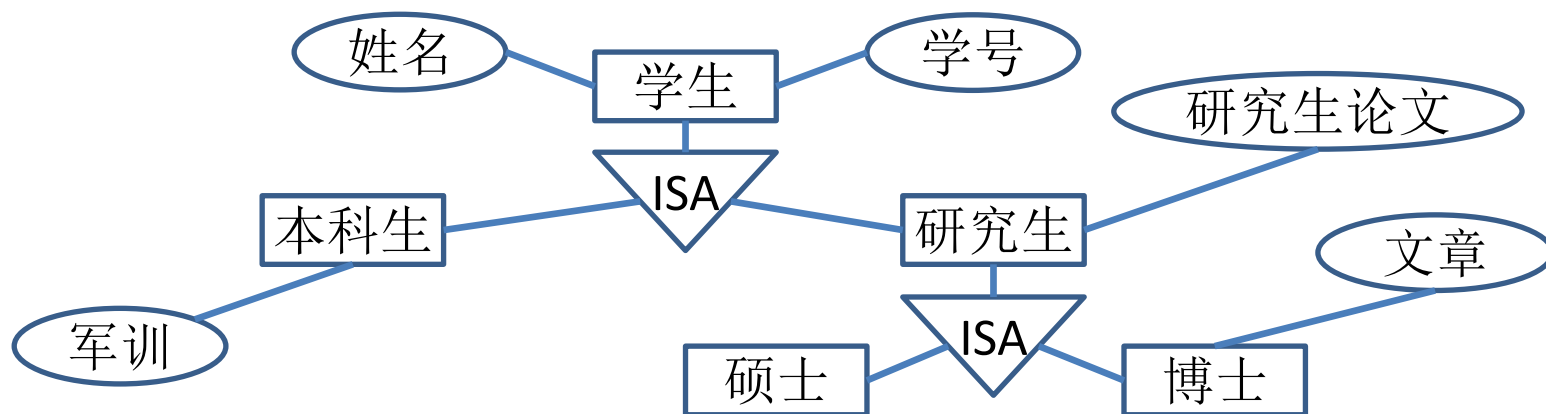
5.2 利用E-R模型/IDEF1X模型进行数据库设计

--分类联系的进一步理解

➤ 特殊化和概括在E-R图中用标记为ISA的三角形表示

▣ ISA = “is a”, 表示高层实体和低层实体之间的“父类—子类”联系

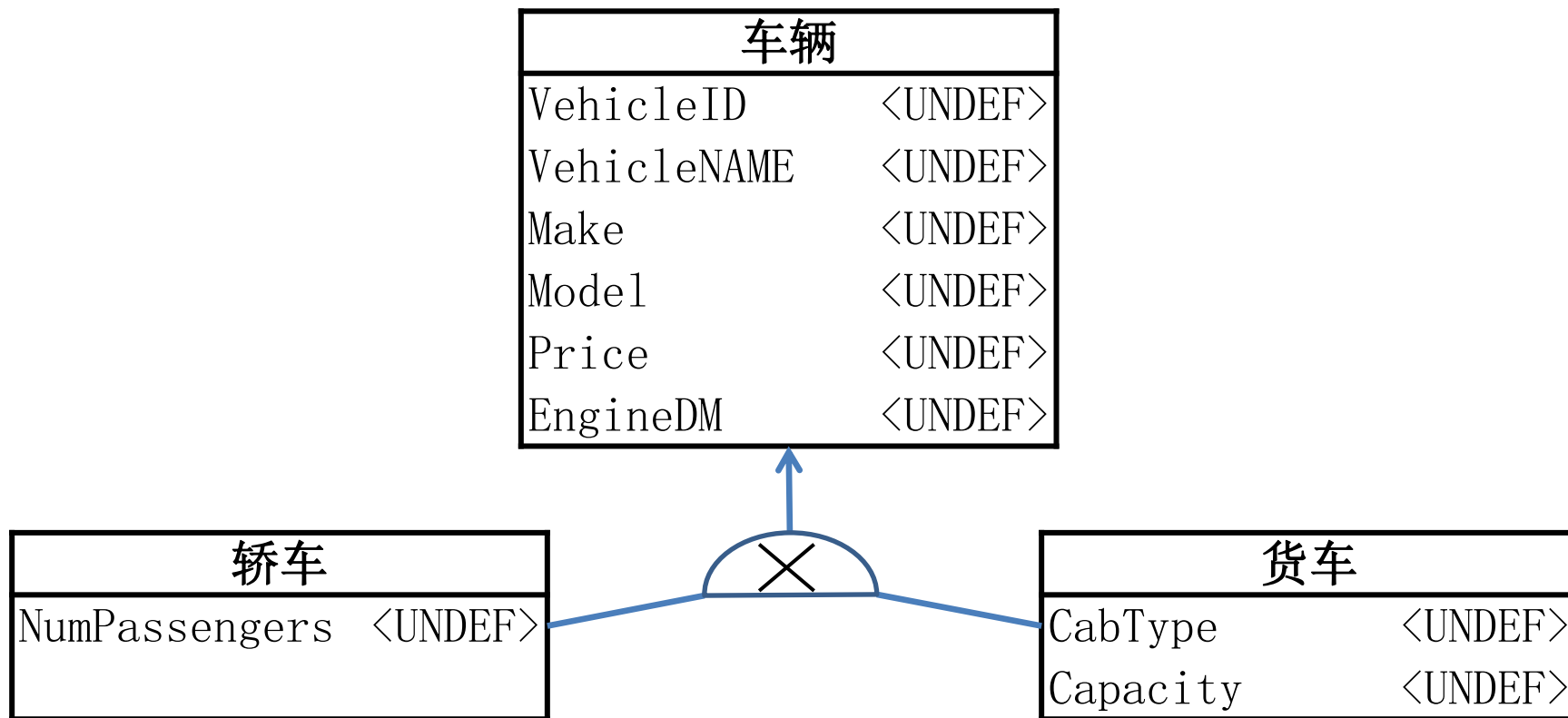
➤ 在IDEF1X中特殊化和概括表征的就是一种分类联系，体现了不同的思路



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--分类联系的进一步理解

➤ Crow's foot表达分类联系的符号



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--分类联系的进一步理解

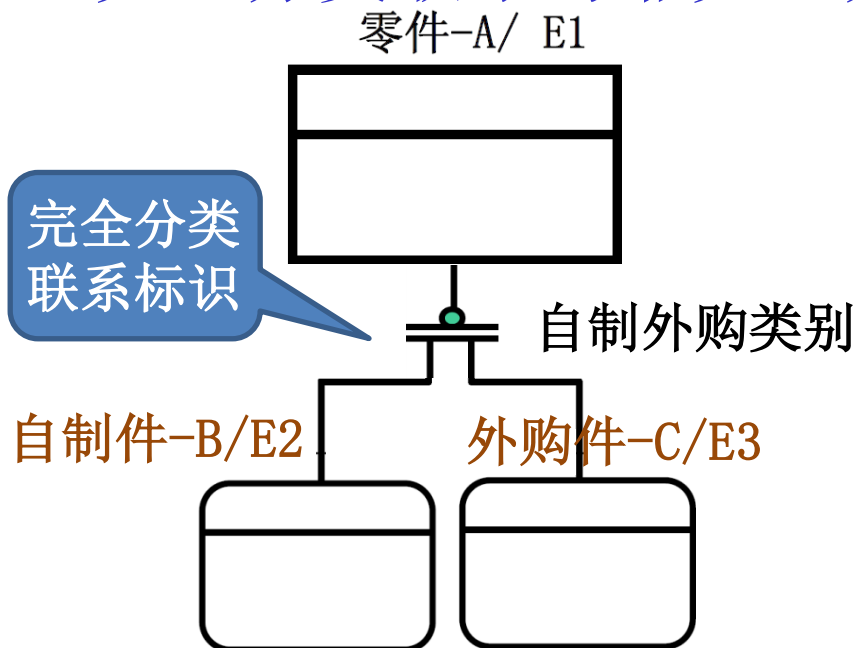
➤ 属性继承

- ▣ 高层实体的属性被低层实体自动继承
 - ▣ 低层实体特有的性质仅适用于某个特定的低层实例
- 如“研究生论文”属性只适用于“研究生”实例

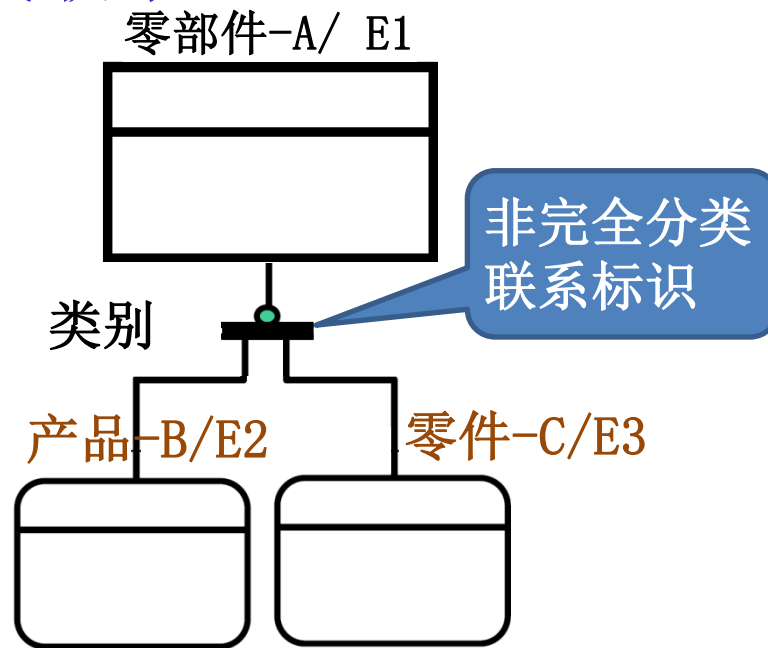
5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 完全分类联系与非完全分类联系



业务规则：一个零件或者是自制件或者是外购件，只能是二者之一。
此分类是分类完全集



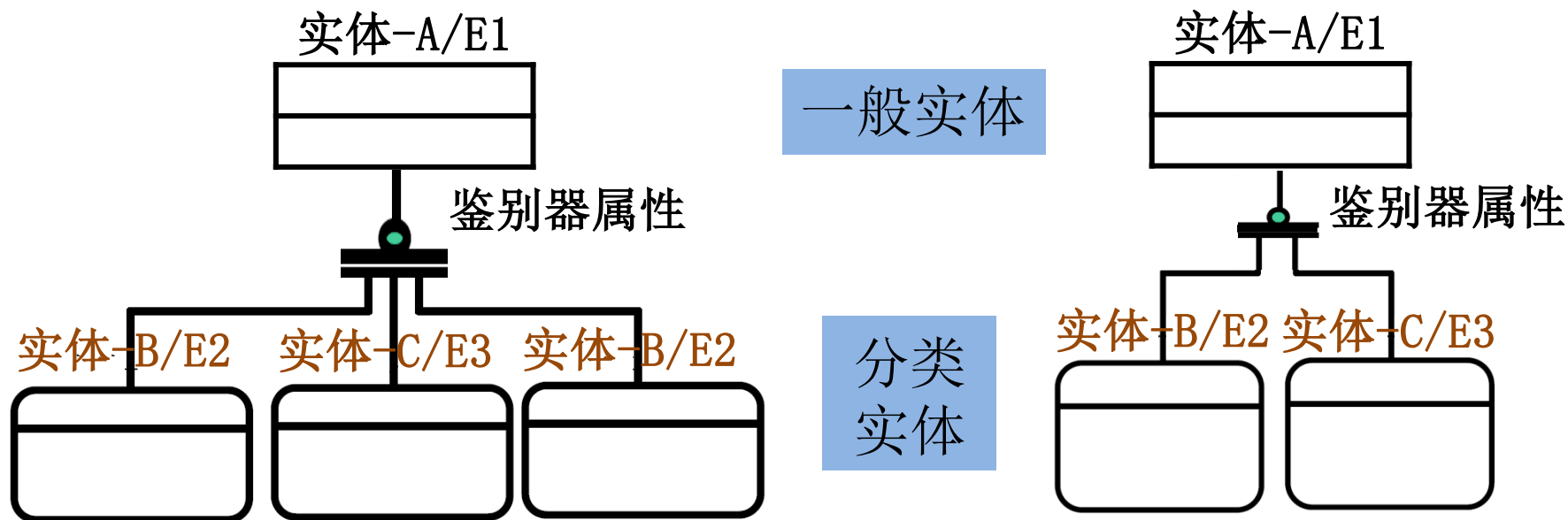
业务规则：零部件除产品、零件外，还可能有部件/部套还有组合件等
此分类是分类非完全集

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 关于分类联系的规则：工程化的要求

- 一圆圈带两横线：完全分类联系；
- 一圆圈带一横线：非完全分类联系；



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 关于分类联系的规则：工程化的要求

- 一个分类实体只能有一个对应的一般实体，即对一分类联系而言，它只能是一个分类集的成员；
- 一个分类联系中的一个分类实体可以是一个其他分类联系中的一般实体；
- 一个实体可以具有任意个分类联系，在这些分类联系中，这个实体作为一般实体。

例如“雇员”实体可分类为“计时雇员”和“月薪雇员”，也可分类为“男雇员”和“女雇员”

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

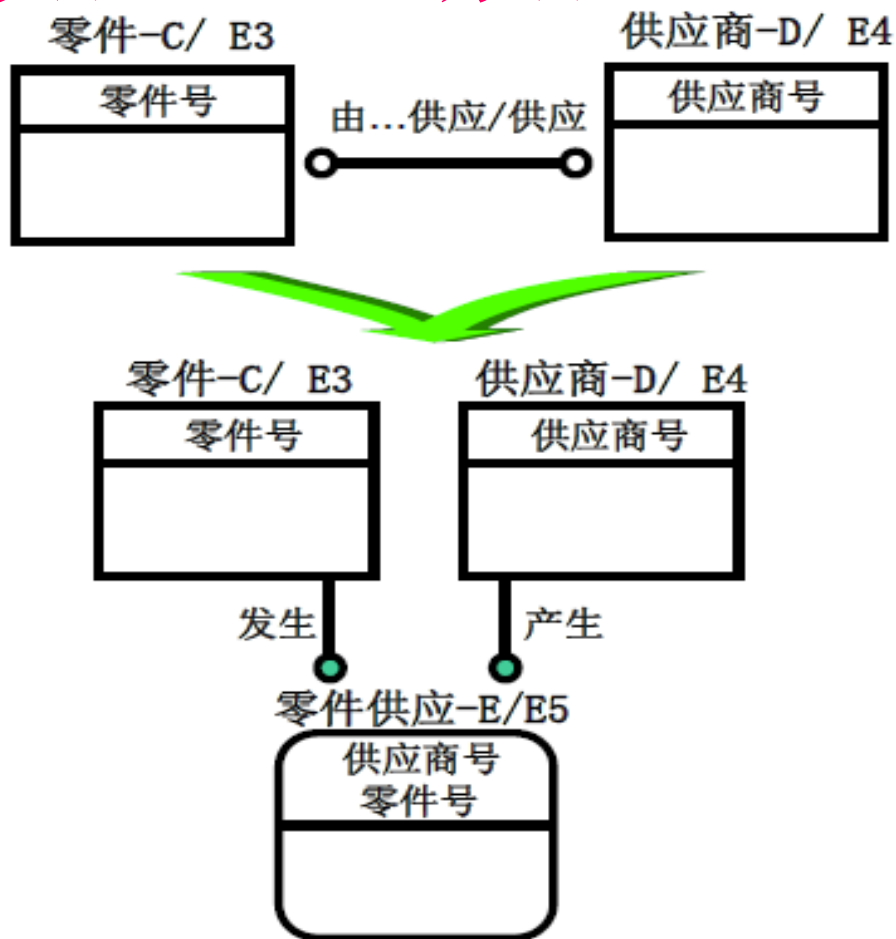
➤ 关于分类联系的规则：工程化的要求

- 一个分类实体不能是可标定联系中的儿子实体；
- 分类实体的主关键字属性必须和一般实体主关键字属性相同；
- 一个分类实体的全部实例都具有相同的“鉴别器值”，并且不同分类实体的实例都具有不同的鉴别器值。

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

- 非确定联系：
为多对多的联系
- 非确定联系需分解为
若干个一对多的联系
- 非确定联系示例

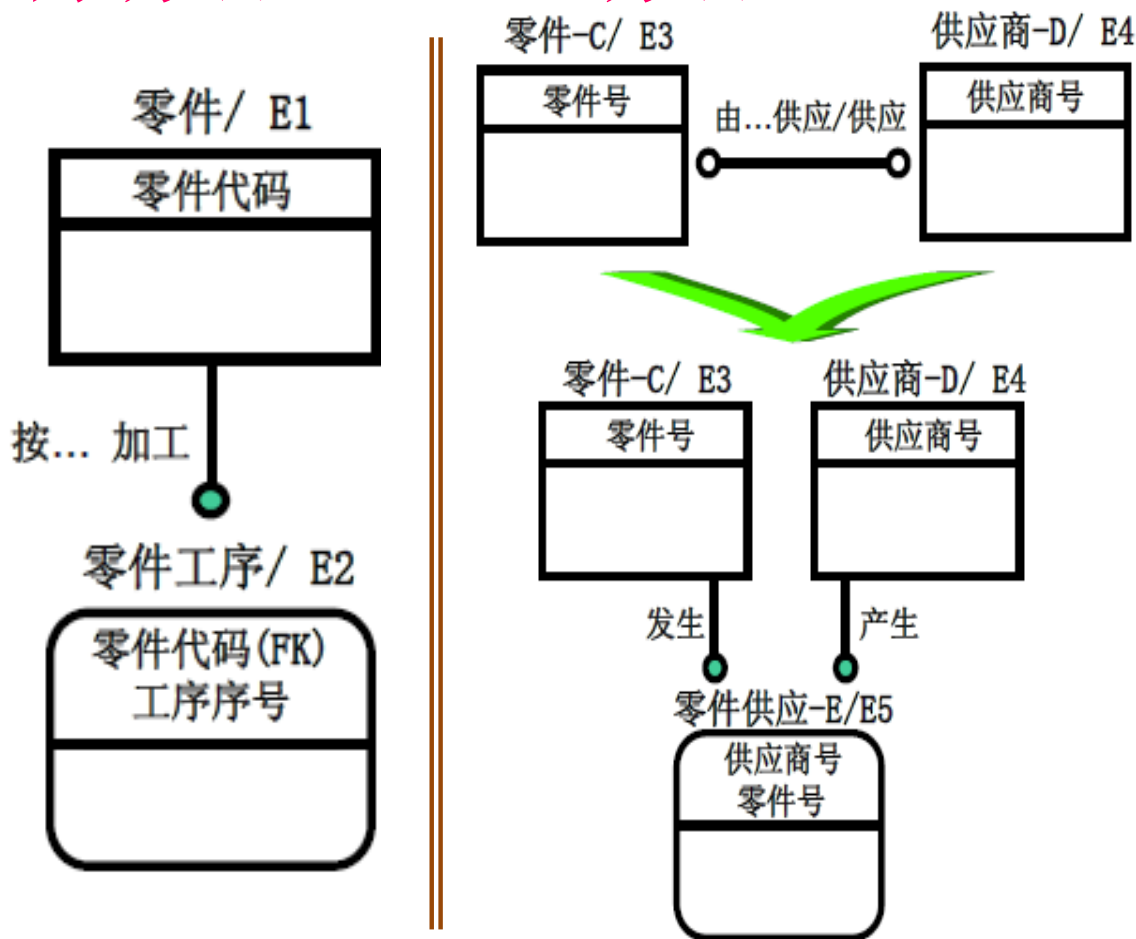


5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 确定性联系通过属性继承实现两实体之间的连接

➤ 非确定性联系通过引入相交实体或相关实体实现两实体的连接



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 关于非确定联系的规则：工程化的要求

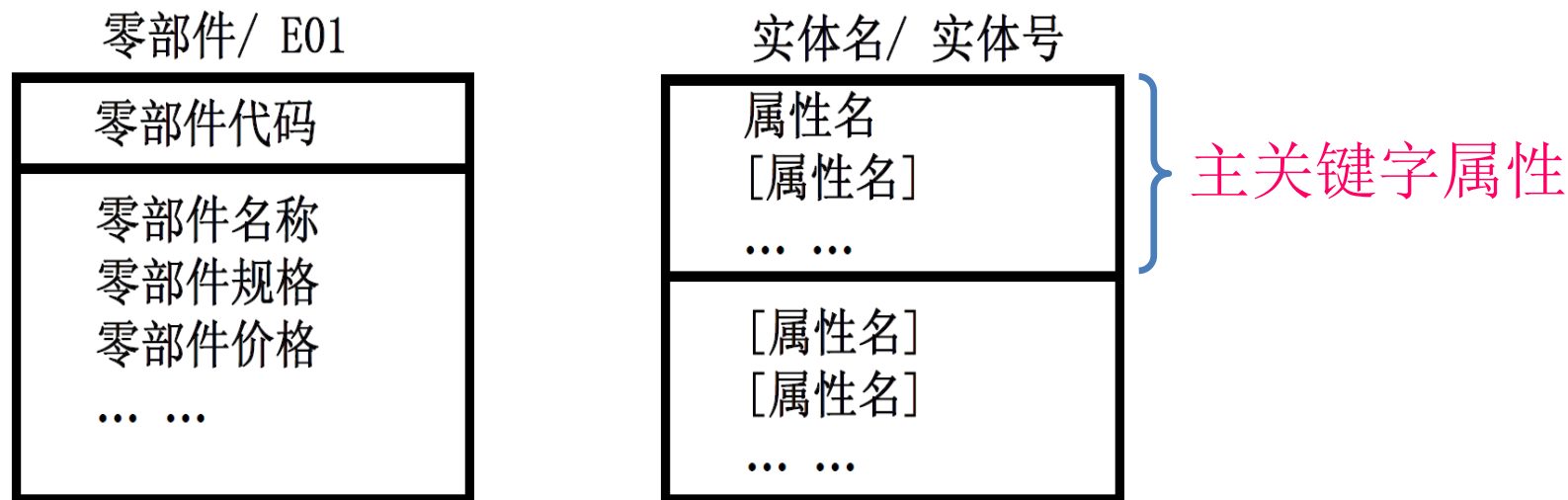
- ❑ 一个非确定联系总是存在于两个实体之间，而不是三个或更多个实体之间
- ❑ 两个实体中，任意一个实体的实例可以与另一实体的0, 1或多个实例相关联，具体情况要视情况而定，在图中标出其基数
- ❑ 为了完全地设计出一个模型，非确定联系必须由确定联系来替代

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 属性：

表示一类现实或抽象事物的一种特征或性质。



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 关于属性的规则：工程化的要求

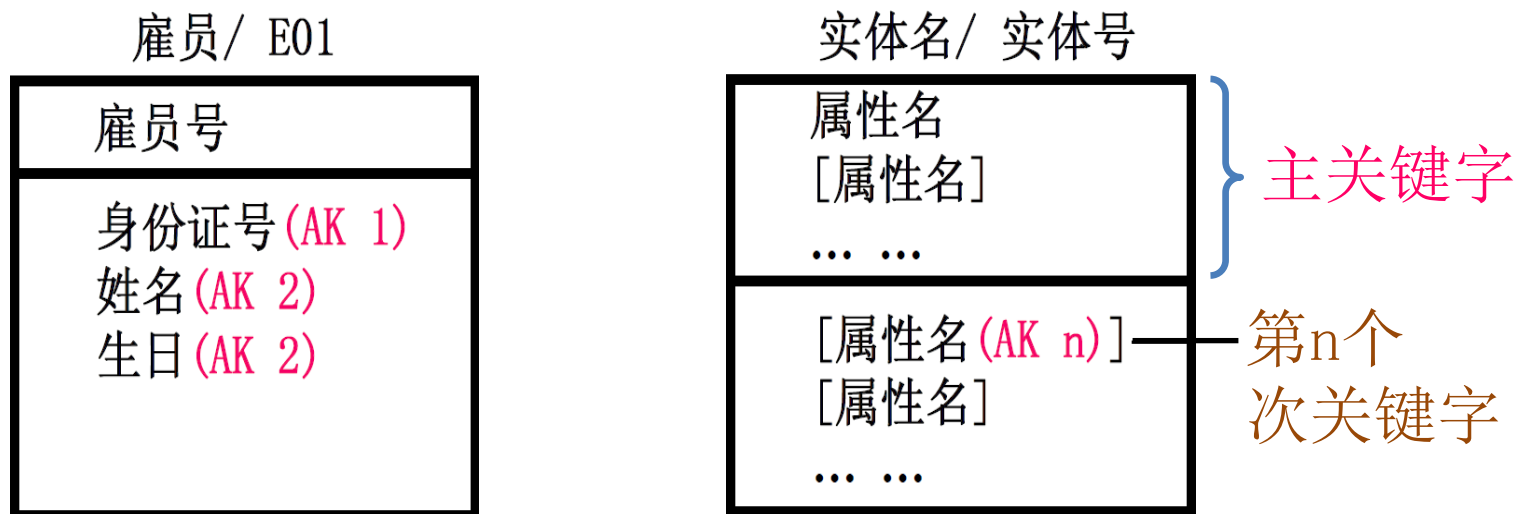
- ❑ 每个属性都必须有一个唯一的名称，且相同的名字必须总是描述相同的含义。因此相同的含义不可能对应于不同的名字(别名除外)
- ❑ 每个实体可以具有任意个属性，一个属性只能归属于一个实体，这一规则称“单主规则”
- ❑ 一个实体可有任意个继承属性，而每个继承属性都必须是某个相关的父亲实体或一般实体主关键字的一部分。
- ❑ 实体的每一个实例，对每一个属性都必须具有一个值。这一规则称为“非空规则”
- ❑ 对于同某实体相关的属性而言，该实体没有一个实例可能具有一个以上的值。这一规则称为“非重复规则”

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 关键字：

能唯一确定实体每一个实例的属性或属性组。



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 关于主关键字和次关键字的规则：工程化的要求

- 每个实体必须有一个主关键字
- 每个实体可有任意个次关键字
- 主关键字和次关键字可由单个或多个属性组成
- 个别属性可以是多个关键字的一部分
- 构成主关键字或次关键字的属性可以是实体自身所具有的或由某些联系继承得到的属性
- 主关键字和次关键字必须仅包含有助于唯一标识实体的那些属性。

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

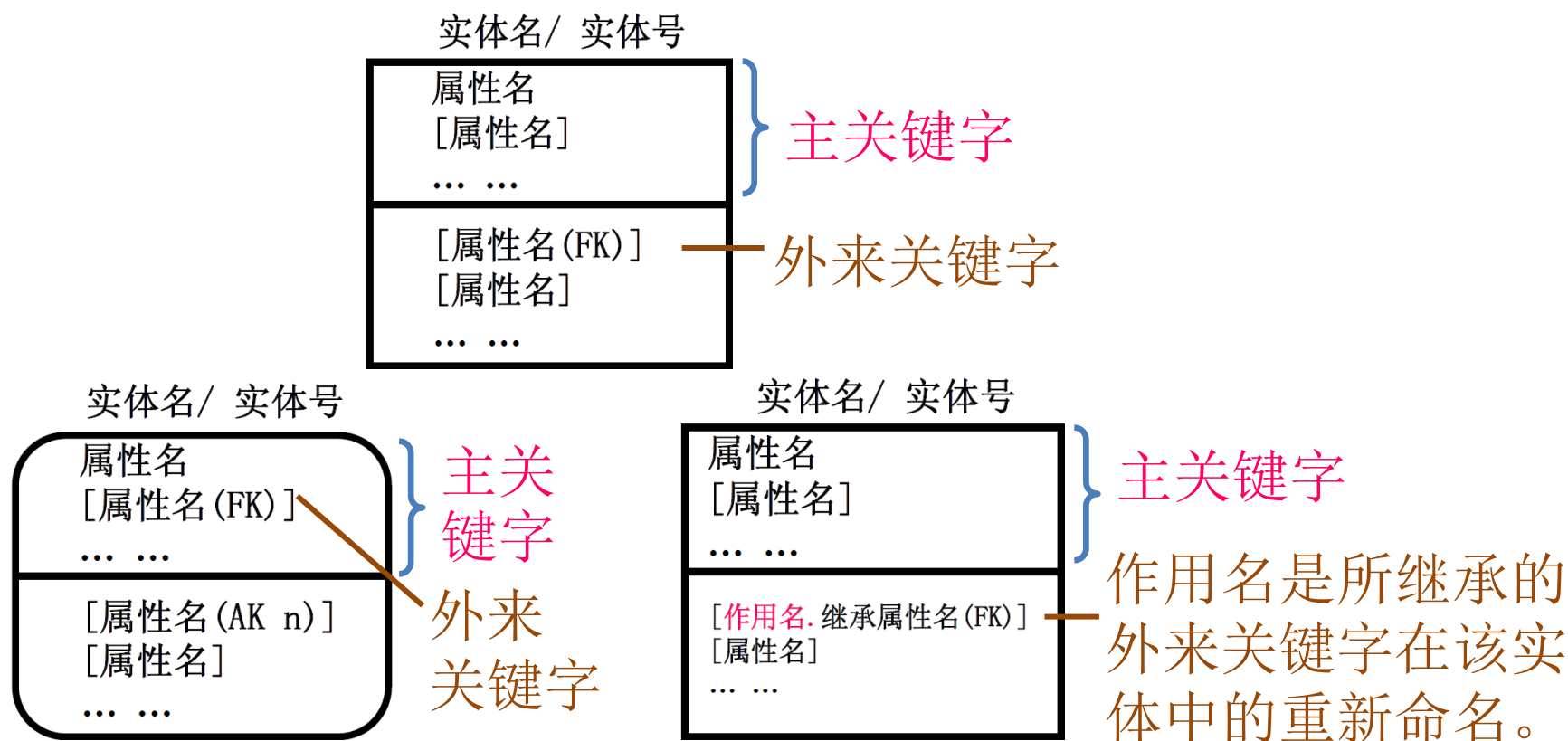
➤ 关于主关键字和次关键字的规则：工程化的要求

- 也就是说，如果主关键字或次关键字中去掉任一部分属性，那么都无法唯一确定实体的实例。此规则称“最小关键字规则”。
- 如主关键字是由多个属性组成，那么每个非键属性的值必须完全函数依赖于主关键字，也就是，如果主关键字的一部分属性被确定了，那么非键属性的值无法唯一确定。此规则称“完全函数依赖规则”。
- 每个非键属性必须是仅仅函数依赖于主关键字和次关键字，也就是，没有一个非键属性的值能够由其他非键属性的值所确定。此规则称“非传递依赖规则”

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

➤ 外来关键字：是其他实体的关键字



数据库系统

外来关键字的例子

零件/ E1

| 零件代码 |
|------|
| |

按...加工

零件工艺路线/ E2

| 零件代码(FK) 工艺路线序号 |
|--------------------|
| |

按...加工

零件工序/ E2

| 零件代码(FK) 工艺路线号(FK) 工序序号 |
|-------------------------------|
| |

| 零件代码 | 零件名称 | 其他属性 |
|------|------|------|
| P001 | 螺栓 | |
| P002 | 螺母 | |
| P003 | 钉子 | |

零件

| 零件代码 | 工艺路线号 | 部门 | 工艺内容 |
|------|-------|------|------|
| P001 | 1 | 组装车间 | ... |
| P001 | 2 | 组装车间 | ... |
| P002 | 1 | 焊接车间 | |

零件
工艺
路线

| 零件代码 | 工艺路线号 | 工序序号 | 工序内容 | 工作中心 |
|------|-------|------|------|------|
| P001 | 1 | 1 | ... | WH01 |
| P001 | 1 | 2 | ... | WH02 |
| P001 | 2 | 1 | ... | WH01 |
| P002 | 1 | 1 | ... | WH01 |

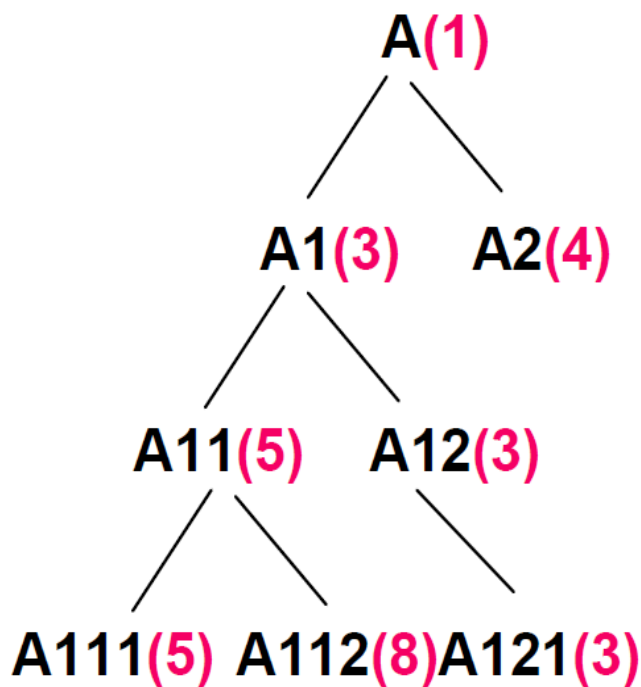
零件
工序



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

- 作用/角色：当一个实体与其父亲实体有多种联系时，此时需使用“作用/角色”来区分每一种联系。



| 零件代码 | 零件名称 | 其他属性 |
|------|--------|--------|
| P001 | 产品A | |
| P002 | 部套A1 | |
| P003 | 部套A2 | |
| P004 | 部件A11 | |
| P005 | 部件A12 | |
| P006 | 部件A111 | |
| P007 | 部件A112 | |
| P008 | 部件A121 | |

零部件清单

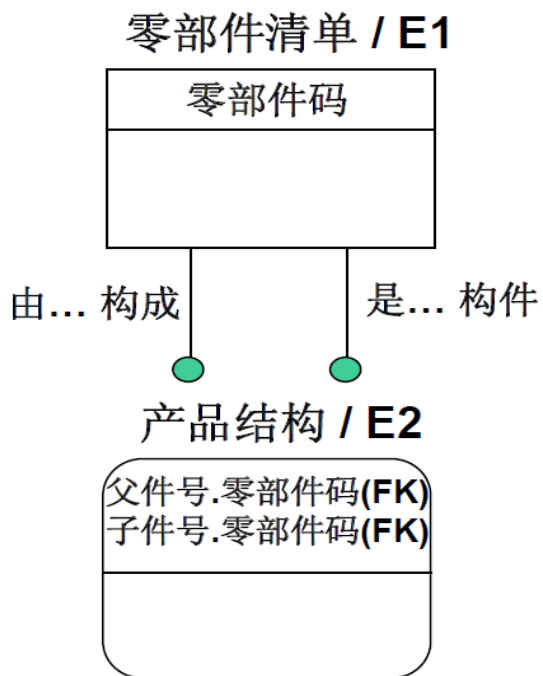
| 父件号 | 子件号 | 数量 |
|------|------|----|
| P001 | P002 | 3 |
| P001 | P003 | 4 |
| P002 | P004 | 5 |
| P002 | P005 | 3 |
| P004 | P006 | 5 |
| P004 | P007 | 8 |
| P005 | P008 | 3 |

产品结构

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

- 作用/角色：当一个实体与其父亲实体有多种联系时，此时需使用“作用/角色”来区分每一种联系。



作用名. 继承属性名(FK)

| 零件代码 | 零件名称 | 其他属性 |
|------|--------|--------|
| P001 | 产品A | |
| P002 | 部套A1 | |
| P003 | 部套A2 | |
| P004 | 部件A11 | |
| P005 | 部件A12 | |
| P006 | 部件A111 | |
| P007 | 部件A112 | |
| P008 | 部件A121 | |

零部件清单

| 父件号 | 子件号 | 数量 |
|------|------|----|
| P001 | P002 | 3 |
| P001 | P003 | 4 |
| P002 | P004 | 5 |
| P002 | P005 | 3 |
| P004 | P006 | 5 |
| P004 | P007 | 8 |
| P005 | P008 | 3 |

产品结构

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R模型的几种图示方法：IDEF1X方法

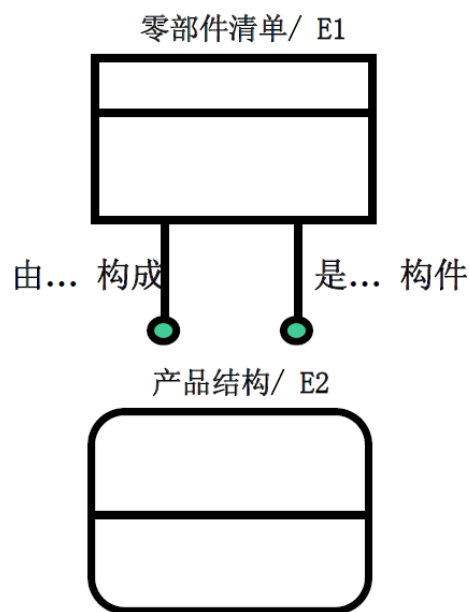
➤ 关于外来关键字的规则：工程化的要求

- ❑ 在确定连接联系或分类联系中的儿子实体或分类实体时必须包含一个外来关键字
- ❑ 一般实体的主关键字必须遗传为每一个分类实体的主关键字
- ❑ 存在一个联系，只能有一个外来关键字
- ❑ 被继承属性只能是主关键字所包含的属性
- ❑ 分配给继承属性的每一个作用名(Role Name)都必须是唯一的，同时同一含义必须应用于同一作用名
- ❑ 如果在某实体的任一给定实例中，对于两个外来关键字而言，单一遗传属性总是具有相同值，那么，该属性可以是多个外来关键字的部分

5.2 利用E-R模型/IDEF1X模型进行数据库设计

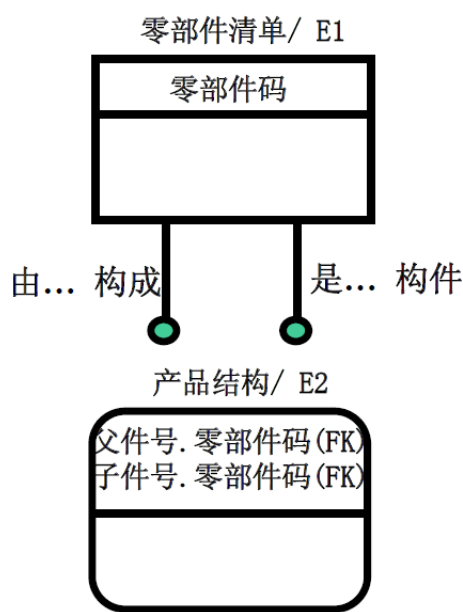
--E-R模型的几种图示方法：IDEF1X方法

➤ IDEF1X描述的层次：实体级图、键级图及完整图



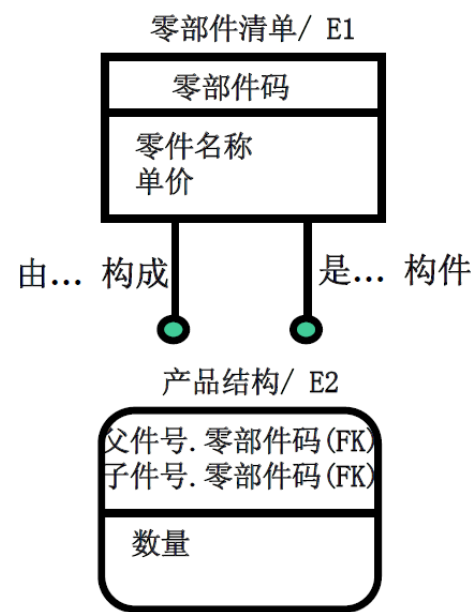
实体级图

以实体为建模单位的IDEF1x图



键级图

以实体为建模单位并且标注键属性的IDEF1x图



完整图

以实体为建模单位并且标注所有属性的IDEF1x图

5.2 利用E-R模型/IDEF1X模型进行数据库设计

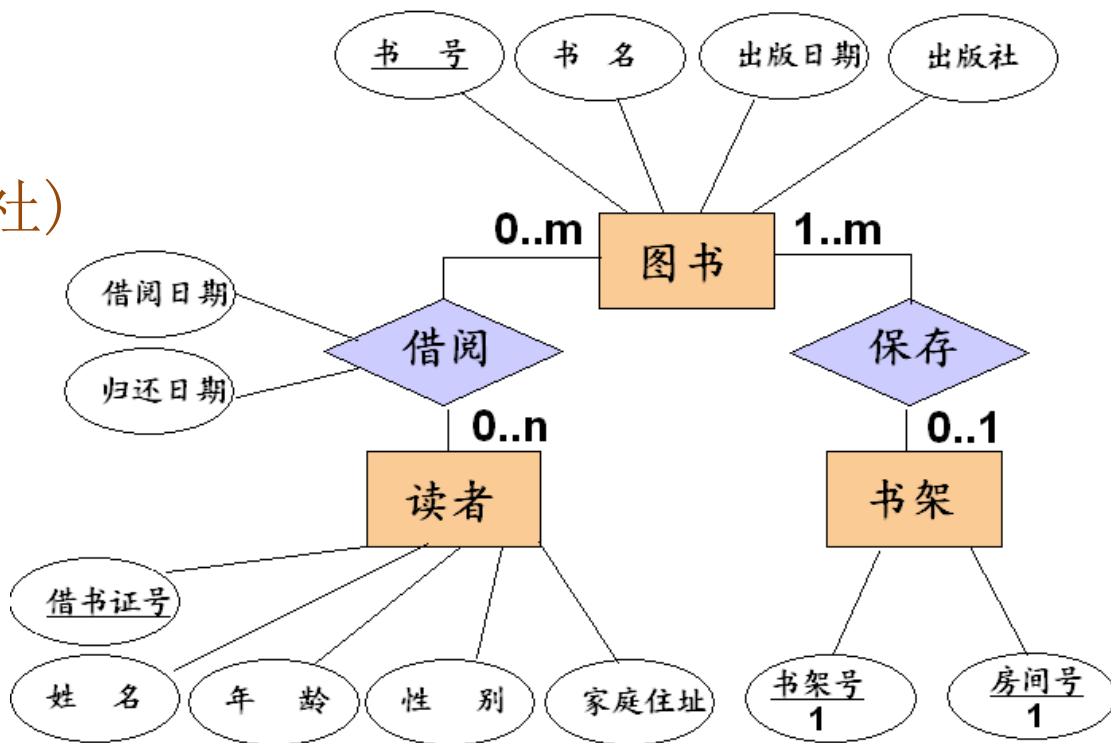
--E-R图向关系模式的转换

- E-R模型中的实体转换为关系
- E-R模型中的实体的属性转换为关系的属性

- 图书(书号, 书名, 出版日期, 出版社)

- 读者(借书证号, 姓名, 年龄, 性别, 家庭住址)

- 书架(书架号, 房间号)

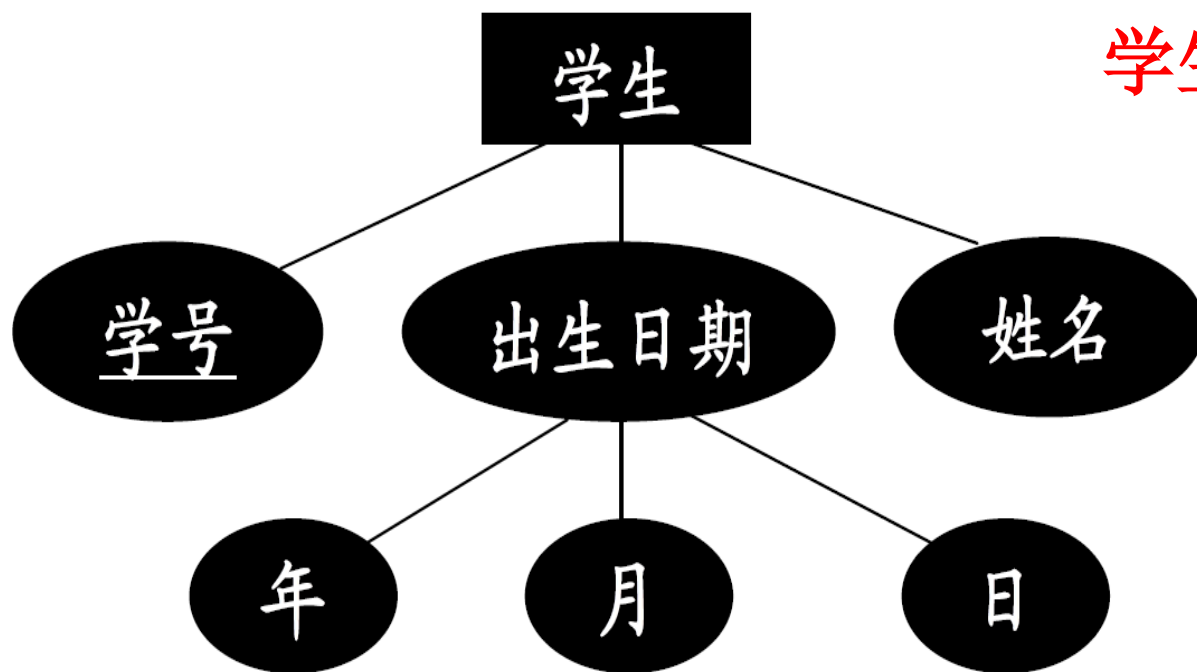


5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R图向关系模式的转换

➤ 复合属性:

将每个分量属性作为复合属性所在实体的属性



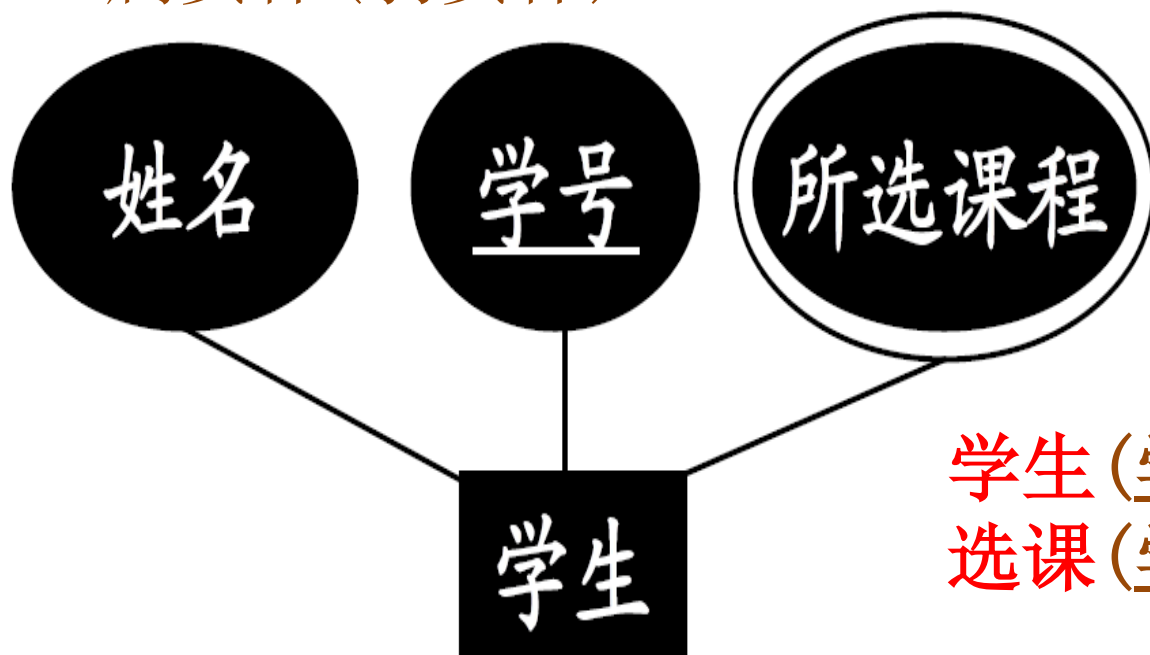
学生(学号, 姓名,
年, 月, 日)

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R图向关系模式的转换

➤ 多值属性:

将多值属性与所在实体的码一起组成一个新的从属实体(弱实体)



学生(学号, 姓名)

选课(学号, 所选课程号)

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R图向关系模式的转换

➤ 一对一联系:

- 若联系双方均部分参与(0..1),
则将联系定义为一个新的关系, 属性为参与双方的码。



- 若联系一方全部参与(1..1),
则将联系另一方的码作为全部参与一方的属性。



5.2 利用E-R模型/IDEF1X模型进行数据库设计

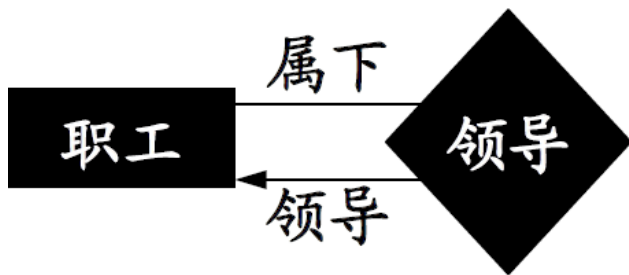
--E-R图向关系模式的转换

➤ 一对多联系:

- ▣ 将单方参与实体的码作为多方参与实体的属性



教师 (教工号, ...)
学生 (学生号, 学生名, 班主任教工号)



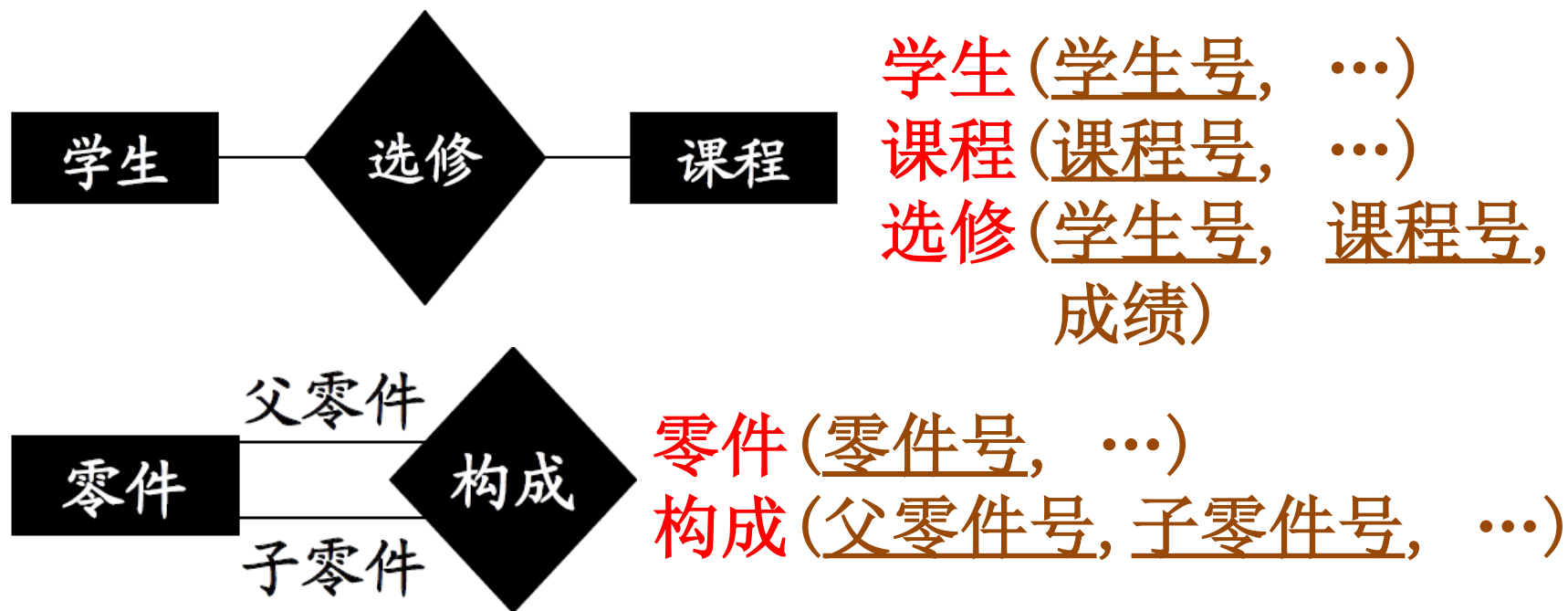
职工 (职工号, 职工名, 部门号, 领导职工号)

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R图向关系模式的转换

➤ 多对多联系:

- ▣ 将联系定义为新的关系，属性为参与双方的码

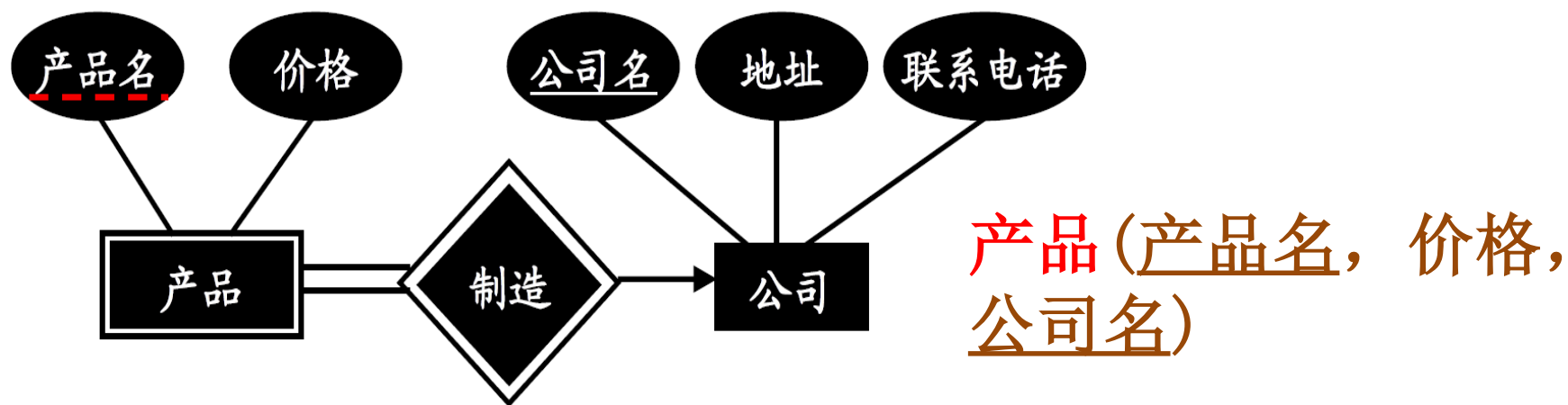


5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R图向关系模式的转换

➤ 弱实体(从属实体):

- 所对应关系的码由弱实体本身的分辩符再加上所依赖的强实体(独立实体)的码构成。



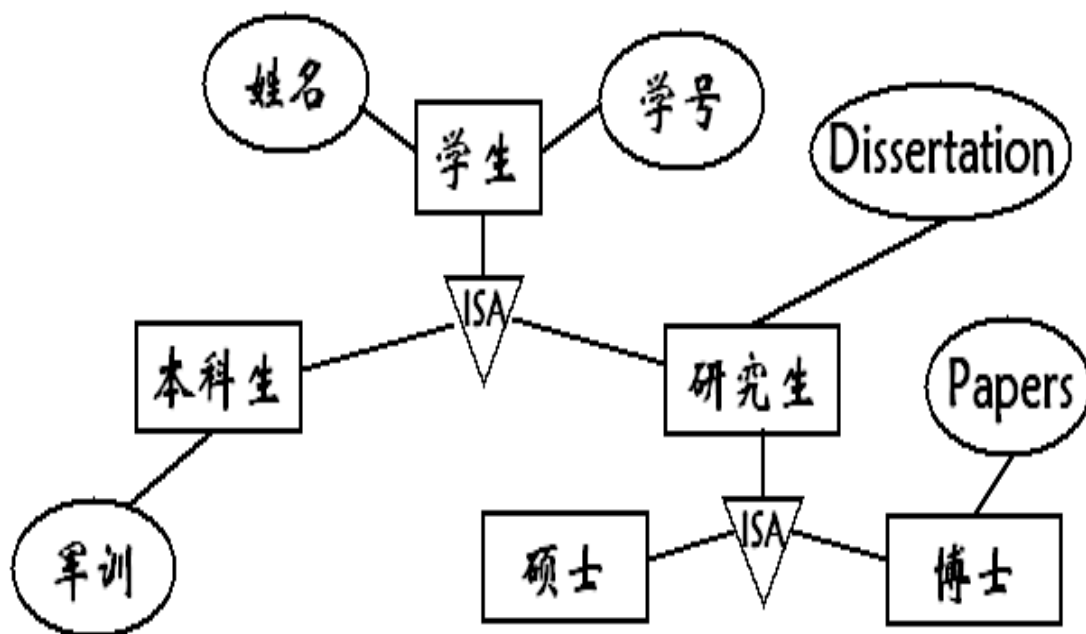
- 弱实体集(从属实体)与强实体集(独立实体)之间的联系已经在弱实体集所对应的关系中表示出来了。

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R图向关系模式的转换

➤ 特殊化与概括(子类与超类):

- 高层实体和低层实体分别转为不同的关系
- 低层实体所对应的关系包括高层实体的码



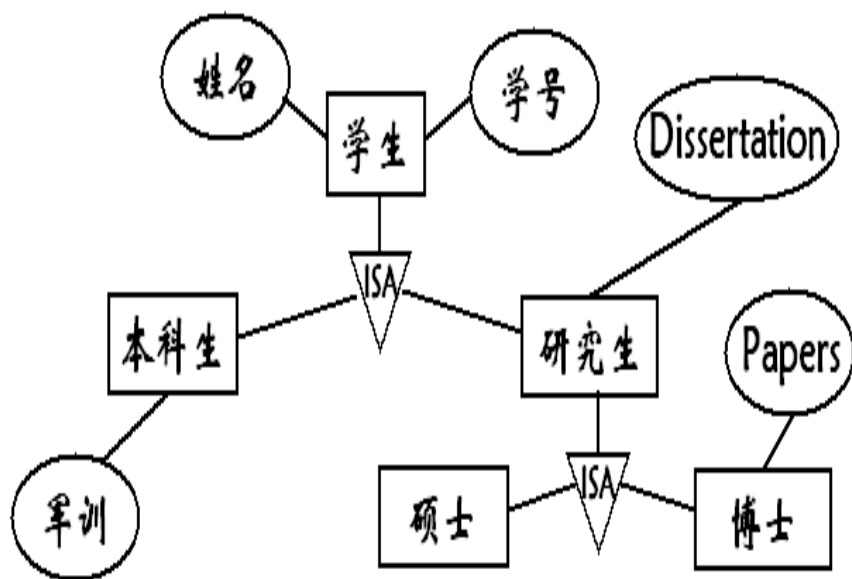
学生(姓名, 学号)
本科生(学号, 军训)
研究生(学号, 论文)

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R图向关系模式的转换

➤ 特殊化与概括(子类与超类):

- 如果概括是不相交并且是全部的，即一个高层实体最多并且只能属于一个低层实体，则可以不为高层实体建立关系码，低层实体所对应的关系包括上层实体的所有属性。



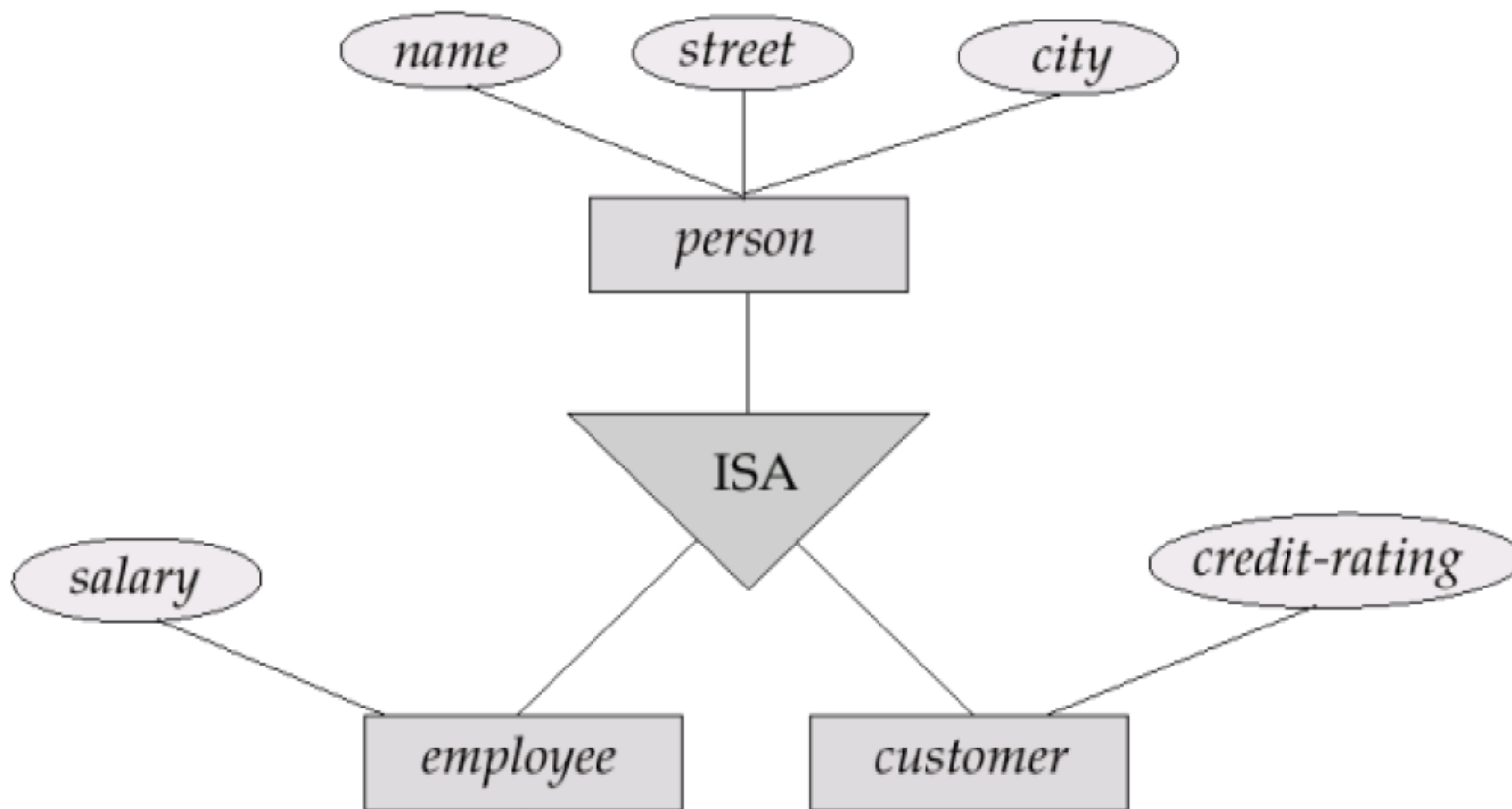
学生(姓名, 学号)
//如果概括是全部的, 甚至
无须创建“学生”关系。

本科生(姓名, 学号, 军训)
研究生(姓名, 学号, 论文)

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R图向关系模式的转换

➤ 例如，下面的E-R图如何转换：



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--E-R图向关系模式的转换

➤ 方案1

person (name, street, city)

customer (name, credit-rating)

employee (name, salary)

✓缺点：查询employee的地址需要访问两个表

➤ 方案2

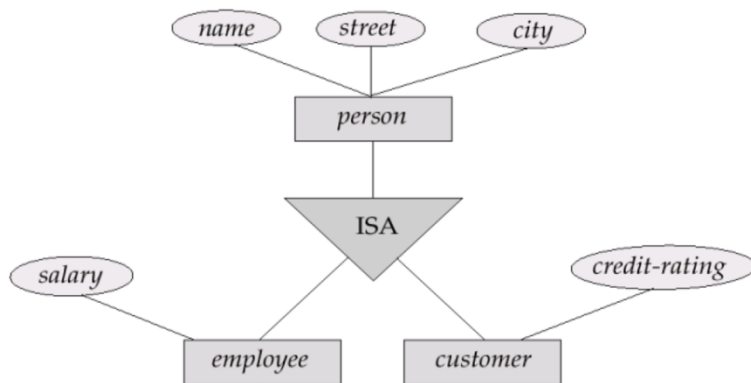
person (name, street, city)

customer (name, street, city, credit-rating)

employee (name, street, city, salary)

✓如果概括是全部的，无须创建person表

✓缺点：地址信息对同时是customer和employee的人存储两次



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--多元联系向关系模式转换及注意的问题

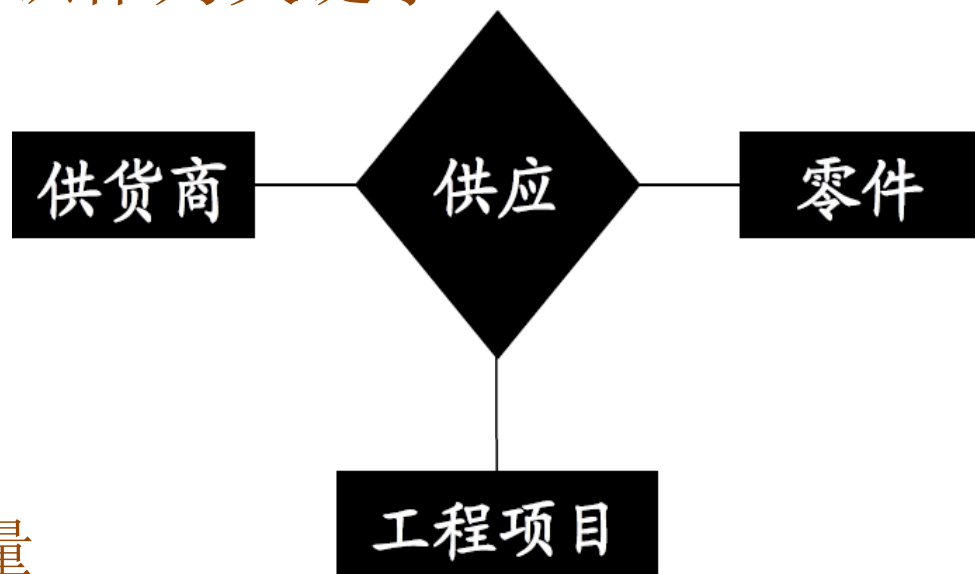
➤ 多元联系

- 可以通过继承参与联系的各个实体的码而形成新的关系；这些继承过来的码可以作为新关系的关键字。
- 也可以新增一个分辨标识作为关键字。

注意这两种转换的差异

1型转换：

供应(工程项目号, 供货商号,
零件号, 数量)



2型转换：

供应(条目号, 工程项目号,
供货商号, 零件号, 数量)

5.2 利用E-R模型/IDEF1X模型进行数据库设计

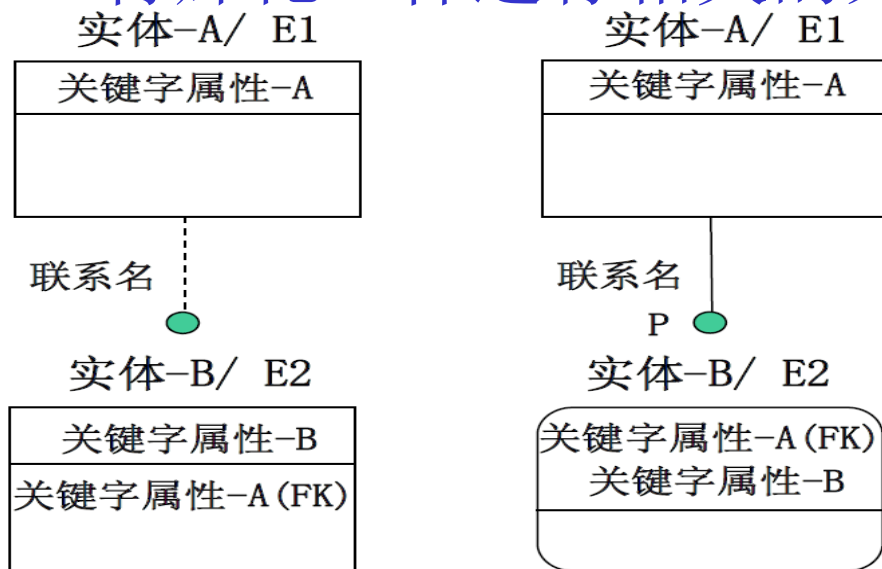
--多元联系向关系模式转换及注意的问题

- 多元联系更需注意分析参与联系的实体的**最小基数**和**最大基数**。
 - 如是否允许参与联系的多实体中有一个或多个实体不参与？
- 多元联系可以转换为多个二元联系进行处理。

5.2 利用E-R模型/IDEF1X模型进行数据库设计

--IDEF1X图向关系模式的转换

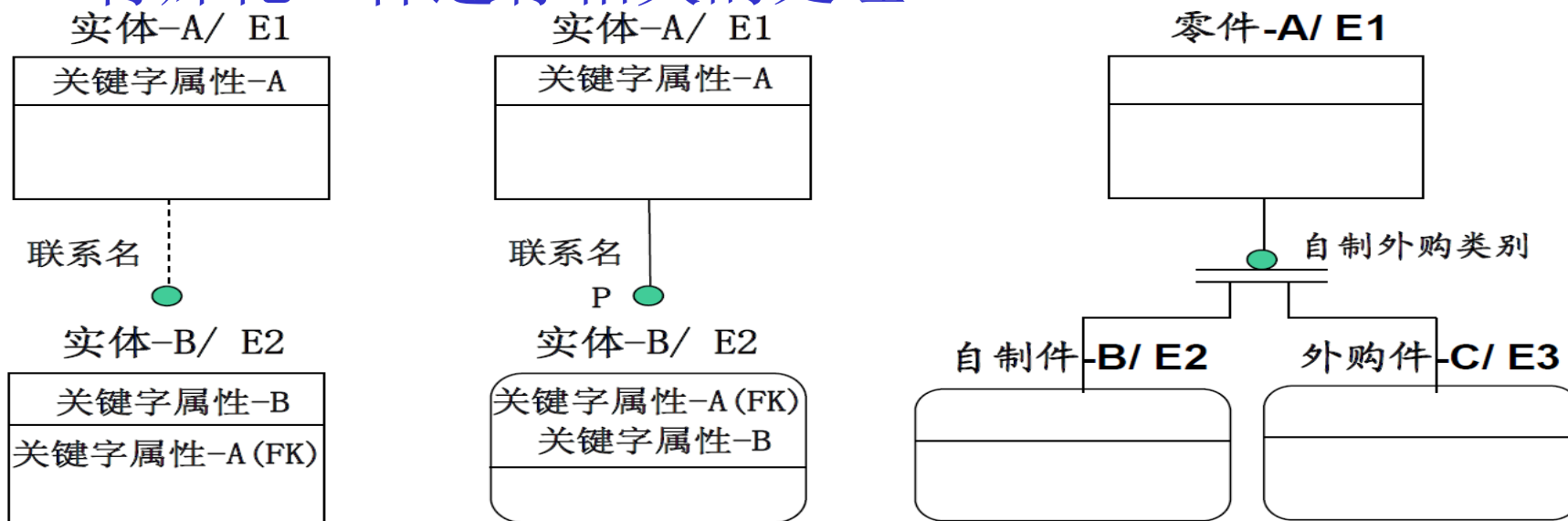
- IDEF1X图只需将实体转换成关系模式即可，而其联系的信息已经融入相关实体的关系描述中了。
- 对IDEF1X图的分类型联系，可以如E-R图中的概括和特殊化一样进行相关的处理。



5.2 利用E-R模型/IDEF1X模型进行数据库设计

--IDEF1X图向关系模式的转换

- IDEF1X图只需将实体转换成关系模式即可，而其联系的信息已经融入相关实体的关系描述中了。
- 对IDEF1X图的分类型联系，可以如E-R图中的概括和特殊化一样进行相关的处理。



第5章数据库设计

5.1 数据库设计

5.2 利用E-R模型/IDEF1X模型进行数据库设计

5.3 数据库设计正确性分析

--不正确设计数据库所引发的问题(冗余及处理异常)

--如何避免:

- ✓设计满足规范性, 由DBMS或数据库本身来保证
- ✓设计不满足规范性, 由使用者或应用程序员使用过程中加以注意

--什么是规范的数据库设计:

- ✓函数依赖、完全函数依赖与部分函数依赖, 传递函数依赖;
- ✓关系1NF, 2NF, 3NF, BCNF

5.4 数据库设计方法与设计过程

5.3 数据库设计正确性分析

--不正确设计数据库所引发的问题

➤ 冗余:

数据库中存在大量冗余

❑ 非受控冗余

✓例如，右侧数据库设计

❑ 非受控冗余问题

✓当数据发生改变时，如何
使冗余数据同步更新？
即更新异常问题

| 学号 | 姓名 | 班级 | 班主任 | 班主任 职称 |
|--------|----|------|-----|-----------|
| 110101 | 张一 | 1101 | 孙一 | 讲师 |
| 110102 | 张二 | 1101 | 孙一 | 讲师 |
| 110103 | 张三 | 1101 | 孙一 | 讲师 |
| 110104 | 张四 | 1101 | 孙一 | 讲师 |
| 110201 | 张五 | 1102 | 孙二 | 副教授 |
| 110202 | 张六 | 1102 | 孙二 | 副教授 |
| 110203 | 张七 | 1102 | 孙二 | 副教授 |
| 110204 | 张八 | 1102 | 孙二 | 副教授 |



5.3 数据库设计正确性分析

--不正确设计数据库所引发的问题

▣ 受控冗余:

- ✓如Table中的外键
(继承其他Table中的键值)

| 班级 | 班主任 | 班主任 职称 |
|------|-----|-----------|
| 1101 | 孙一 | 讲师 |
| 1102 | 孙二 | 副教授 |

| 学号 | 姓名 | 班级 |
|--------|----|------|
| 110101 | 张一 | 1101 |
| 110102 | 张二 | 1101 |
| 110103 | 张三 | 1101 |
| 110104 | 张四 | 1101 |
| 110201 | 张五 | 1102 |
| 110202 | 张六 | 1102 |
| 110203 | 张七 | 1102 |
| 110204 | 张八 | 1102 |

5.3 数据库设计正确性分析

--不正确设计数据库所引发的问题

➤ 冗余引发的异常

□ 插入异常:

当一名新同学入学时，尚未指定系，则因系的有关信息不完整，便无法输入到数据库中，如图所示。

□ 删除异常:

当四系所有同学被删除后，则四系的有关信息则随之丢失。

| 学号 | 姓名 | 系别 | 系名称 | 系主任 |
|--------|----|----|-----|-----|
| 110101 | 张一 | 三系 | 计算机 | 刘一 |
| 110102 | 张二 | 三系 | 计算机 | 刘一 |
| 110103 | 张三 | 三系 | 计算机 | 刘一 |
| 110104 | 张四 | 三系 | 计算机 | 刘一 |
| 110201 | 张五 | 四系 | 船舶 | 钱一 |
| 110202 | 张六 | 四系 | 船舶 | 钱一 |
| 110203 | 张七 | 四系 | 船舶 | 钱一 |
| 110204 | 张八 | 四系 | 船舶 | 钱一 |



5.3 数据库设计正确性分析

--不正确设计数据库所引发的问题

➤ 插入异常和删除异常的另外的例子

职工表(职工, 级别, 工资)

- ❑ 若删除张一, 则级别4信息丢失
- ❑ 若无人的级别是7, 则级别7的工资信息无法在表中反映
- ❑ 若有人级别未定, 则不能插入该表中?

| 职工 | 级别 | 工资 |
|----|----|------|
| 张一 | 4 | 800 |
| 张二 | 6 | 2000 |
| 张三 | 5 | 1000 |
| 张四 | 6 | 2000 |
| | | |

| 级别 | 工资 |
|----|------|
| 7 | 3000 |


| 职工 | 级别 | 工资 |
|----|------|------|
| 钱三 | (未定) | (未定) |

5.3 数据库设计正确性分析

--如何避免

➤ 设计满足规范性，由DBMS或数据库本身来保证

➤ 设计不满足规范性，
由使用者或应用程序员
使用过程中加以注意



| 班级 | 班主任 | 班主任职称 |
|------|-----|-------|
| 1101 | 孙一 | 讲师 |
| 1102 | 孙二 | 副教授 |



| 学号 | 姓名 | 班级 | 班主任 | 班主任职称 |
|--------|----|------|-----|-------|
| 110101 | 张一 | 1101 | 孙一 | 讲师 |
| 110102 | 张二 | 1101 | 孙一 | 讲师 |
| 110201 | 张五 | 1102 | 孙二 | 副教授 |
| 110202 | 张六 | 1102 | 孙二 | 副教授 |

| 学号 | 姓名 | 班级 |
|--------|----|------|
| 110101 | 张一 | 1101 |
| 110102 | 张二 | 1101 |
| 110201 | 张五 | 1102 |
| 110202 | 张六 | 1102 |

5.3 数据库设计正确性分析

--具体解决之道:

分解、分解、再分解

职工表
(职工, 级别, 工资)

| 职工 | 级别 | 工资 |
|----|----|------|
| 张一 | 4 | 800 |
| 张二 | 6 | 1500 |
| 张三 | 5 | 1000 |
| 张四 | 6 | 1500 |

分解

职工的工资级别表

| 职工 | 级别 |
|----|----|
| 张一 | 4 |
| 张二 | 6 |
| 张三 | 5 |
| 张四 | 6 |

工资级别信息表

| 级别 | 工资 |
|----|------|
| 4 | 800 |
| 5 | 1000 |
| 6 | 1500 |
| 7 | 2000 |

5.3 数据库设计正确性分析

--具体解决之道:

分解、分解、再分解

- 解决了插入异常: 7级工资可存储在工资表中, 不依赖于职工而独立存在, 需要时可插入。
- 解决了删除异常: 4级工资存储在工资表中, 即使职工张一被删除, 4级工资仍然存在。
- 解决了数据冗余: 5, 6级工资数额在工资表中仅存储一次。
- 解决了更新异常: 5级工资数额发生变化, 只需在工资表中更新一次即可。

5.3 数据库设计正确性分析

--问题的根源：
不良的函数依赖

职工表
(职工, 级别, 工资)

| 职工 | 级别 | 工资 |
|----|----|------|
| 张一 | 4 | 800 |
| 张二 | 6 | 1500 |
| 张三 | 5 | 1000 |
| 张四 | 6 | 1500 |

职工→级别
级别→工资

蕴含

职工→工资

工资依赖于职工

职工的工资级别表

| 职工 | 级别 |
|----|----|
| 张一 | 4 |
| 张二 | 6 |
| 张三 | 5 |
| 张四 | 6 |

职工→级别

工资不再依赖于职工

工资级别信息表

| 级别 | 工资 |
|----|------|
| 4 | 800 |
| 5 | 1000 |
| 6 | 1500 |
| 7 | 2000 |

级别→工资

分解

5.3 数据库设计正确性分析

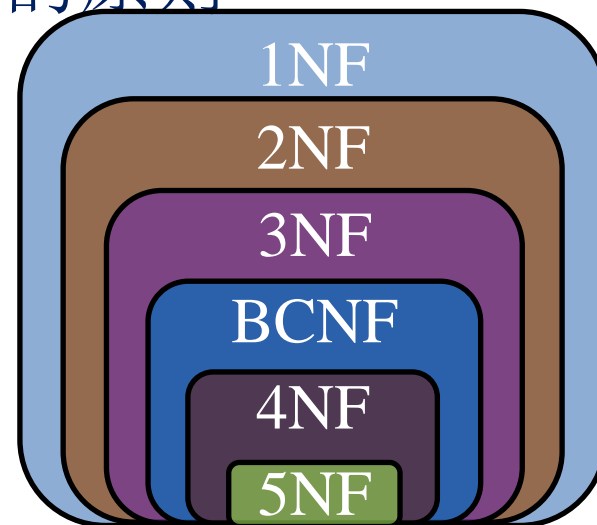
--规范的数据库设计

➤ 明确数据库表中的属性在取值方面存在的依存关系

- ❑ 函数依赖;
- ❑ 完全函数依赖与部分函数依赖;
- ❑ 传递函数依赖;

➤ 明确数据库设计过程中应遵循的原则

- ❑ 关系第1范式原则 (1NF)
- ❑ 关系第2范式原则 (2NF)
- ❑ 关系第3范式原则 (3NF)
- ❑ 关系Boyce-Codd范式原则 (BCNF)
- ❑ 4NF、5NF等



第5章数据库设计

5.1 数据库设计

5.2 利用E-R模型/IDEF1X模型进行数据库设计

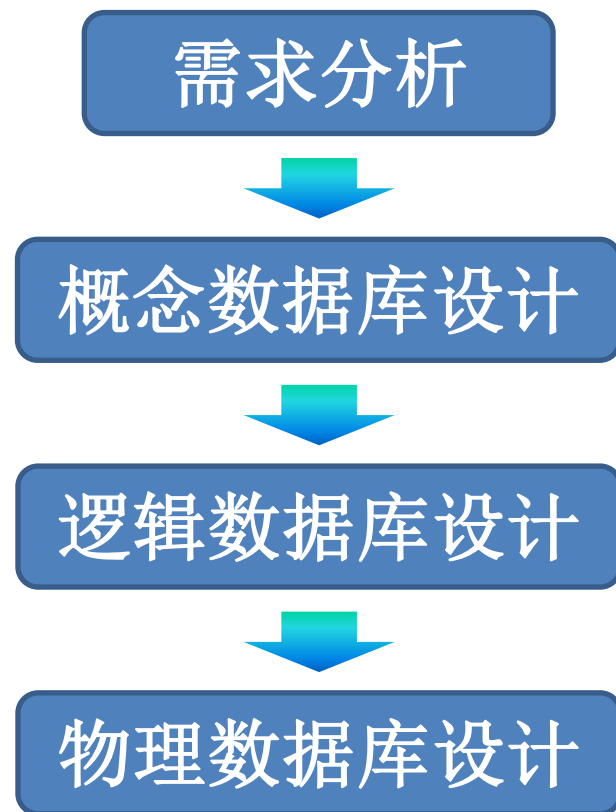
5.3 数据库设计正确性分析

5.4 数据库设计方法与设计过程

—数据库设计过程与设计方法

5.4 数据库设计方法与设计过程

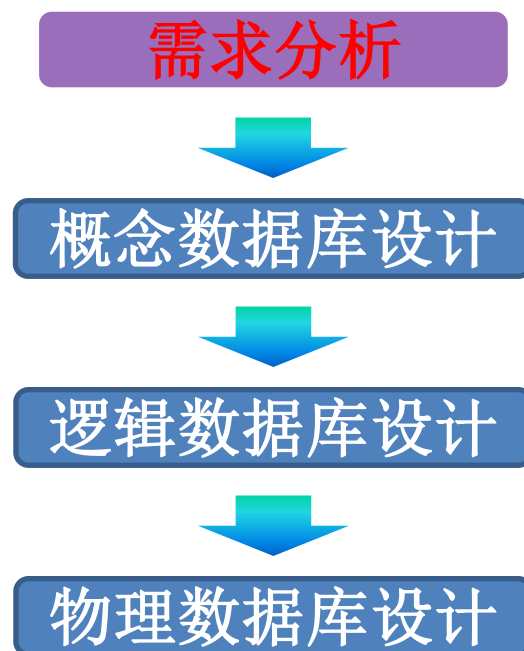
--数据库设计过程



5.4 数据库设计方法与设计过程

--需求分析阶段

- 目标：理解企业、理解企业业务过程与数据处理流程、理解数据处理的性能需求。
- 提交物：需求分析报告
- 使以下内容清楚：
 - ❑ 企业各种日常管理信息表/报表
 - ❑ 形成各种报表的基础数据表
 - ❑ 各种数据表之间的处理关系
 - ❑ 围绕数据表的业务处理关系
 - ❑ 尚未实施但未来可能实施的需求
- 形成数据库设计的“源”清单和“属性”清单以及相关的详细描述，尤其是注意业务规则与属性处理规则
- 结合数据流图等辅助分析与理解



5.4 数据库设计方法与设计过程

-- “源” 清单

| | | |
|-------|---------|--|
| 总序号 | 源表名称 | |
| | 处理及存档要求 | 注意收集和理解： 1. 业务规则及其在表的处理方面的体现 2. 不仅报表、单据是源，企业的查询需求与管理需求等也是源 |
| | 源表属性名表 | |
| 总序号 | 源表名称 | |
| | 处理及存档要求 | |
| | 源表属性名表 | |
| | | |

5.4 数据库设计方法与设计过程

-- “属性” 清单

| 局部 序号 | 属性名称 | | 全局统一 序号 | 属性统一名称 | | 对应 “源” 序号 |
|----------|--|----|------------|--------|----|--------------|
| | 中文 | 英文 | | 中文 | 英文 | |
| | | | | | | |
| | <p>注意命名：</p> <p>1. 命名要规范，并且要含义明确</p> <p>2. 尤其要注意类似于“数量”这样的多含义属性：</p> <p>✓ 比如“计划数量”、“采购数量”、“到货数量”、“装配数量”、“完工数量”、“销售数量”、“发货数量”</p> | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

5.4 数据库设计方法与设计过程

-- “属性”定义表

| 统一序号 | 属性含义 | 属性建库要求(类型/长度/取值/含义) |
|------|------|---------------------|
| | | |

注意

1. 准确理解对属性的业务规则，尤其是约束规则：

✓ 比如：成绩只能取“优秀”、“良好”、“中等”、“及格”、“不及格”这五个值；

✓ 再比如：工资只能升不能降，年龄大于15且小于23岁等；

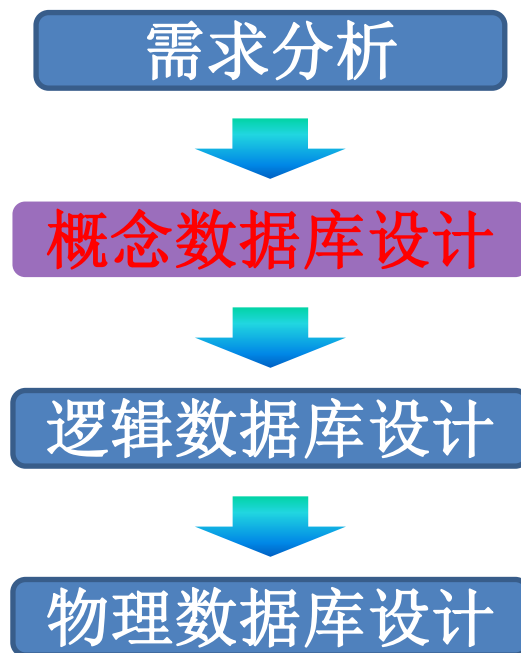
2. 编码属性的编码规则；

3. 分类属性的分类标准及分类值等。

5.4 数据库设计方法与设计过程

--概念数据库设计阶段

- 目标：进一步深入理解企业，对信息源进行抽象，发现信息(属性)之间的内在本质联系，这些本质联系可能隐藏于需求分析得到的信息源中。
- 提交物：概念数据库设计报告
- 使以下内容清楚：
 - ❑ 各种实体的发现、划分和定义
 - ❑ 各种实体属性的发现、分析和定义
 - ❑ 各种实体联系的发现、分析和定义
 - ❑ 外部视图和概念视图的定义需求
- 用形式化方法，如E-R模型/IDEF1X模型给出描述；不仅绘制出来，而且绘制正确；应用依赖关系进行分析。
- 不仅使自己清楚，而且要使相关人员理解无误。

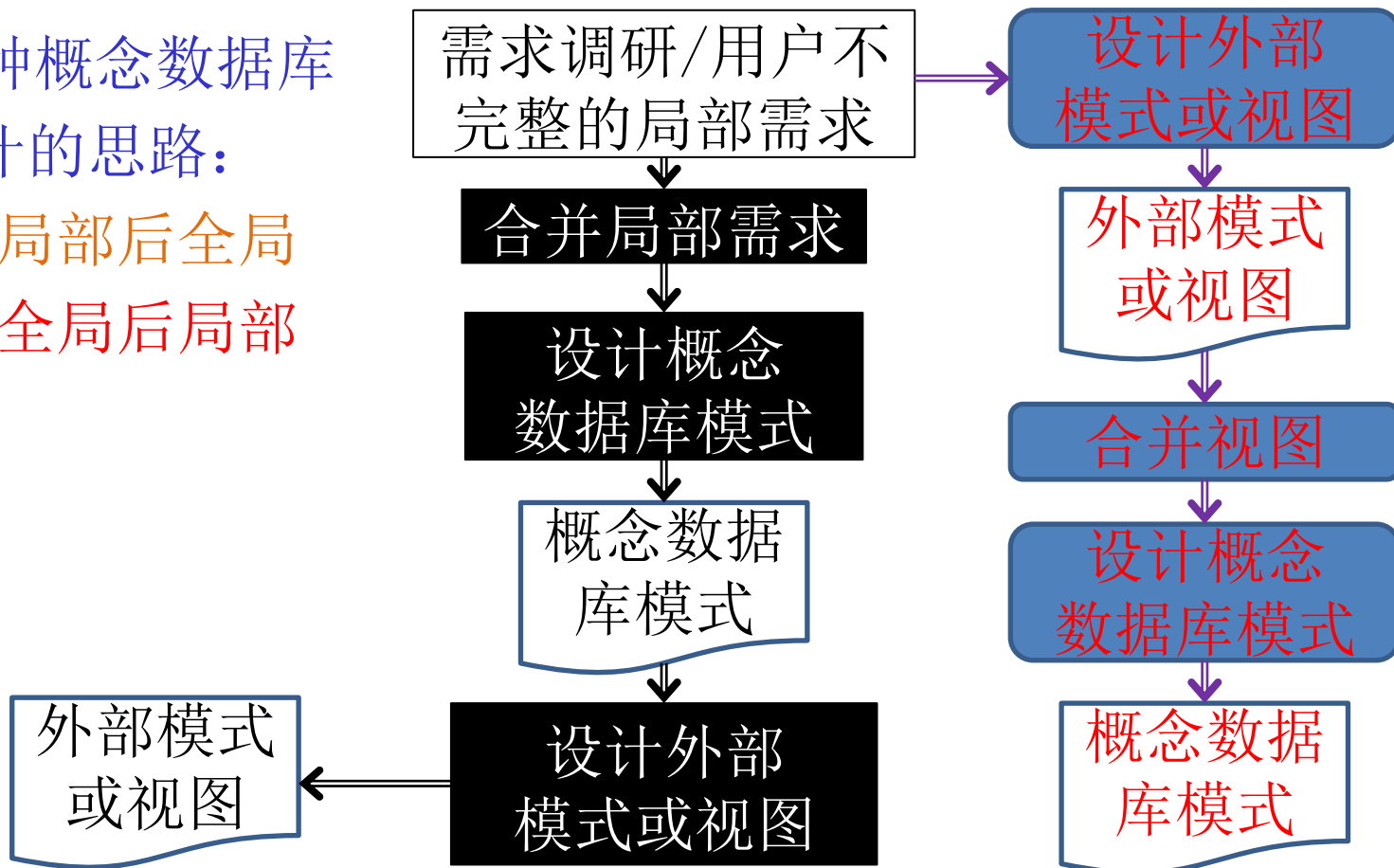


5.4 数据库设计方法与设计过程

--概念数据库设计方法

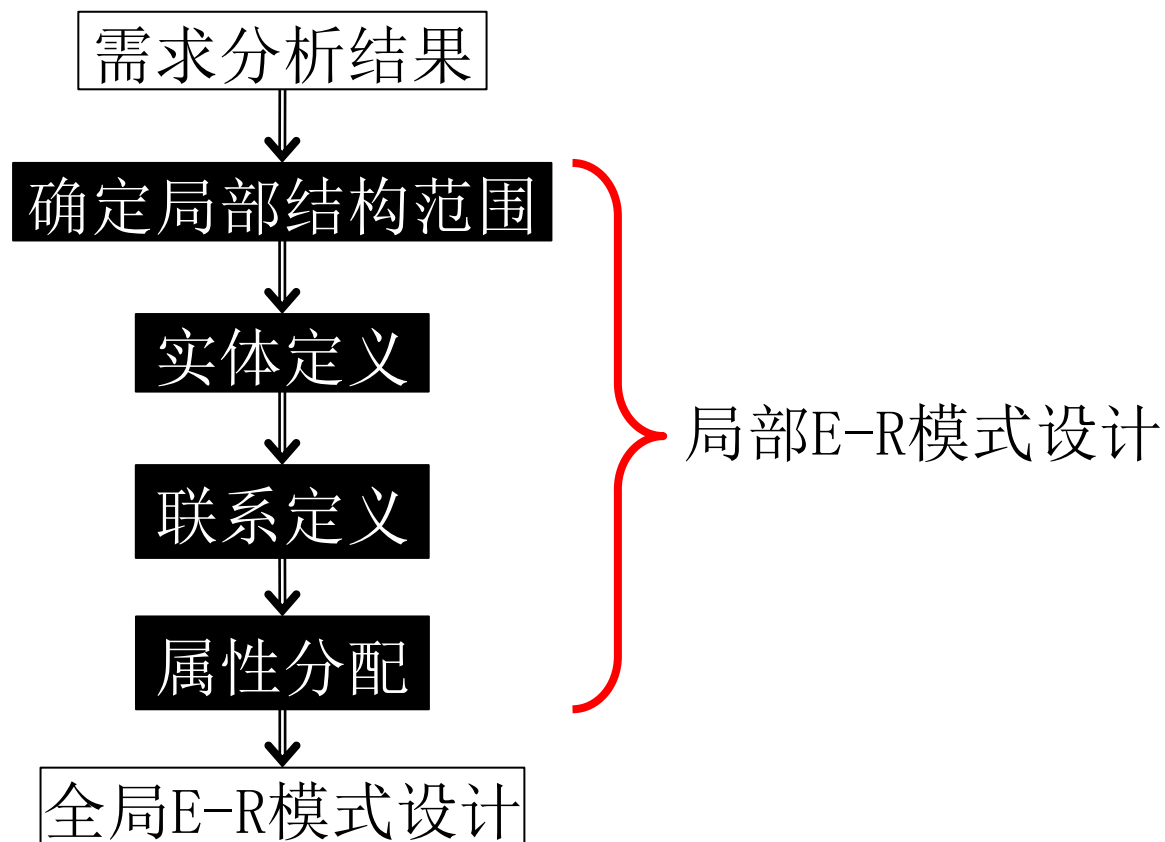
➤ 两种概念数据库设计的思路:

- 先局部后全局
- 先全局后局部



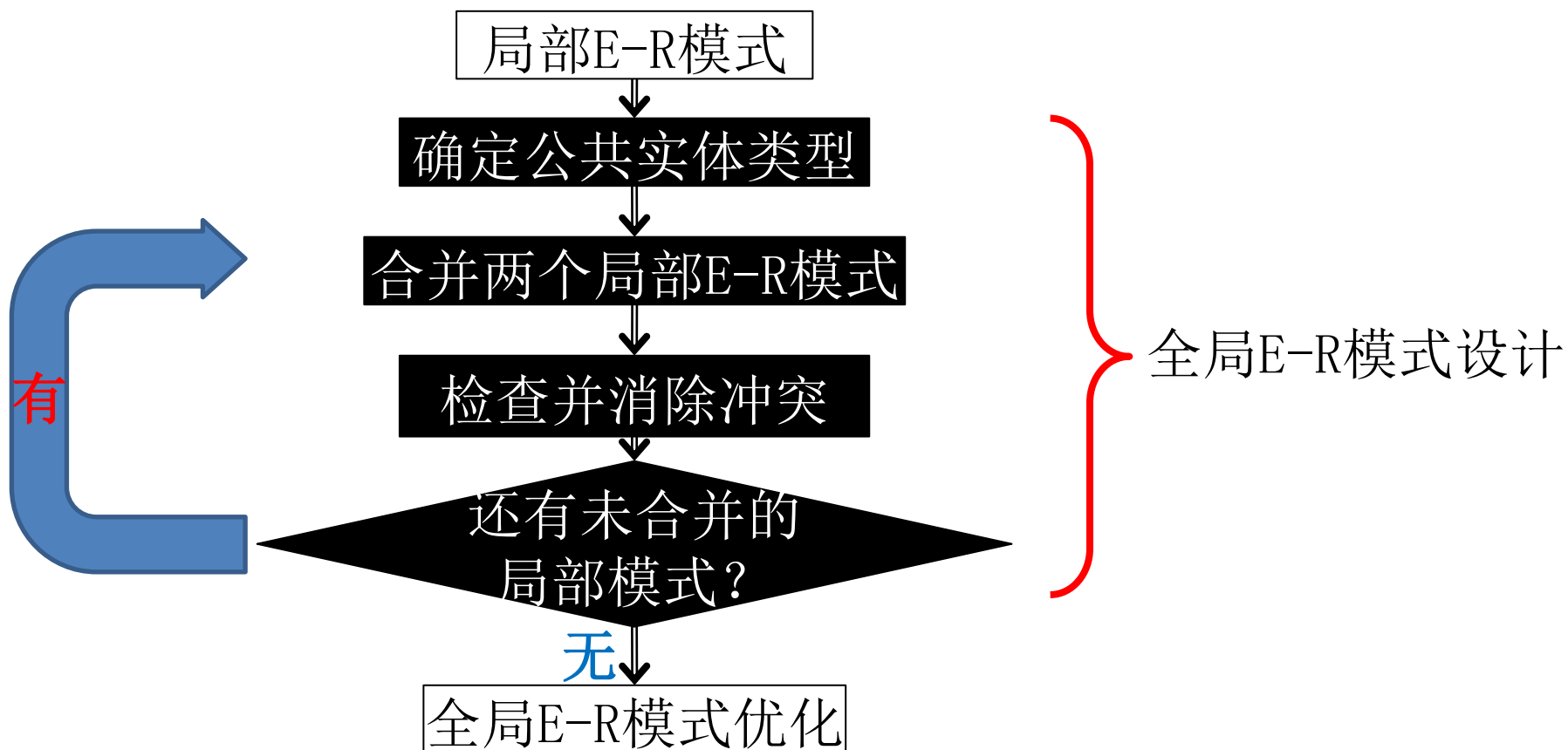
5.4 数据库设计方法与设计过程

--局部E-R模式设计



5.4 数据库设计方法与设计过程

--全局E-R模式设计



5.4 数据库设计方法与设计过程

--消除冲突

➤ 属性冲突

- 属性域的冲突：属性的类型、取值范围不同

- ✓ 如不同学校的学号编码方式不同

- 属性取值单位冲突

- ✓ 如重量分别采用磅、千克

➤ 属性冲突的解决方法

- 通常用讨论、协商等行政手段加以解决



5.4 数据库设计方法与设计过程

--消除冲突

➤ 结构冲突

- 同一对象在不同应用中的抽象不同
 - ✓ 如职工在某应用中是实体，在另一应用中则抽象为属性
- 同一实体在不同E-R图中属性组成不同
 - ✓ 例如不同学校计算机系的课程不同
- 实体之间的联系在不同视图中呈现不同的类型
 - ✓ 例如实体E1与E2在局部应用A中是多对多联系，而在局部应用B中是一对多联系

➤ 结构冲突的解决方法

- 根据应用语义对实体联系的类型进行综合或调整

5.4 数据库设计方法与设计过程

--消除冲突

➤ 命名冲突

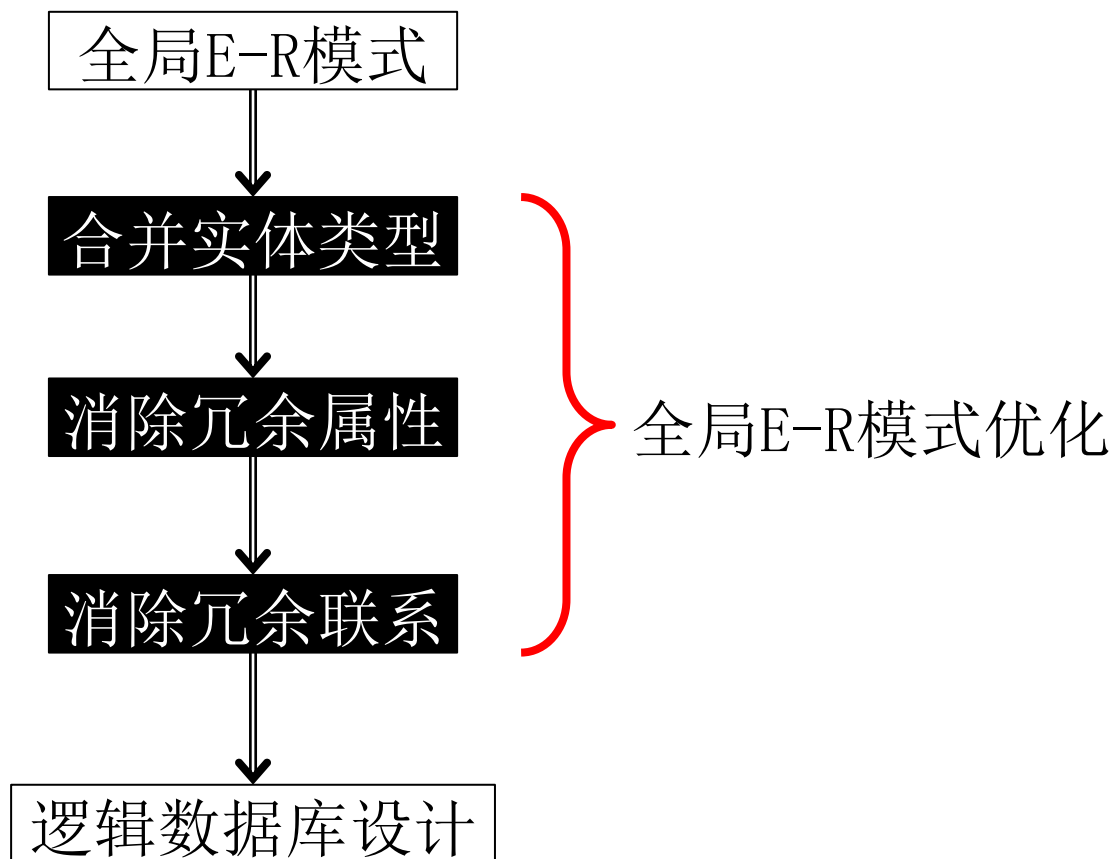
- ❑ 同名异义：不同意义的对象在不同的局部应用中具有相同的名字
 - ✓ 例如局部应用A中将教室称为房间，局部应用B中将学生宿舍称为房间
- ❑ 异名同义：同一意义的对象在不同的局部应用中具有不同的名字
 - ✓ 例如有的部门把教科书称为课本，有的部门则把教科书称为教材

➤ 命名冲突的解决方法

- ❑ 通常用讨论、协商等行政手段加以解决

5.4 数据库设计方法与设计过程

--全局E-R模式优化



5.4 数据库设计方法与设计过程

--全局E-R模式优化

- 冗余的数据是指可由基本数据导出的数据。
- 冗余的联系是指可由其他联系导出的联系。
- 冗余数据和冗余联系容易破坏数据库的完整性，给数据库维护增加困难
- 并不是所有的冗余数据与冗余联系都必须加以消除，有时为了提高某些应用的效率，不得不以冗余信息作为代价。
- 设计数据库概念结构时，哪些冗余信息必须消除，哪些冗余信息允许存在，需要根据用户的整体需求来确定。
- 消除不必要的冗余后的初步E-R图称为基本E-R图

5.4 数据库设计方法与设计过程

-- “实体” 清单

| 局部 序号 | 实体名称 | | 统一 序号 | 实体统一名称 | | 对应 “源” 序号 |
|----------|------|-------------------|----------|--------|----|--------------|
| | 中文 | 英文 | | 中文 | 英文 | |
| | | | | | | |
| | | | | | | |
| | | 注意： 实体的识别与精确命名 | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

5.4 数据库设计方法与设计过程

-- “实体”定义表

| 统一序号 | 实体含义 | 实体建库要求 |
|------|--|--------|
| | | |
| | | |
| | | |
| | 注意 1. 实体的规范化与非规范化的折中 2. 注意非规范化的处理要求 | |
| | | |
| | | |
| | | |
| | | |

5.4 数据库设计方法与设计过程

-- “实体-联系” 矩阵

| | | 实体A | 实体B | 实体C | 实体D | 实体E | 实体F | | | | | |
|-----|----|-----|-----|-----|-----|-----|-----|---|---|---|----|----|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 实体A | 1 | | | | | | | | | | | |
| 实体B | 2 | X | | | | | | | | | | |
| 实体C | 3 | | | | | | X | | | | | |
| 实体D | 4 | | | | | | | | | | | |
| 实体E | 5 | | X | | | | | | | | | |
| 实体F | 6 | | | | | | | | | | | |
| | 7 | | | | | | | | | | | |
| | 8 | | | | | | | | | | | |
| | 9 | | | | | | | | | | | |
| | 10 | | | | | | | | | | | |
| | 11 | | | | | | | | | | | |

5.4 数据库设计方法与设计过程

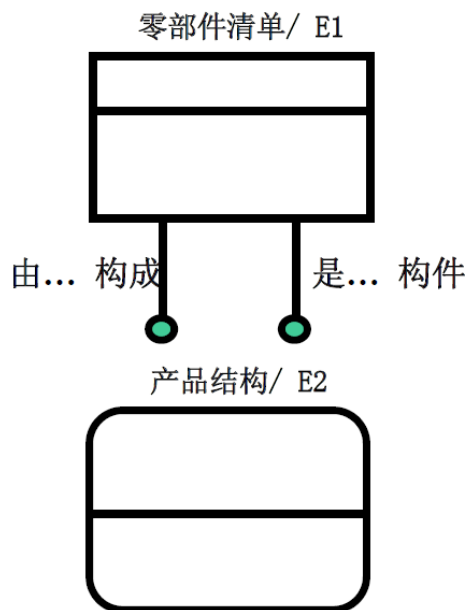
-- “实体-属性” 矩阵

| | | 属性A | 属性B | 属性C | 属性D | 属性E | 属性F | 属性G | 属性H | 属性I | 属性J | 属性K | 属性L | 属性M | 属性N | 属性O | | |
|-----|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|--|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | | |
| 实体A | 1 | | OK | | | | | | I | | | | | | | | | |
| 实体B | 2 | I | | IK | | O | | | K | | | O | | | | | | |
| 实体C | 3 | | | | I | | I | | | | | | | | | | | |
| 实体D | 4 | | | | | | | | | | | | | | | | | |
| 实体E | 5 | | K | | | | | O | | | | | | | | | | |
| 实体F | 6 | | | O | | | K | | | | | | | O | | | | |
| | 7 | | | | | | | | | | | | | | | | | |
| | 8 | | | | | | | | | | | | | | | | | |

注释：O 实体的占有属性；I 实体的继承属性；K 实体的主键属性。

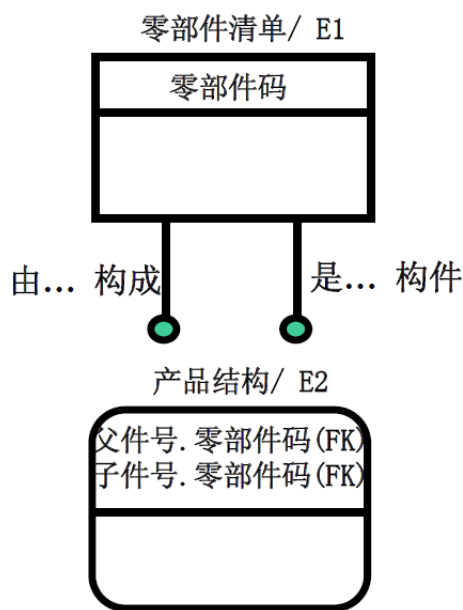
5.4 数据库设计方法与设计过程

--逐渐细化地绘制实体级图、键级图及完整图



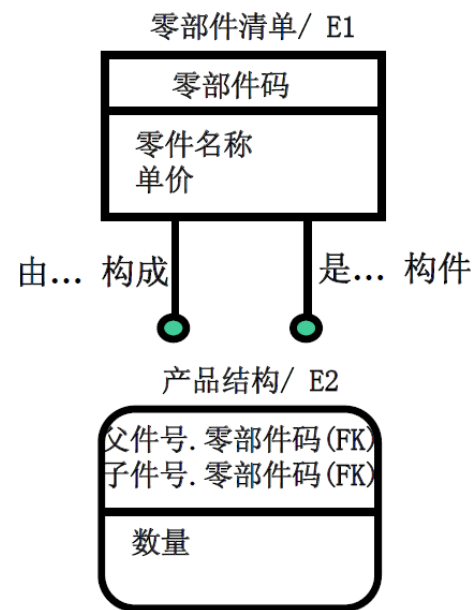
实体级图

以实体为建模单
位的**IDEF1x**图



键级图

以实体为建模单位并且
标注键属性的**IDEF1x**图



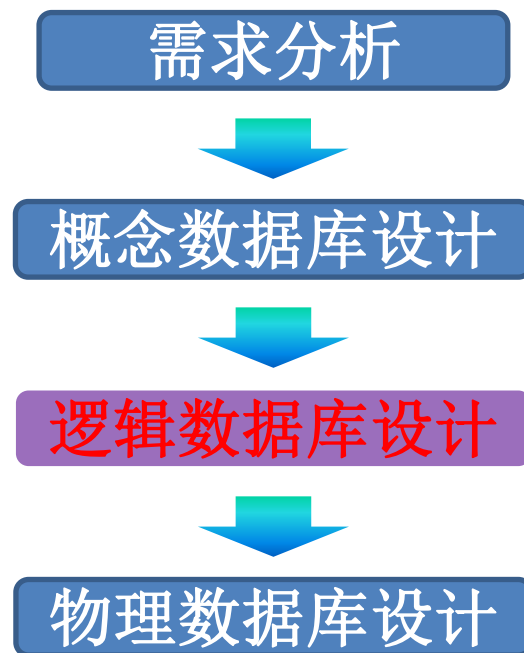
完整图

以实体为建模单位并且
标注所有属性的**IDEF1x**图

5.4 数据库设计方法与设计过程

--逻辑数据库设计阶段

- 目标：用指定DBMS要求的模式描述方法，给出概念数据库的逻辑模式描述。
- 提交物：逻辑数据库设计报告
- 使以下内容清楚：
 - ❑ 将E-R/IDEF1X转换成逻辑模式
 - ❑ 遵循关系范式的设计原则
 - ❑ 也要注意折中，但折中时需要提示应用开发者或使用用户可能存在的问题
 - ❑ 外模式和概念模式的定义
- 用关系模型、网状模型或层次模型规定的模式进行描述，按照属性定义要求进行模式描述。



5.4 数据库设计方法与设计过程

--逻辑数据库设计方法

- 将前一阶段的概念数据库(如用E-R图/IDEF1X描述的)转换成逻辑数据模式
- 识别依赖关系, 并定义完整性约束(如外键、主键等)
- 用相交实体/相关实体消除M:N联系
- 消除复杂的联系, 如多元联系(可通过插入另一个实体的办法)
 - ❑ 多元联系可能引发关键字冲突, 更需注意最小基数和最大基数, 由其是有最小基数为0的实体时。

5.4 数据库设计方法与设计过程

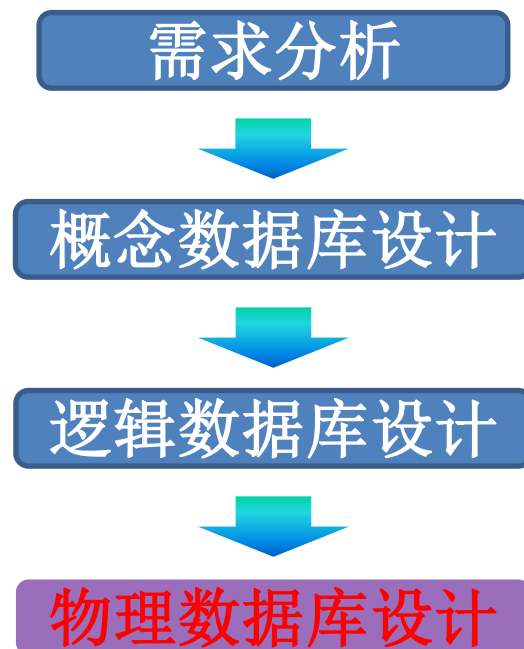
--逻辑数据库设计方法

- 消除多值属性与复合属性
- 消除冗余联系
- 用关系范式规则进行检查
- 与用户进行确认或与应用程序员 (Client) 进行确认

5.4 数据库设计方法与设计过程

--物理数据库设计阶段

- 目标：结合指定DBMS物理数据库管理方法，给出概念数据库的物理模式描述。
- 提交物：物理数据库设计报告
- 使以下内容清楚：
 - DBMS选型
 - 确定数据库的存储结构，文件类型。
如定长文件、不定长文件；堆文件、散列文件或B-Tree文件等
 - 用Triggers, 设计一些完整性控制约束
 - 确定数据库的高效访问方式
(索引访问，直接访问……)
 - 评估磁盘空间需求



5.4 数据库设计方法与设计过程

--物理数据库设计阶段

- 目标：结合指定DBMS物理数据库管理方法，给出概念数据库的物理模式描述。
- 提交物：物理数据库设计报告
- 使以下内容清楚：
 - ❑ 设计用户视图及访问控制规则，以进行安全性控制
 - ❑ 建立索引
 - ❑ 设计使数据库运行达到最佳效率的一些措施
 - ❑ 设计备份和恢复的步骤
- 理解Oracle、Sybase或其他DBMS的物理数据库管理方式，这是数据库管理员(DBA)的基本责任。

