

## PowerDesigner16 下画 E-R 图

### 一、概念数据模型概述

数据模型是现实世界中数据特征的抽象。数据模型应该满足三个方面的要求：

- 1) 能够比较真实地模拟现实世界
- 2) 容易为人所理解
- 3) 便于计算机实现

概念数据模型也称信息模型，它以实体 - 联系(Entity-RelationShip,简称 E-R)理论为基础，并对这一理论进行了扩充。它从用户的观点出发对信息进行建模，主要用于数据库的概念级设计。

通常人们先将现实世界抽象为概念世界，然后再将概念世界转为机器世界。换句话说，就是先将现实世界中的客观对象抽象为实体(Entity)和联系(Relationship),它并不依赖于具体的计算机系统或某个 DBMS 系统，这种模型就是我们所说的 CDM;然后再将 CDM 转换为计算机上某个 DBMS 所支持的数据模型，这样的模型就是物理数据模型,即 PDM。

CDM 是一组严格定义的模型元素的集合，这些模型元素精确地描述了系统的静态特性、动态特性以及完整性约束条件等，其中包括了数据结构、数据操作和完整性约束三部分。

- 1) 数据结构表达为实体和属性;
- 2) 数据操作表达为实体中的记录的插入、删除、修改、查询等操作;
- 3) 完整性约束表达为数据的自身完整性约束（如数据类型、检查、规则等）和数据间的参照完整性约束（如联系、继承联系等）；

### 二、实体、属性及标识符的定义

实体 (Entity) , 也称为实例, 对应现实世界中可区别于其他对象的“事件”或“事物”。例如, 学校中的每个学生, 医院中的每个手术。

每个实体都有用来描述实体特征的一组性质, 称之为属性, 一个实体由若干个属性来描述。如学生实体可由学号、姓名、性别、出生年月、所在系别、入学年份等属性组成。

实体集 (Entity Set) 是具体相同类型及相同性质实体的集合。例如学校所有学生的集合可定义为“学生”实体集, “学生”实体集中的每个实体均具有学号、姓名、性别、出生年月、所在系别、入学年份等性质。

实体类型 (Entity Type) 是实体集中每个实体所具有的共同性质的集合, 例如“患者”实体类型为: 患者 {门诊号, 姓名, 性别, 年龄, 身份证号.....}。实体是实体类型的一个实例, 在含义明确的情况下, 实体、实体类型通常互换使用。

实体类型中的每个实体包含唯一标识它的一个或一组属性, 这些属性称为实体类型的标识符 (Identifier) , 如“学号”是学生实体类型的标识符, “姓名”、“出生日期”、“信址”共同组成“公民”实体类型的标识符。

有些实体类型可以有几组属性充当标识符, 选定其中一组属性作为实体类型的主标识符, 其他的作为次标识符。

### 三、实体、属性及标识符的表达

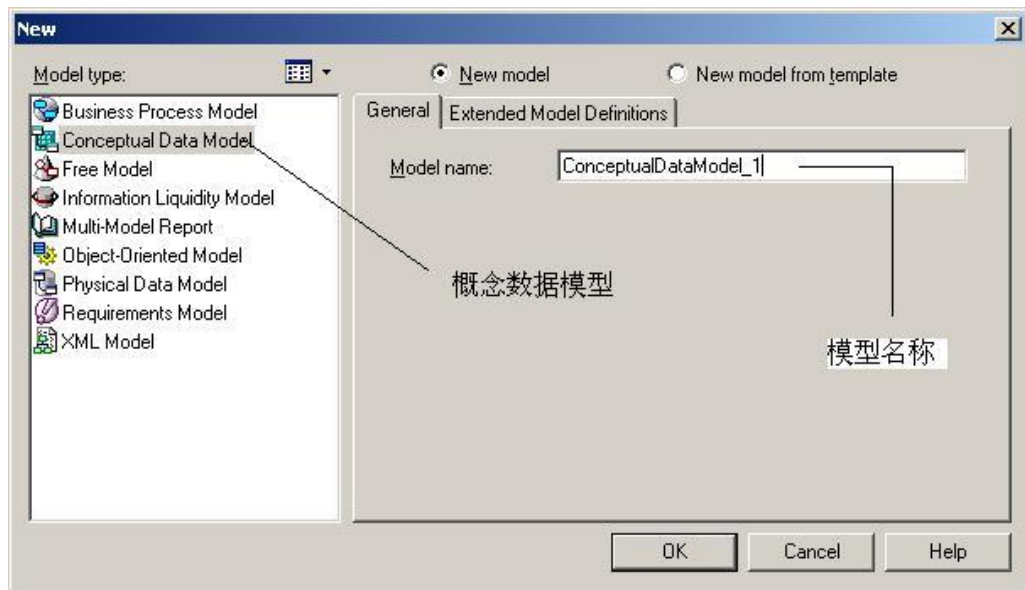
学生				
学号	<pi>	Integer		<M>
姓名		Variable characters (20)		<M>
性别		Characters (1)		
出生日期		Date		<M>
身份证号	<ai>	Variable characters (20)		<M>
主标识符	<pi>			
次标识符	<ai>			

实体的表示方法

介绍 PowerDesigner 概念数据模型以及实体、属性创建。

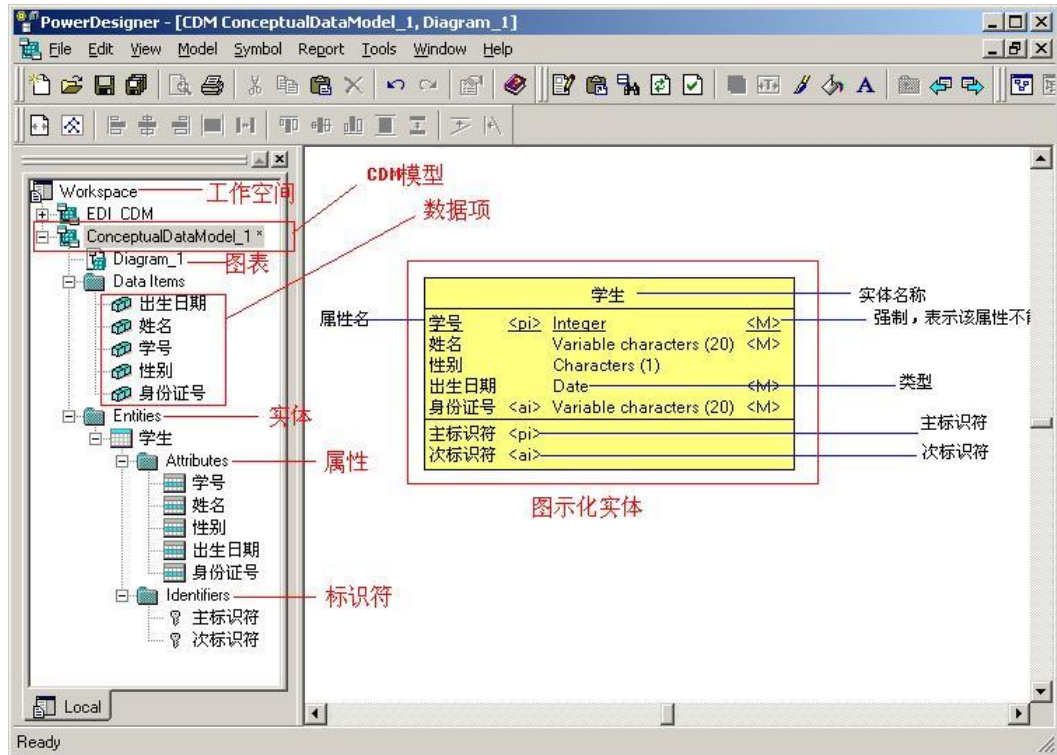
### 一、新建概念数据模型

1) 选择 File-->New,弹出如图所示对话框，选择 CDM 模型（即概念数据模型）建立模型。

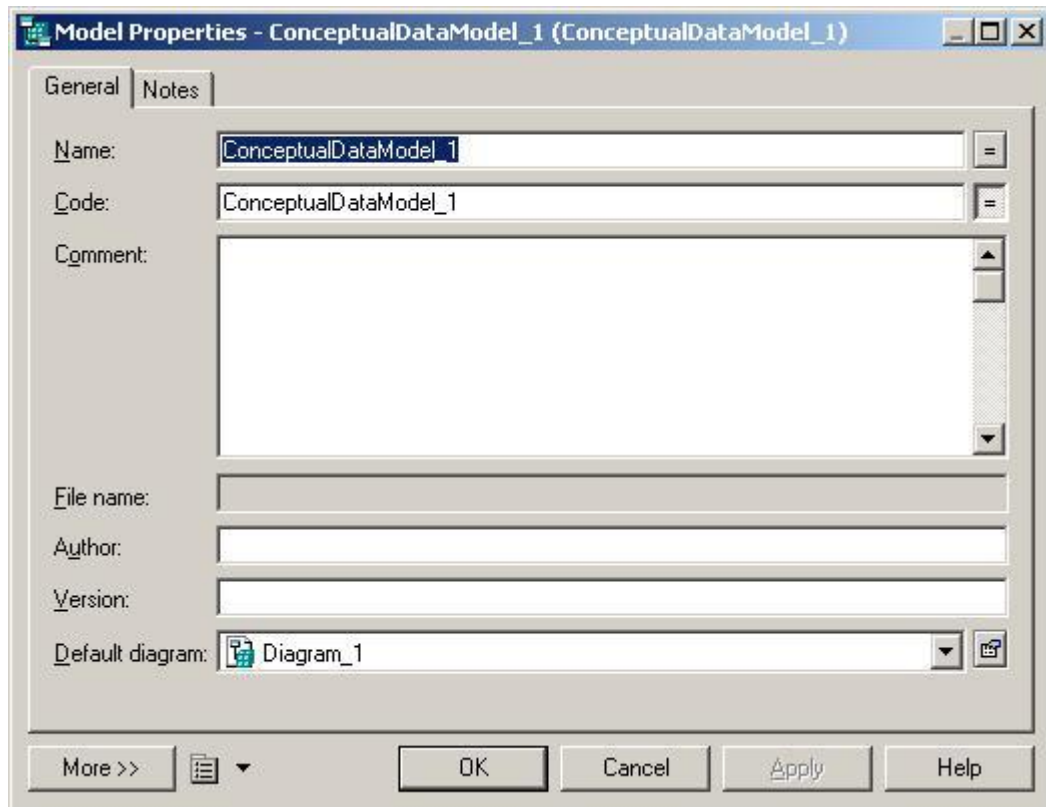


2) 完成概念数据模型的创建。以下图示，对当前的工作空间进行简单介绍。

(以后再更详细说明)

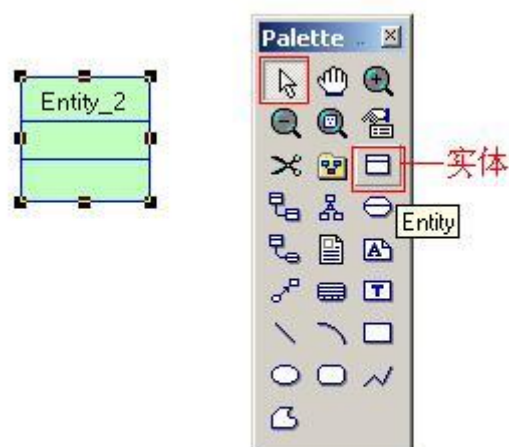


3) 选择新增的 CDM 模型，右击，在弹出的菜单中选择“Properties”属性项，弹出如图所示对话框。在“General”标签里可以输入所建模型的名称、代码、描述、创建者、版本以及默认的图表等等信息。在“Notes”标签里可以输入相关描述及说明信息。当然再有更多的标签，可以点击“More>>”按钮，这里就不再进行详细解释。



## 二、创建新实体

1) 在 CDM 的图形窗口中，单击工具选项版上的 Entity 工具，再单击图形窗口的空白处，在单击的位置就出现一个实体符号。点击 Pointer 工具或右击鼠标，释放 Entity 工具。如图所示



2) 双击刚创建的实体符号，打开下列图标窗口，在此窗口“General”标签中可以输入实体的名称、代码、描述等信息。

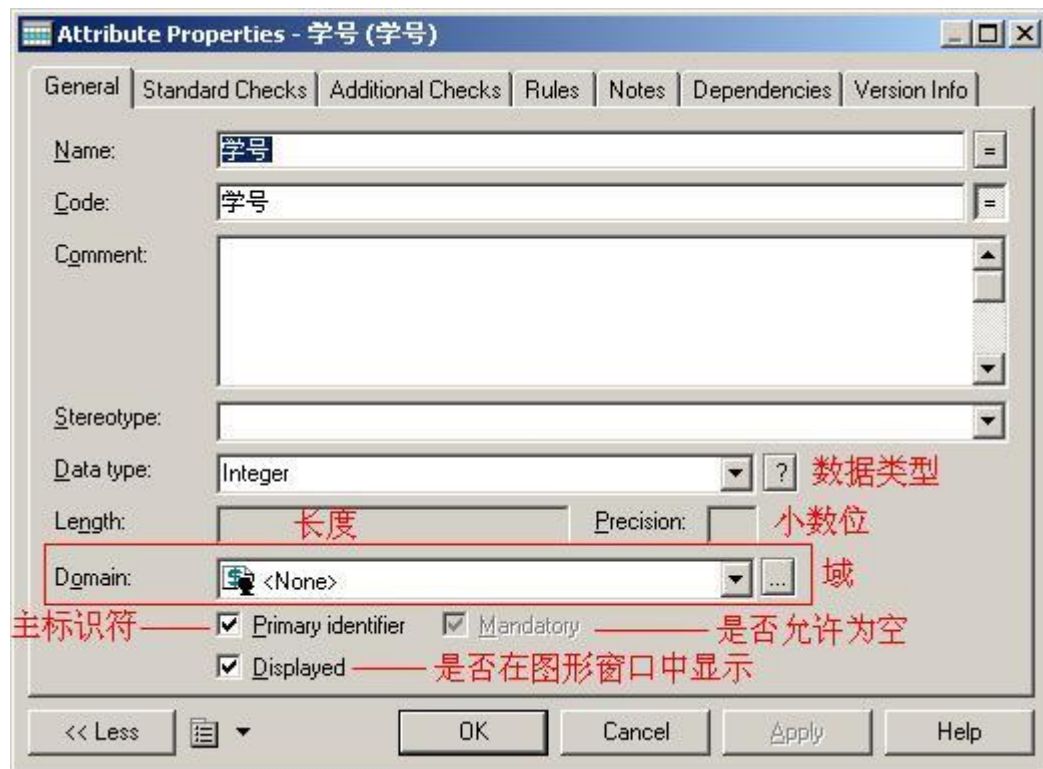


注意：

数据项中的“添加属性”和“重用已有数据项”这两项功能与模型中 Data Item 的 Unique code 和 Allow reuse 选项有关。

P 列表示该属性是否为主标识符;D 列表示该属性是否在图形窗口中显示;M 列表示该属性是否为强制的，即该列是否为空值。

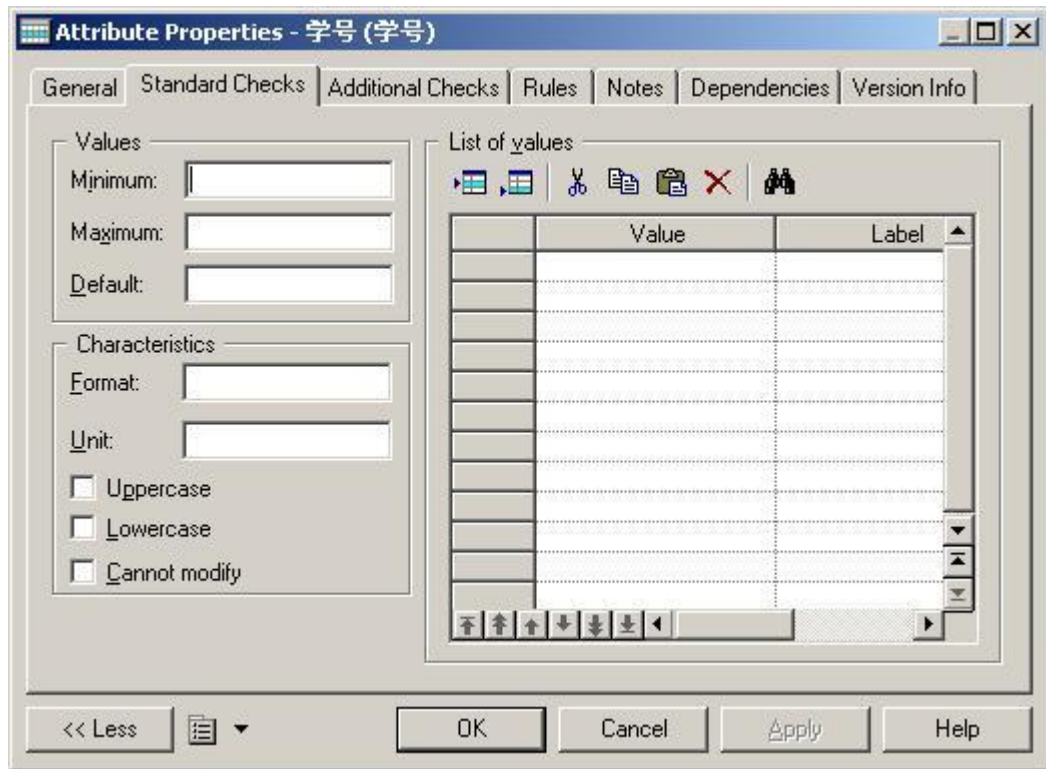
如果一个实体属性为强制的，那么，这个属性在每条记录中都必须被赋值，不能为空。2) 在上图所示窗口中，点击插入属性按钮，弹出属性对话框，如下图所示。



注意：这里涉及到域的概念，即一种标准的数据结构，它可应用至数据项或实体的属性上

#### 一、定义属性的标准检查约束

标准检查约束是一组确保属性有效的表达式。在实体属性的特性窗口，打开如图所示的检查选项卡。



在这个选项卡可以定义属性的标准检查约束，窗口中每项的参数的含义，如下

参数说明 Minimum 属性可接受的最小数 Maximum 属性可接受的最大数

Default 属性不赋值时，系统提供的默认值 Unit 单位，如公里、吨、元 Format 属性

的数据显示格式 Lowercase 属性的赋值全部变为小写字母 Uppercase 属性的赋值全

部变为大写字母 Cannot modify 该属性一旦赋值不能再修改 List Of Values 属性赋值

列表，除列表中的值，不能有其他的值 Label 属性列表值的标签

## 二、定义属性的附加检查

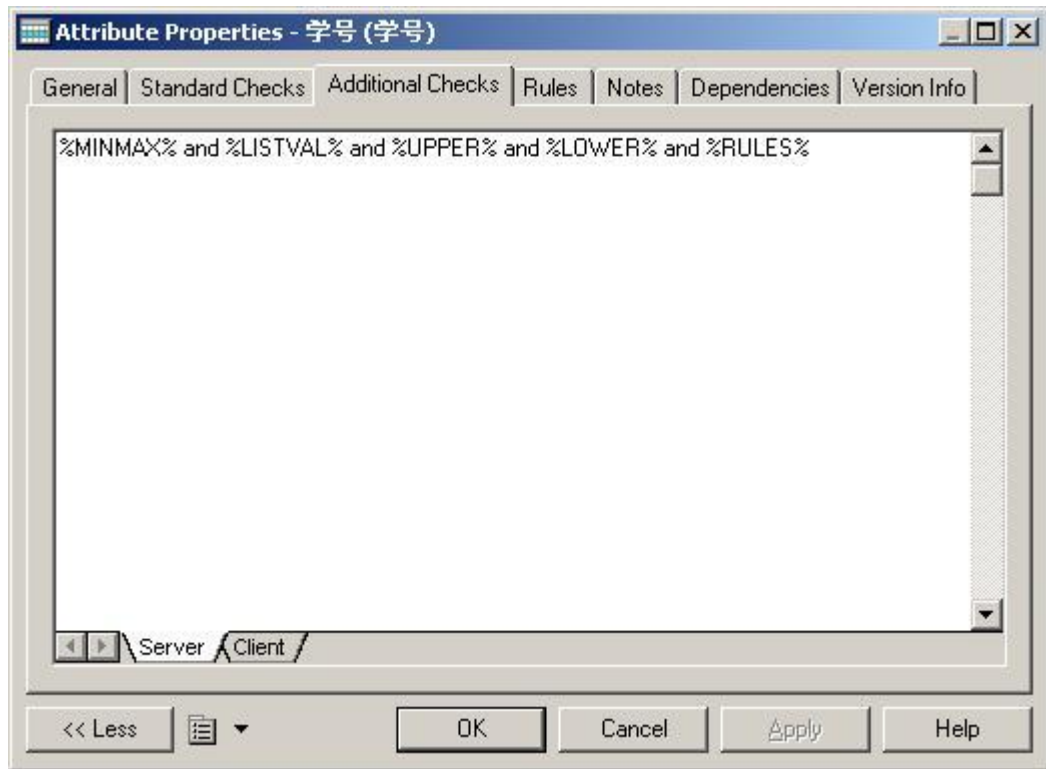
当 Standard checks 或 Rules 不能满足检查的要求时，可以在 Additional

Checks 选项卡的 Server 子页上，通过 SQL 语句中使

用%MINMAX%、%LISTVAL%、%RULES%、%UPPER%、%LOWER%几个变量来

定义 Standard 和 Rule,如图所示





%MINMAX%、%LISTVAL%、%UPPER%、%LOWER%

在 Standard Check 中定义的 Minimum 和 Maximum、List values 、  
uppervalues、lowervalues

%RULES%

在 Rules 特性窗口 Expression 选项卡中定义的有效性规则表达式

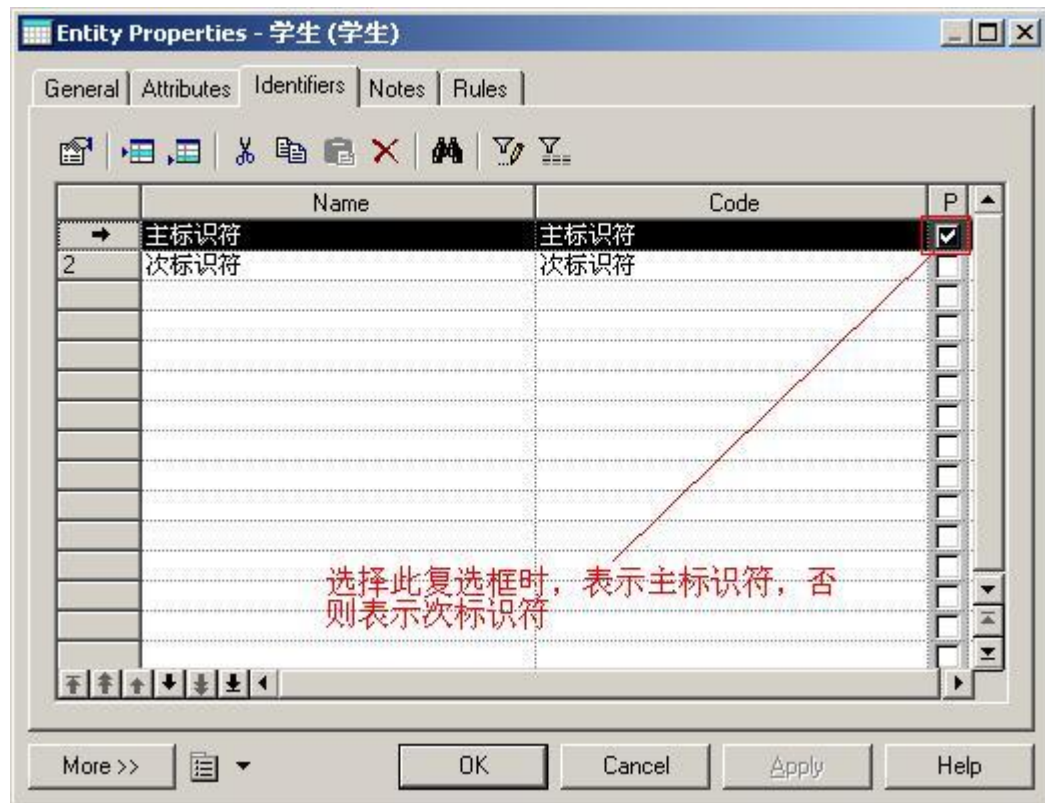
### 一、标识符

标识符是实体中一个或多个属性的集合，可用来唯一标识实体中的一个实例。要强调的是，CDM 中的标识符等价于 PDM 中的主键或候选键。

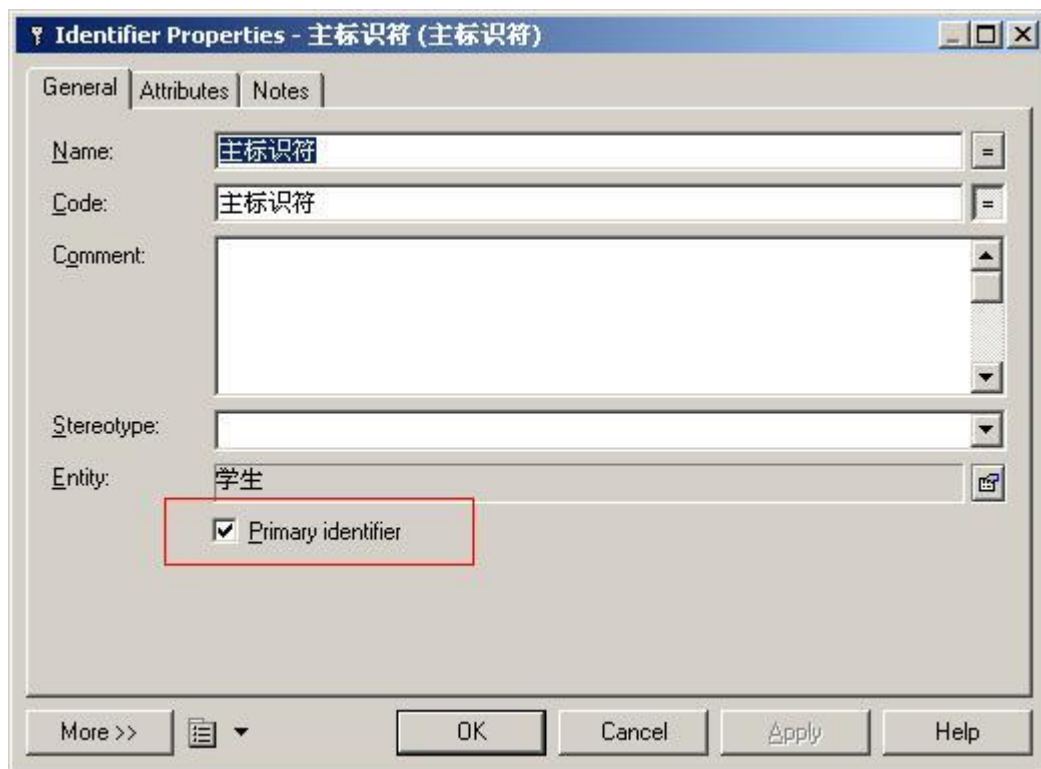
每个实体都必须至少有一个标识符。如果实体只有一个标识符，则它为实体的主标识符。如果实体有多个标识符，则其中一个被指定为主标识符，其余的标识符就是次标识符了。

### 二、如果定义主、次标识符

1) 选择某个实体双击弹出实体的属性对话框。在 Identifiers 选项卡上可以进行实体标识符的定义。如下图所示



2) 选择第一行“主标识符”，点击属性按钮或双击第一行“主标识符”，弹出属性对话框，如图所示



3) 选择"Attributes"选项卡，再点击“Add Attributes”工具，弹出如图所示窗口，选择某个属性作为标识符就行了。



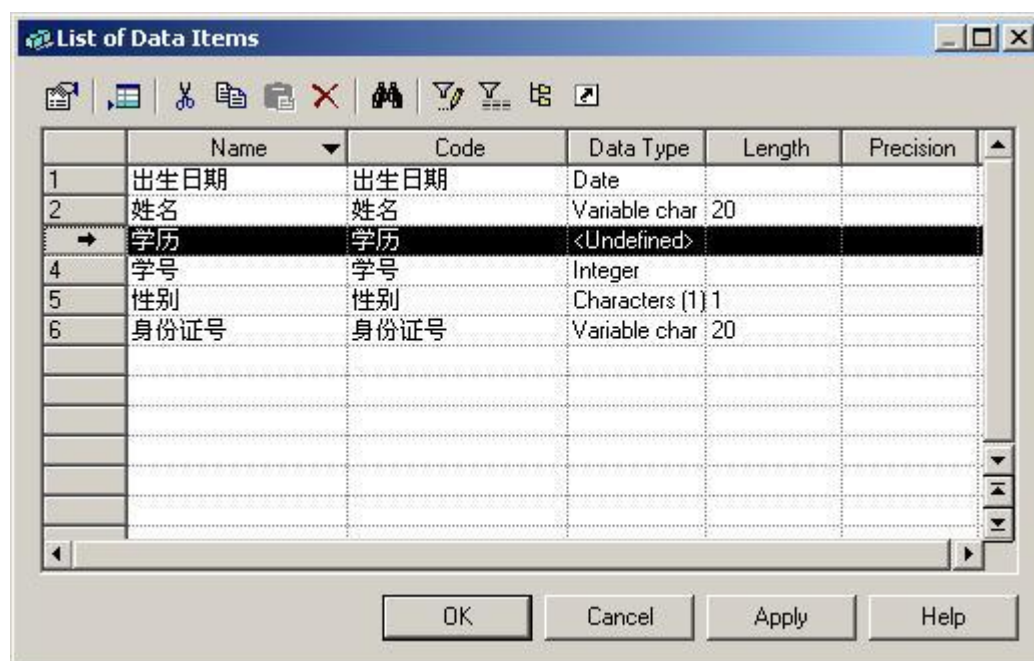
一、数据项

数据项 (Data Item) 是信息存储的最小单位, 它可以附加在实体上作为实体的属性。

注意: 模型中允许存在没有附加至任何实体上的数据项。

## 二、新建数据项

1) 使用 “Model” ---> Data Items 菜单, 在打开的窗口中显示已有的数据项的列表, 点击 “Add a Row” 按钮, 创建一个新数据项, 如图所示



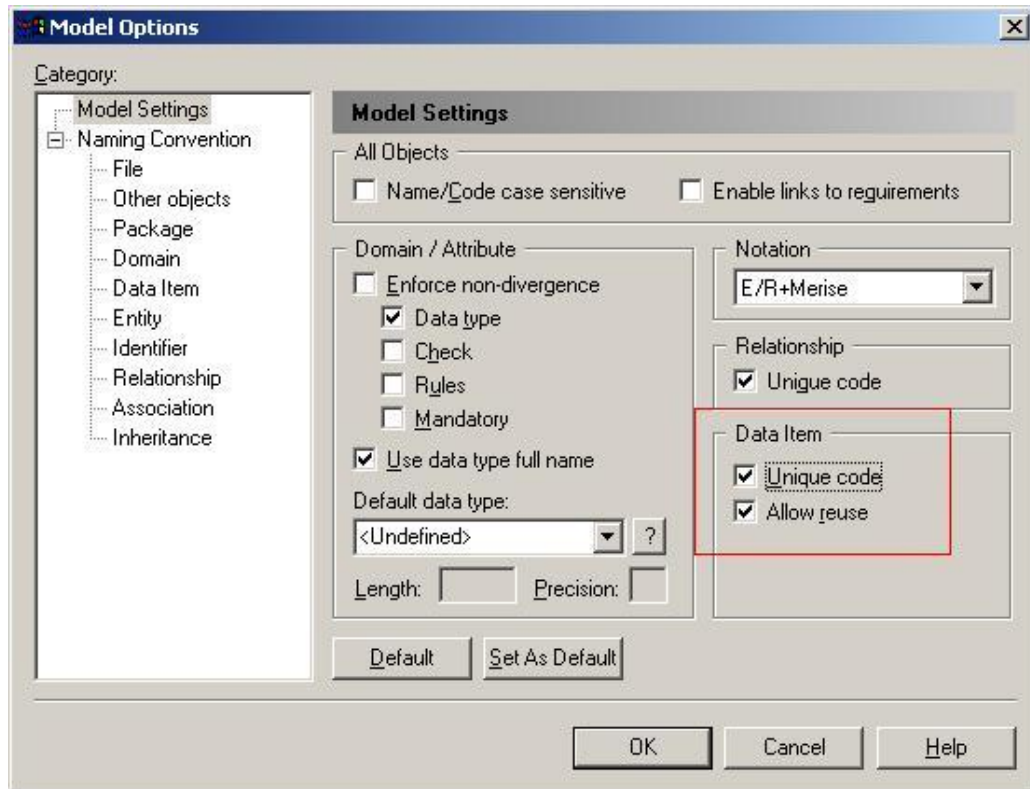
2) 当然您可以继续设置具体数据项的 Code、DataType、Length 等等信息。这里就不再详细说明了。

## 三、数据项的唯一性代码选项和重用选项

使用 Tools--->Model Options->Model Settings。在 Data Item 组框中定义数据项的唯一性代码选项(Unique Code)与重用选项 (Allow Reuse) 。

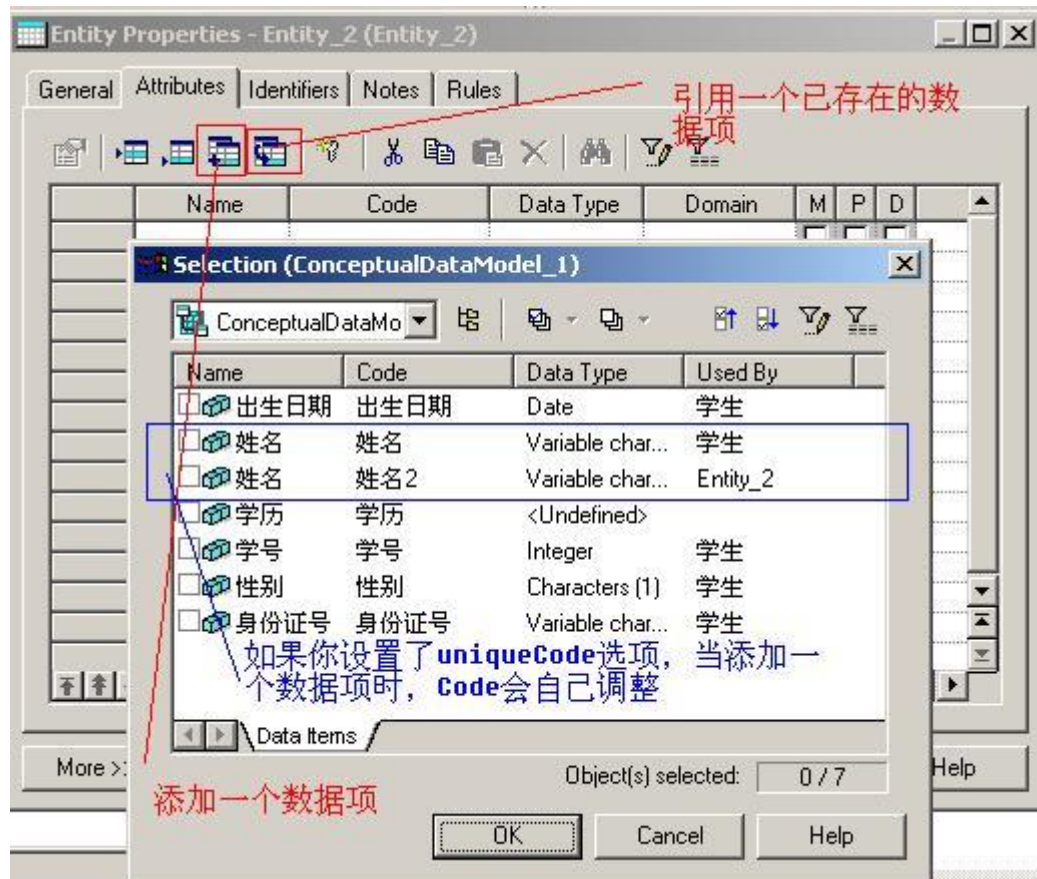
注意:

如果选择 Unique Code 复选框, 每个数据项在同一个命名空间有唯一的代码, 而选择 Allow reuse, 一个数据项可以充当多个实体的属性。



#### 四、在实体中添加数据项

- 1) 双击一个实体符号，打开该实体的属性窗口。
- 2) 单击 Attributes 选项卡，打开如下图所示窗口



注意:

Add a DataItem 与 Reuse a DataItem 的区别在于

Add a DataItem 情况下, 选择一个已经存在的数据项, 系统会自动复制所选择的数据项。如果您设置了 UniqueCode 选项, 那系统在复制过程中, 新数据项的 Code 会自动生成一个唯一的号码, 否则与所选择的数据项完全一致。

Reuse a DataItem 情况下, 只引用不新增, 就是引用那些已经存在的数据项, 作为新实体的数据项

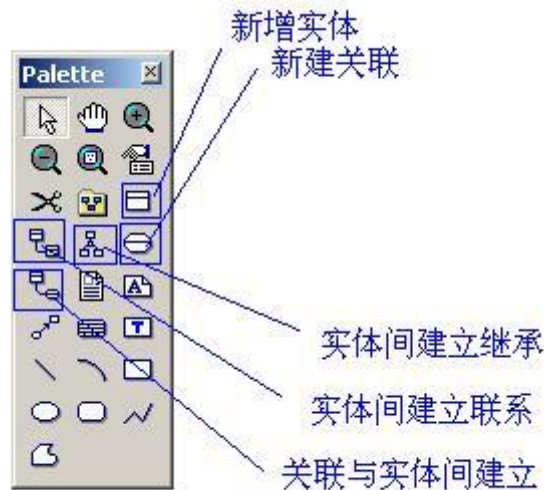
## 一、联系

联系 (Relationship) 是指实体集这间或实体集内部实例之间的连接。

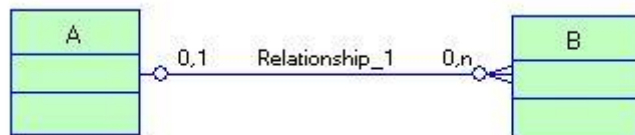
实体之间可以通过联系来相互关联。与实体和实体集对应，联系也可以分为联系和联系集，联系集是实体集之间的联系，联系是实体之间的联系，联系是具有方向性的。联系和联系集在含义明确的情况之下均可称为联系。

按照实体类型中实例之间的数量对应关系，通常可将联系分为 4 类，即一对一 (ONE TO ONE) 联系、一对多 (ONE TO MANY) 联系、多对一 (MANY TO ONE) 联系和多对多联系 (MANY TO MANY)。二、 建立联系

在 CDM 工具选项板中除了公共的工具外，还包括如下图所示的其它对象产生工具。

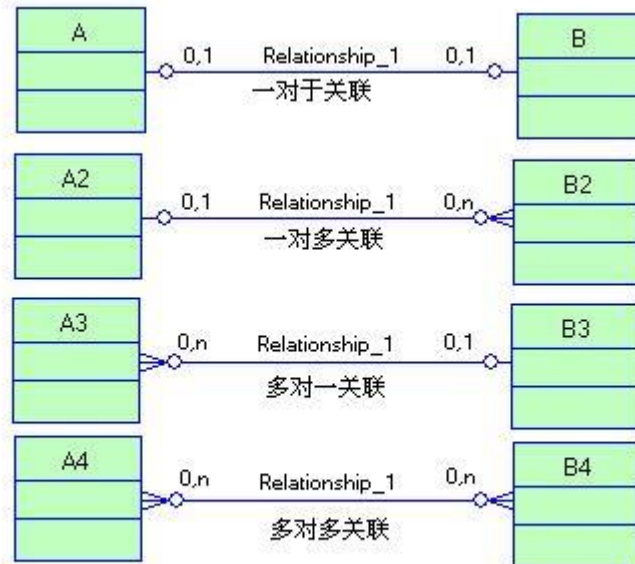


在图形窗口中创建两个实体后，单击“实体间建立联系”工具，单击一个实体，在按下鼠标左键的同时把光标拖至另一个实体上并释放鼠标左键，这样就在两个实体间创建了联系，右键单击图形窗口，释放 Relationship 工具。如下图所示



### 三、 四种基本的联系

即一对一 (ONE TO ONE) 联系、一对多 (ONE TO MANY) 联系、多对一 (MANY TO ONE) 联系和多对多联系 (MANY TO MANY)。如图所示



四种基本的联系

#### 四、 其他几类特殊联系

除了 4 种基本的联系之外，实体集与实体集之间还存在标定联系 (Identify Relationship)、非标定联系 (Non-Identify Relationship) 和递归联系 (Recursive Relationship)。

标定联系：

每个实体类型都有自己的标识符，如果两个实体集之间发生联系，其中一个实体类型的标识符进入另一个实体类型并与该实体类型中的标识符共同组成其标识符时，这种联系则称为标定联系，也叫依赖联系。反之称为非标定联系，也叫非依赖联系。

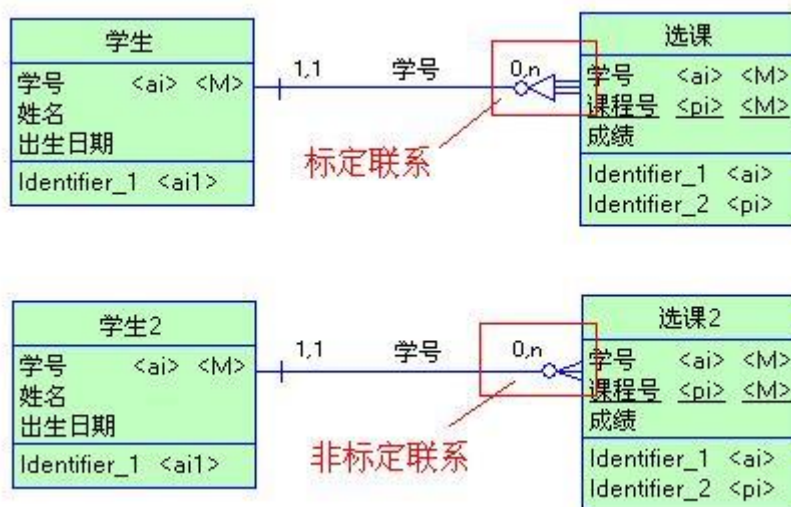
注意：

在非标定联系中，一个实体集中的部分实例依赖于另一个实例集中的实例，在这种依赖联系中，每个实体必须至少有一个标识符。而在标定联系中，一个实体集中的全部实例完全依赖于另一个实例集中的实例，在这种依赖联系中一个实体必须至少有一



个标识符，而另一个实体却可以没有自己的标识符。没有标识符的实体用它所依赖的实体的标识符作为自己的标识符。

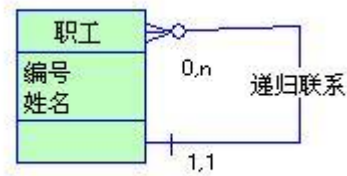
换句话说理解，在标定联系中，一个实体（选课）依赖 一个实体（学生），那么（学生）实体必须至少有一个标识符，而（选课）实体可以没有自己的标识符，没有标识符的实体可以用实体（学生）的标识符作为自己的标识符。



递归联系：

递归联系是实体集内部实例之间的一种联系，通常形象地称为自反联系。同一实体类型中不同实例之间的联系也称为递归联系。

例如：在“职工”实体集中存在很多的职工，这些职工之间必须存在一种领导与被领导的关系。又如“学生”实体集中的实体包含“班长”子实体集与“普通学生”子实体集，这两个子实体集之间的联系就是一种递归联系。创建递归联系时，只需要单击“实体间建立联系”工具从实体的一部分拖至该实体的别一个部分即可。如图



## 五、 定义联系的特性

在两个实体间建立了联系后，双击联系线，打开联系特性窗口，如图所示。



## 六、 定义联系的角色名

在联系的两个方向上各自包含有一个分组框，其中的参数只对这个方向起作用，Role Name 为角色名，描述该方向联系的作用，一般用一个动词或动宾组表。

如：“学生 to 课目 ” 组框中应该填写“拥有”，而在“课目 To 学生”组框中填写“属于”。（在此只是举例说明，可能有些用词不太合理）。

## 七、 定义联系的强制性

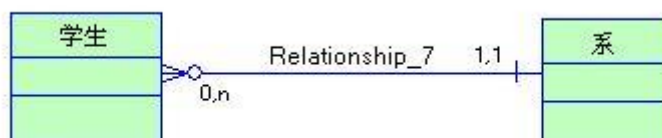
Mandatory 表示这个方向联系的强制关系。选中这个复选框，则在联系线上产生一个联系线垂直的竖线。不选择这个复选框则表示联系这个方向上是可选的，在联系线上产生一个小圆圈。

#### 八、有关联系的基数

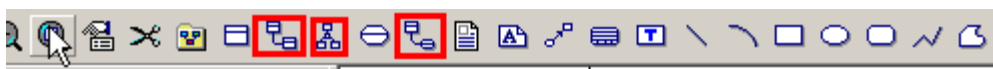
联系具有方向性，每个方向上都有一个基数。

举例，

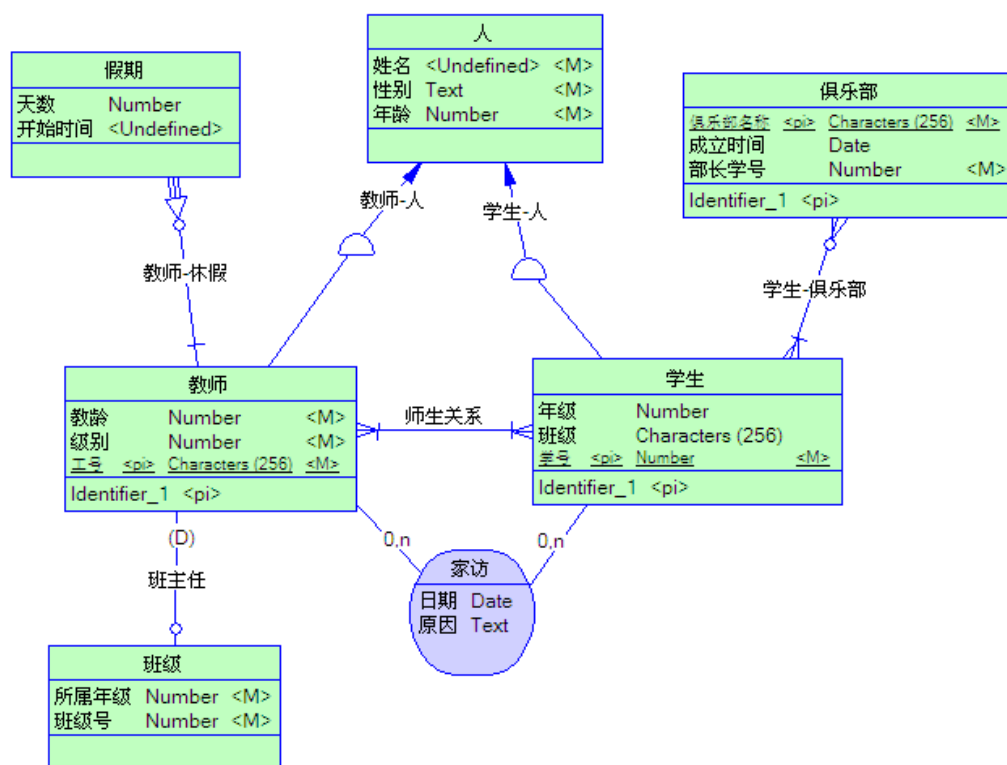
“系”与“学生”两个实体之间的联系是一对多联系，换句话说“学生”和“系”之间的联系是多对一联系。而且一个学生必须属于一个系，并且只能属于一个系，不能属于零个系，所以从“学生”实体至“系”实体的基数为“1,1”，从联系的另一方向考虑，一个系可以拥有多个学生，也可以没有任何学生，即零个学生，所以该方向联系的基数就为“0,n”，如图所示



CDM 是大多数开发者使用 PD 时最先创建的模型，也是整个数据库设计最高层的抽象。CDM 是建立在传统的 ER 图模型理论之上的，ER 图中有三大主要元素：实体型，属性和联系。其中实体型对应到 CDM 中的 Entity，属性对应到 CDM 中每个 Entity 的 Attribute，在概念上基本上是一一对应的。但在联系上，CDM 有了比较大的扩展，除了保留 ER 图原有的 Relationship 概念之外，还增加了 Association，Inheritance 两种实体关系，下面就让我们分别看看这些关系的用法和之间的区别（下图中被标红的工具栏按钮就是用来向实体中添加这些关系的）。



另外，在介绍所有这些 CDM 中的元素之前，笔者先给出一个很简单的 CDM 图，是对我们最最熟悉的学校场景的一个建模，下文中提到的所有概念在图中都有体现，大家在看下文的时候可以对照着来看：

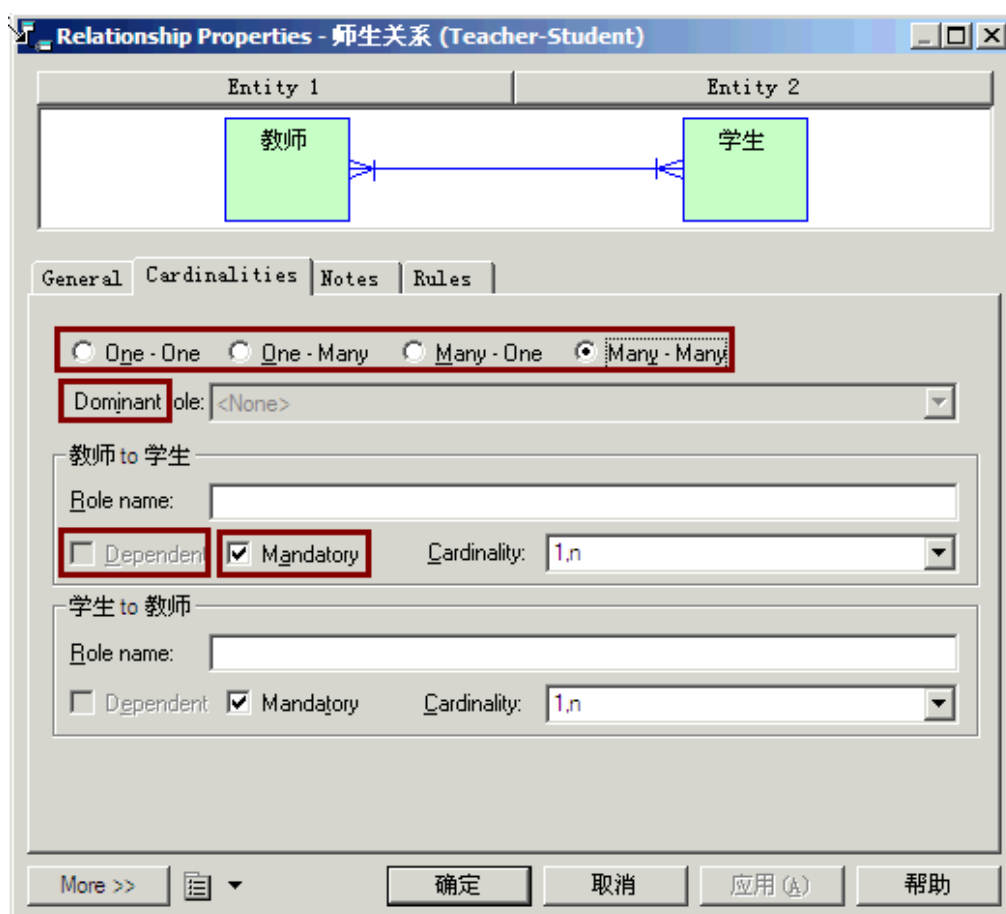


## 一. Relationship(联系)

先给出 PD 手册里对联系的定义：“A relationship is a link between entities.

For example, in a CDM that manages human resources, the relationship Member links the entities Employee and Team, because employees can be members of teams. This relationship expresses that each employee works in a team and that each team has employees.” 可见，也许联系的概念真的太简单了吧，所以反而不那么好表述，所以 PD 的文档里也是用一个例子来说明出现了什么样的情况我们就认为两个实体间是有联系的。

当我们提起实体间联系的时候，最先想到的恐怕是 one to one, one to many 和 many to many 这三种联系类型，这些联系类型也是大家最熟悉的。笔者对 ER 图原本的概念并不精通，但在 CDM 中，联系还有另外三个可以设置的属性：mandatory（强制性联系），dependent（依赖性联系/标定关联）和 dominant（统制联系）。这些属性对后面 PDM 的生成都有比较大的影响，需要我们——有所了解。它们都是在联系的属性控制面板中设定的，见下图：



### 1.mandatory

联系是否具有强制性，指的是实体间是不是一定会出现这种联系；或者换句话说，当我们在谈及一个联系的应用场景的时候，联系对应的那两个实体型的实体实例的个数可不可能为零。也许这样的解释还是有点抽象，让我们举两个联系的例子，一个是对两边的实体都有强制性的，另一个则不然。

### (1) 教师--学生 联系

这个联系首先是一个多对多联系，因为每个老师可以教多个学生，每个学生也都有多个老师来负责他们的学业。同时，这个联系对教师和学生都是强制性的，也就是说，不存在任何一个老师，他不负责任何一个学生的教学；也不存在任何一个学生，他没有任何一个任课老师。

### (2) 学生--俱乐部 联系

这个联系也是一个多对多关系，但它对学生这个实体型而言就不是强制的 (Optional,可选的)。每个俱乐部都有至少一个学生参加，但并不是每个学生都要去参加俱乐部的活动。完全可以有一些学生，他们什么俱乐部都没参加。

上面的例子主要是从概念的角度来区分了 mandatory 和 optional 的区别。实际上如果把模型对应到我们最后生成的表，如果 A-B 间的联系对 A 是 mandatory 的话，那么如果在 A 里面包含 B 的外键，这个外键不能为空值，反之可以为空值。后面我们谈到 PDM 和实际数据库的时候，大家会看到这一点。

## 2.dependent

每一个 Entity 型都有自己的 Identifier，如果两个 Entity 型之间发生关联时，其中一个 Entity 型的 Identifier 进入另一个 Entity 型并与该 Entity 型中的 Identifier 共同组成其 Identifier 时，这种关联称为标定关联,也叫依赖性关联(dependent relationship)。一个 Entity 型的 Identifier 进入另一个 Entity 型后充当其非 Identifier 时，这种关联称为非标定关联,也叫非依赖关联。

概念的定义说起来还是有些拗口，说白了其实就是主-从表关系，从表要依赖于主表。比如在我们系统里要记录教师休假的情况，有一个实体型 Holiday，其属性包括休假的开始时间和天数，每次有教师休假的时候，都要在这个表留下记录。从我们

的场景描述中可以看到，实体型假期必须依附于实体型教师，即对于每一个假期实例，必须指向某一个教师实例。

对于依赖型联系，必须注意它不可能是一个多对多联系，在这个联系中，必须有一个作为主体的实体型。一个 dependent 联系的从实体可以没有自己的 identifier.

### 3.dominant

这个联系属性是最为简单的，它仅作用于一对一联系，并指明这种联系中的主从表关系。在 A,B 两个实体型的联系中，如果 A-->B 被指定为 dominant，那么 A 为这个一对一联系的主表，B 为从表，并且在以后生成的 PDM 中会产生一个引用（如果不指定 dominant 属性的话会产生两个引用）。比如老师和班级之间的联系，因为每个班级都有一个老师做班主任，每个老师也最多只能做一个班级的班主任，所以是一个一对一关系。同时，我们可以将老师作为主表，用老师的工号来唯一确定一个班主任联系。

## 二.Association (关联)

先来看一下 PD 给 association 的定义：“An association is a connection between entities. In the Merise modeling methodology an association is used to connect several entities that each represents clearly defined objects, but are linked by an event, which may not be so clearly represented by another entity.”。

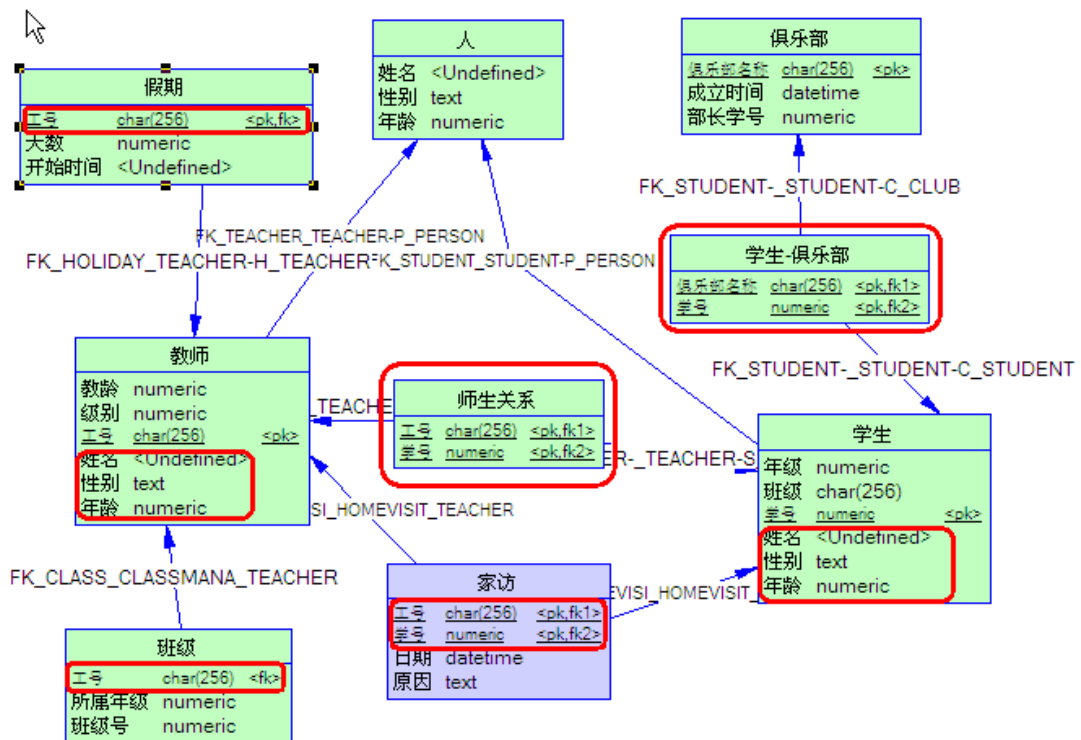
在上一小段提到的那些 Relationship，在很多情况下（特别是多对多关系中），我们会把联系专门提出来，作为一个实体型放在两个需要被关联的实体型中间（在 PD 中，选中任何一个联系，在右键的弹出菜单中选择“Change to Entity”命令即可完成联系转实体的操作）。但有的时候，把若干个实体型之间的联系抽象为一

个实体型可能不太合适，这个时候你可以选择为这些实体型建立一个 association，那么在生成 PDM 的时候，所有这些相关实体型的 identifier 都会被加入到 association 对应生成的表模型中。所以，说白了，其实 association 就是实体型的一种特例，用来在建模的时候更确切的表达实体间的关联信息。在 PD 的文档中举了一个录音带、顾客、商店三个实体型在租借录音带这个场景上发生关联，然后把租借定义为上述三个实体型之间的 association 的例子，非常确切。在我们的学校模型里，我定义了家访做为老师和学生实体型中间的一个 association，在接下来产生的 PDM 中大家就能看到这种定义所产生的效果。

### 三.Inheritance (继承)

这种关系在概念层面是最容易理解的了，本文就不赘述了。

前面已经介绍了 CDM 中关于实体间关系的主要内容，接下来我们就来看看根据这个 CDM 所生成的 PDM 是一个什么样子：

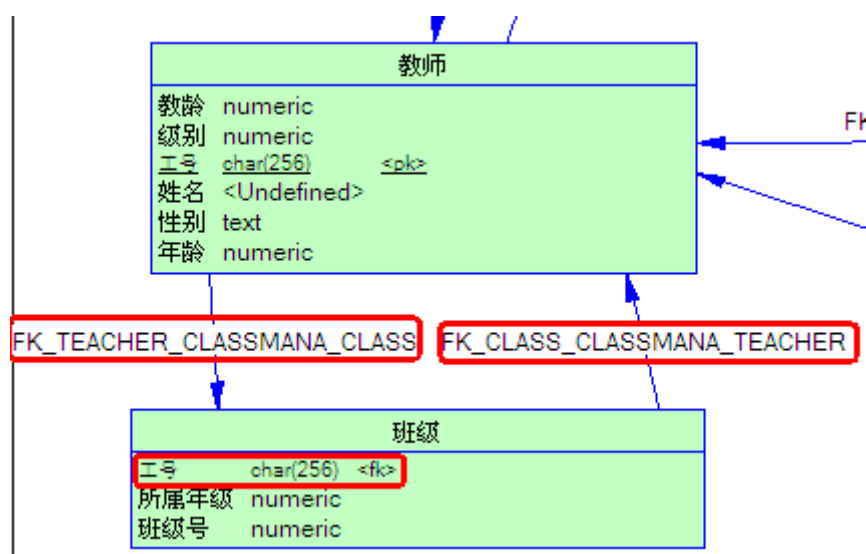




上图中所有标红的部分是我们最应该关注的内容，因为他们都是由于我们对实体型间的关系的定义而产生的，下面给出一些简单的说明。

1. “师生关系”和“学生俱乐部”这两个表是由于我们的多对多关系而产生的。
2. “假期”表的“工号”字段是由于我们将教师-假期关系指定为 dependent 而产生的。
3. “班级”表的“工号”字段是由于我们将教师-班级关系制定为 dominant 而产生的。
4. “家访”表中的“工号”和“学号”字段是由于家访是教师和学生实体型的 association 而产生的。

另外，记得我们在提到 dominant 属性的时候说过，一个没指定 dominant 方向的一对一联系将产生两个引用，下面我们就把原本的 CDM 中的教师-班级关系进行一个小小的修改，去掉这个 relationship 的 dominant 定义，那么最终产生的 PDM 中教师表和班级表将互相包含对方的主键(由于我们的班级表没有自己的主键，所以只能在班级表中看到多出来的列)，截图如下：



对照这个 PDM 截图和上一个 PDM 截图之间的区别，大家可以很容易得看出 dominant 属性对一个一对一关系的作用。