

7.3 概念结构设计

7.3.1 概念结构

7.3.2 E-R模型

*7.3.3 扩展的E-R模型

*7.3.4 UML

7.3.5 概念结构设计



7.3.3 扩展的E-R模型

❖ 为什么要对E-R模型进行扩展？

为了增强和丰富基本E-R模型的表达能力。

❖ 进行了哪些扩展？我们介绍主要的3种扩展：

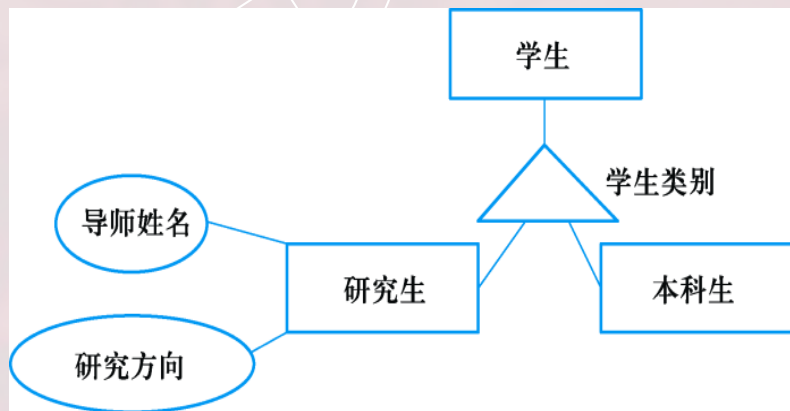
1. ISA联系
2. 基数约束
3. PART-OF联系



1. ISA 联系

1. ISA联系

- ❖ 有的实体型是某个实体型的子类型，这种父类-子类联系称为ISA联系，表示“is a”语义。用 \triangle 表示。
- ❖ ISA联系的性质：子类继承了父类的所有属性，子类也可以有自己的属性。



学生的2个子类



1. ISA 联系

(1) 分类属性

- ❖ 分类属性是父实体的一个属性
- ❖ 分类属性的值把父实体型中的实体分派到子实体型中

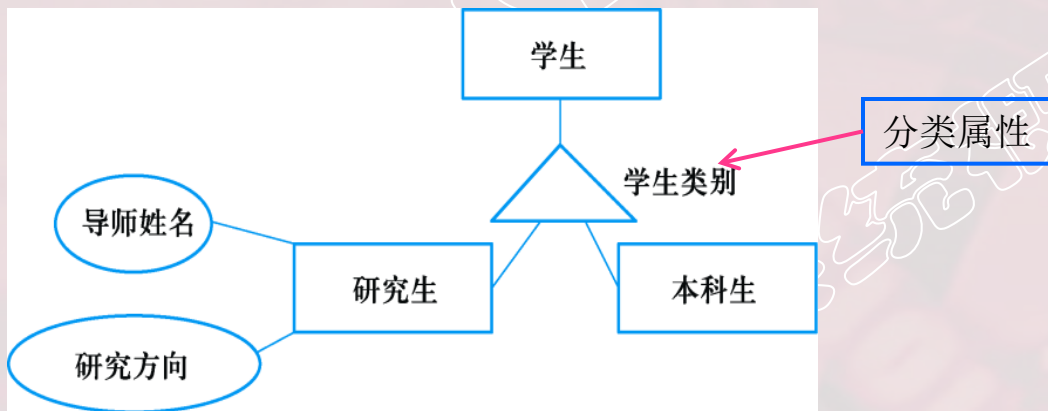


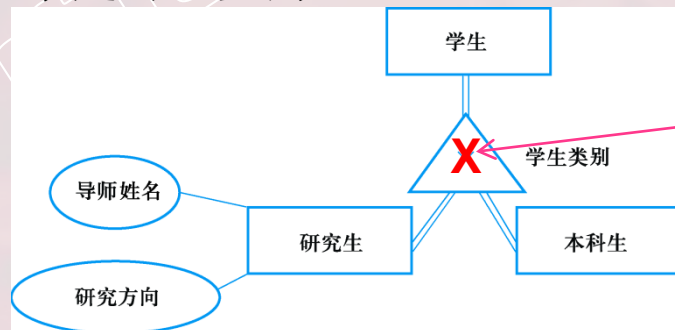
图7.12 学生的2个子类和分类属性



1. ISA 联系

(2) 不相交约束与可重叠约束

- 不相交约束：描述父类中的一个实体不能同时属于多个子类中的实体集。即一个父类中的实体最多属于一个子类实体集。
- 用ISA联系符号 三角形的一个叉号“X”来表示。
- 可重叠约束：父类中的一个实体能同时属于多个子类中的实体集。子类符号中没有叉号表示是可重叠的。



不相交约束

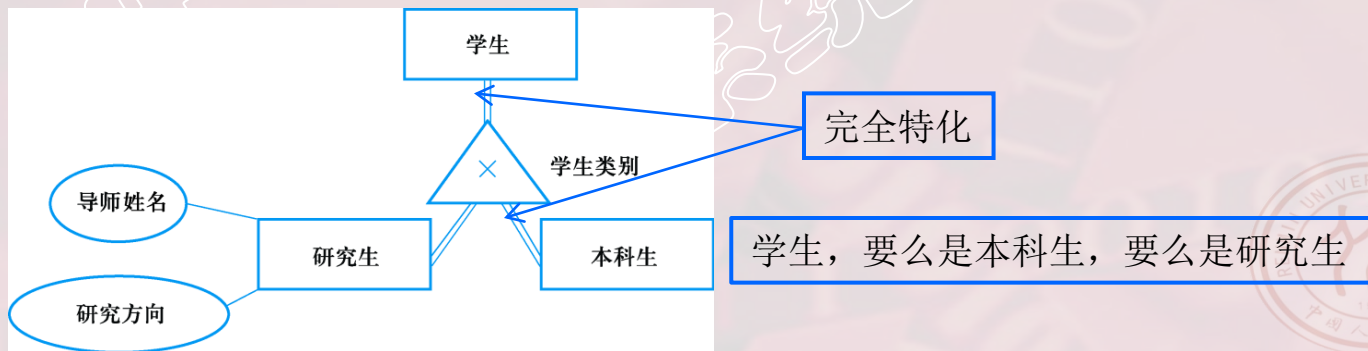
表明一个学生不能既是本科生又是研究生

图7.13 子类的不相交约束

1. ISA 联系

(3) 完备性约束

- 描述父类中的一个实体是否必须是某一个子类中的实体。
 - 如果是，则叫做完全特化 (total specialization)
 - 否则叫做部分特化 (partial specialization)
- 完全特化用父类到子类的双线连接来表示
- 部分特化用父类到子类的单线连接来表示



2. 基数约束

❖ 为什么要对E-R模型进行扩展？

为了增强和丰富基本E-R模型的表达能力。

❖ 进行了哪些扩展？我们介绍主要的3种扩展：

1. ISA联系

2. 基数约束

3. PART-OF联系



2. 基数约束

什么是基数约束

- 说明实体型中的任何一个实体可以在联系中出现的最少次数和最多次数。
- 对实体之间一对一、一对多、多对多联系的细化。
- 约束用一个数对 min..max 表示, $0 \leq \min \leq \max$ 。例如, 0..1, 1..3, 1..*, 其中 * 代表无穷大。

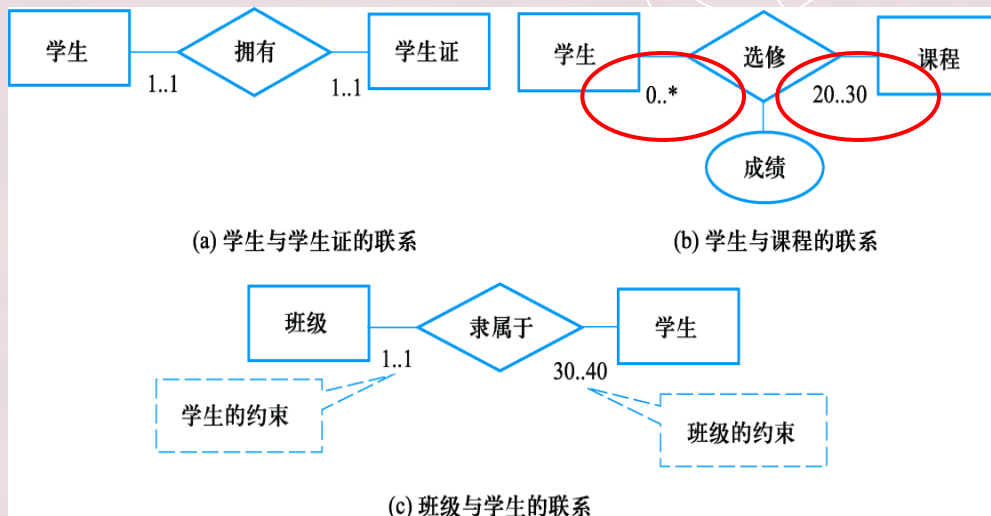


图7.14 一对一、一对多、多对多的基数约束示例

学生实体型的基数约束是20..30
表示每个学生必须选修20~30门课程;

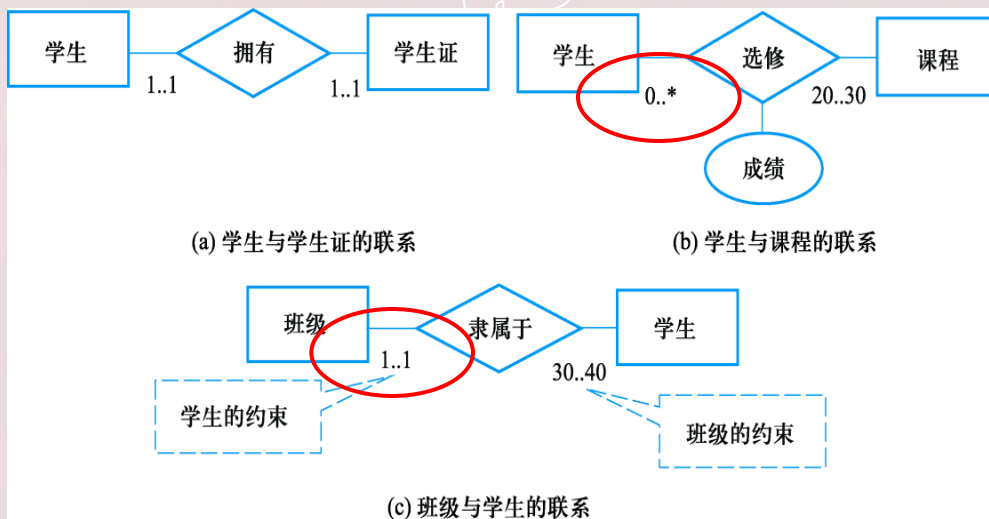
课程的一个基数约束是0..*, 即
一门课程可以被很多同学选修也
可能还没有同学选修, 如新开课。



2. 基数约束

基数约束的类别

- $\min=1$ 的约束叫做**强制参与约束**，即被施加基数约束的实体型中的每个实体都要参与联系；
- $\min=0$ 的约束叫做**非强制参与约束**，被施加基数约束的实体型中的实体可以出现在联系中，也可以不出现在联系中



3. PART-OF联系

❖ 为什么要对E-R模型进行扩展？

为了增强和丰富基本E-R模型的表达能力。

❖ 进行了哪些扩展？我们介绍主要的3种扩展：

1. ISA联系

2. 基数约束

3. PART-OF联系



3. PART-OF联系

❖ 什么是Part-of 联系

描述某个实体型是另外一个实体型的一部分。

❖ Part-of 联系可以分为两种情况：

- 非独占的Part-of联系，简称非独占联系

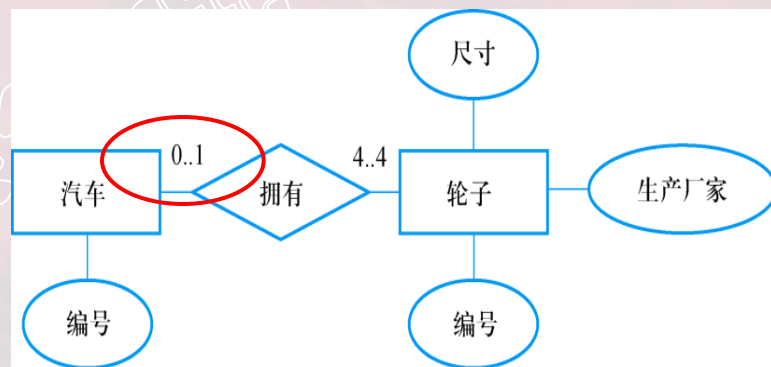
整体实体如果被破坏，另一部分实体仍然可以独立存在

- 独占的Part-of联系，简称独占联系

整体实体如果被破坏，部分实体不能存在

❖ Part-of 联系如何表示？

- 用非强制参与联系表示非独占的Part-of联系
- 用弱实体类型和识别联系来表示独占联系



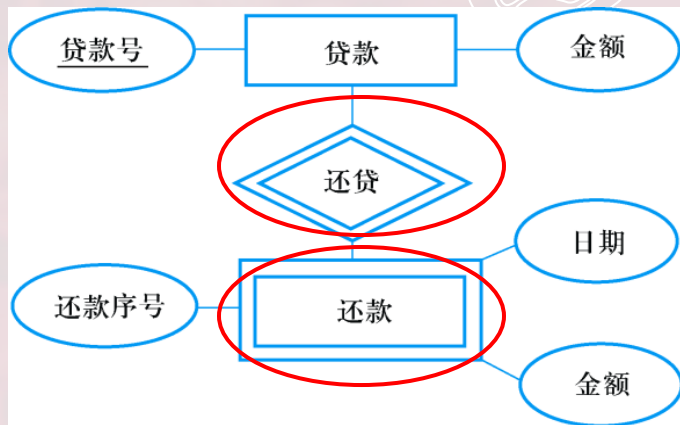
PART-OF联系

● 什么是弱实体型？

➤ 如果一个实体型的存在依赖于其它实体型的存在，则这个实体型叫做弱实体型，否则叫做强实体型。

● 用弱实体类型和识别联系来表示独占联系

双矩形表示弱实体型，用双菱型表示识别联系。



7.3 概念结构设计

7.3.1 概念结构

7.3.2 E-R模型

*7.3.3 扩展的E-R模型

*7.3.4 UML

7.3.5 概念结构设计



7.3.4 UML

- ❖ **UML称为统一建模语言**（Unified Modeling Language），是对象管理组织（Object Management Group, OMG）的一个标准。
- ❖ UML是**为软件开发的各个阶段**提供模型化和可视化支持的规范语言，从需求规格描述到系统完成后的测试和维护。
- ❖ UML可以用于数据建模，业务建模，对象建模，组件建模等。
- ❖ UML提供了多种类型的模型描述图（diagram），借助这些图可以使应用程序更易理解。



用UML的类图建立概念模型

- ❖ UML中的类（class）大致对应E-R图中的实体。
- ❖ 用UML的类图表示E-R图
 - 实体型：用类表示，矩形框上部记上实体名，下面列出属性名。
 - 实体的码：在类图中在属性后面加“PK”（primary key）来表示码属性。
 - 联系：用类图之间的“关联”来表示。

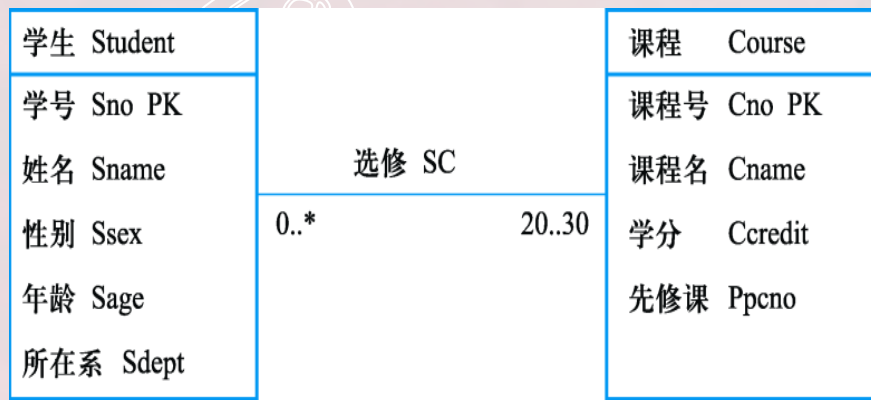


图7.18 用UML的类图表示E-R图 示例



用UML的类图建立概念模型（续）

❖ 基数约束

- UML中关联类之间的基数约束和 E-R图中基数约束的概念类似。

❖ UML中的子类

- 面向对象技术支持超类-子类概念，子类可以继承超类的属性，也可以有自己的属性。
- 这些概念和E-R图的父类-子类联系（ISA联系）是一致的。



7.3 概念结构设计

7.3.1 概念结构

7.3.2 E-R模型

*7.3.3 扩展的E-R模型

1. ISA联系（父类-子类联系）
2. 基数约束（强制参与约束，非强制参与约束）
3. PART-OF联系（非独占联系，独占联系）

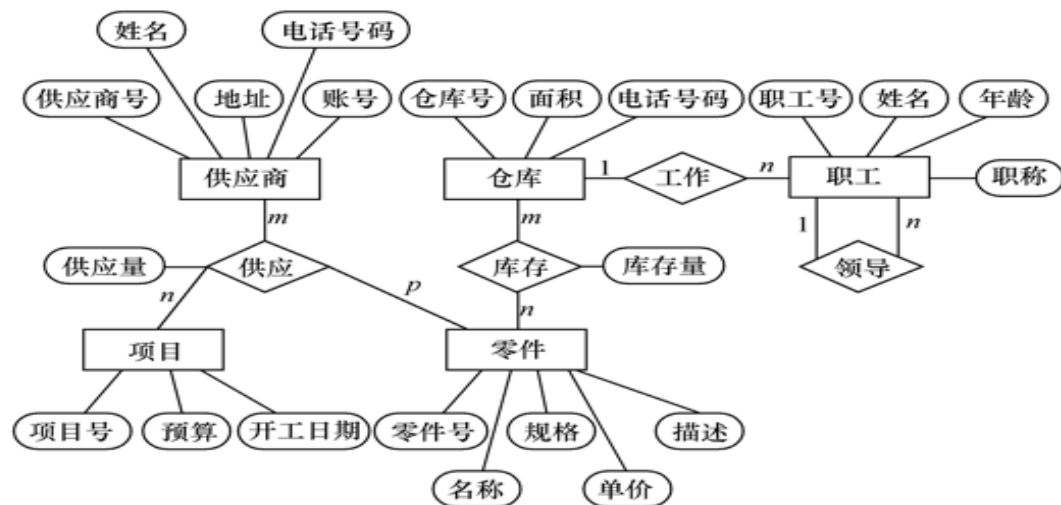
*7.3.4 UML（类图）

7.3.5 概念结构设计



练习题

请把工厂物资管理的E-R图用UML的类图来描述。



(c) 完整的实体-联系图



