

6.7 使用 PowerDesigner 进行数据库设计

6.7.1 Powerdesigner 介绍

1. 概述

PowerDesigner 是 Sybase 公司的 CASE 工具集,使用它可以方便地对管理信息系统进行分析设计,它几乎包括了数据库模型设计的全过程,是一款开发人员常用的数据库建模工具。

PowerDesigner 主要包括以下功能:

- (1) 集成多种建模能力,能建立的模型包括:数据模型 (E/R, Merise)、业务模型 (BPMN, BPEL, ebXML)、应用模型 (UML) 等;
- (2) 自动生产代码,包括:SQL (支持多于 50 种数据库系统), Java, .NET 等;
- (3) 强大的逆向工程能力;
- (4) 可扩展的企业库解决方案,具备强大的安全性及版本控制能力,可支持多用户自动化、可定制的报表能力。

2. PowerDesigner 环境

PowerDesigner 的环境如图 6-11 所示,包含以下几部分:

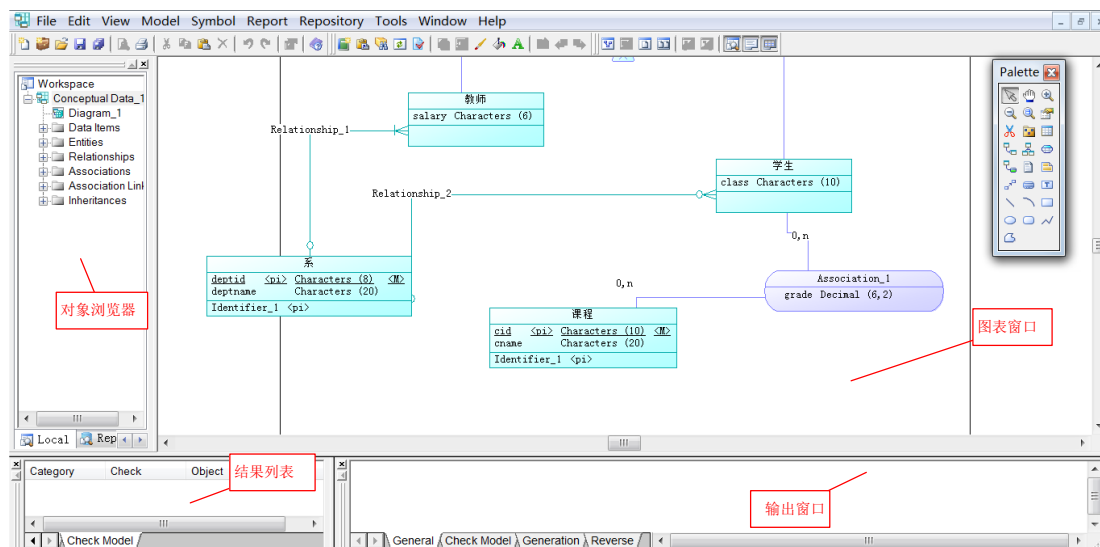


图 6-11 PowerDesigner 环境

◆ 对象浏览器

对象浏览器可以用分层结构显示工作空间,显示模型以及模型中的对象,实现快速导航。通过对象浏览器还可以访问 PowerDesigner 库,可以把模型及相关的文件存放到 PowerDesigner 库中。对象浏览器中主要包含以下对象:

- ◇ 工作空间 (Workspace): 它是对象浏览器中树的根,是组织及管理所有设计元素的虚拟环境。用户可以通过保存工作空间保存自己的设计空间信息以便再次打开时可以还原到保存前的状态。
- ◇ 项目 (Project): 项目中的所有对象可以作为一个单元存到 PowerDesigner 库中。每个项目自动维护一张图用以显示模型以及文档之间的依赖。
- ◇ 文件夹 (folder): 工作空间可以包含用户自定义的文件夹用以组织模型和文件。例

如，有两个独立的项目，希望在一个工作空间中处理，此时可以使用文件夹。

✧ 模型 (Model): 模型是 PowerDesigner 中的基本设计单元。每个模型中有一个或多个图以及若干模型对象。

✧ 包 (Package): 当模型较大时，可能需要把模型拆分成多个“子模型”以便于操作，这些子模型就叫做包，今后可以把不同的包分配给不同的开发组。

✧ 图 (Diagram): 展现模型对象之间的交互，可以在模型或包中创建多个图。

◆ 输出窗口: 显示操作的进程，比如模型检查或从数据库逆向工程。

◆ 结果列: 用于显示生成、覆盖和模型检查结果，以及设计环境的总体信息。

◆ 图表窗口: 用于组织模型中的图表，以图形方式显示模型中各对象之间的关系。

3. 常用的 Powerdesigner 的几种模型:

◆ 概念数据模型 (Conceptual Database Model, CDM)

CDM 帮助分析信息系统的概念结构，识别主要实体、实体的属性及实体之间的联系。CDM 比逻辑数据模型 (LDM) 和物理数据模型 (PDM) 抽象。CDM 表现数据库的全部逻辑的结构，与任何的软件或数据存储结构无关。因此，CDM 是适合于系统分析阶段的工具。由 CDM 可生成 LDM、PDM 和 OOM。

◆ 逻辑数据模型 (Logical Data Model, LDM)

LDM 帮助分析信息系统的结构，它独立于具体物理数据库的实现。LDM 比 CDM 具体，但不允许定义视图、索引以及其它在物理数据模型 (PDM) 中处理的细节。可以把逻辑数据模型作为数据库设计的中间步骤，它在概念数据模型与物理数据模型之间。

◆ 物理数据模型 (Physical Data Model, PDM)

PDM 帮助分析数据库中的表、视图及其它对象，还包括数据仓库所需的多维对象，可针对目前主流数据库进行建模、逆向工程以及产生代码。PDM 叙述数据库的物理实现，主要目的是把 CDM 中建立的现实世界模型生成特定的 DBMS 脚本，产生数据库中保存信息的储存结构，保证数据在数据库中的完整性和一致性。PDM 是适合于系统设计阶段的工具。

◆ 面向对象模型 (Object-oriented Data Model, OOM)

一个 OOM 包含一系列包、类、接口和它们的关系。一个 OOM 本质上是软件系统的一个静态的概念模型。使用 PowerDesigner 面向对象模型建立 OOM，能为纯粹地面向对象的系统建立一个 OOM，产生 Java 文件或者 PowerBuilder 文件，或使用一个来自 OOM 的物理数据模型 (PDM) 对象来表示关系数据库设计分析。

本章中主要介绍和使用 PowerDesigner 的概念数据模型和物理数据模型来设计数据库。

6.7.2 概念数据模型

1. 概述

概念数据模型也称信息模型，它以实体联系理论为基础，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和联系，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）。

CDM 的功能如下：

- (1) 通过创建实体关系（E-R）图来描述数据的组织结构。
- (2) 能够校验数据设计的合理性。
- (3) 能够生成指定了相应物理实现数据库的物理数据模型（PDM）。
- (4) 能够生成用 UML 标准描述 CDM 中对象的面向对象模型（OOM）。
- (5) 为在不同的设计阶段创建另一个模型版本，可以生成概念数据模型（CDM）。

图 6- 中描述了 CDM、PDM 和 OOM 之间的转换关系。

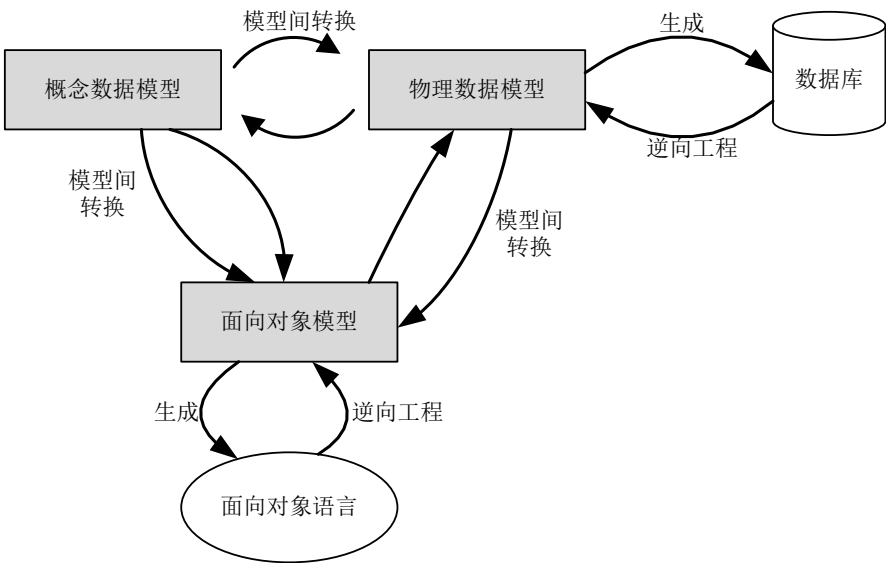


图 6- 1 CDM、PDM 和 OOM 之间的关系

2. 创建 CDM

步骤如下：

- (1) 选择【File】|【New Model】，弹出如图 6- 所示对话框。在【Model types】选项页中选择【Conceptual Data Model】，在【Model name】中输入 CDM 模型的名称（本例中为模型取名 CDM_grade）。

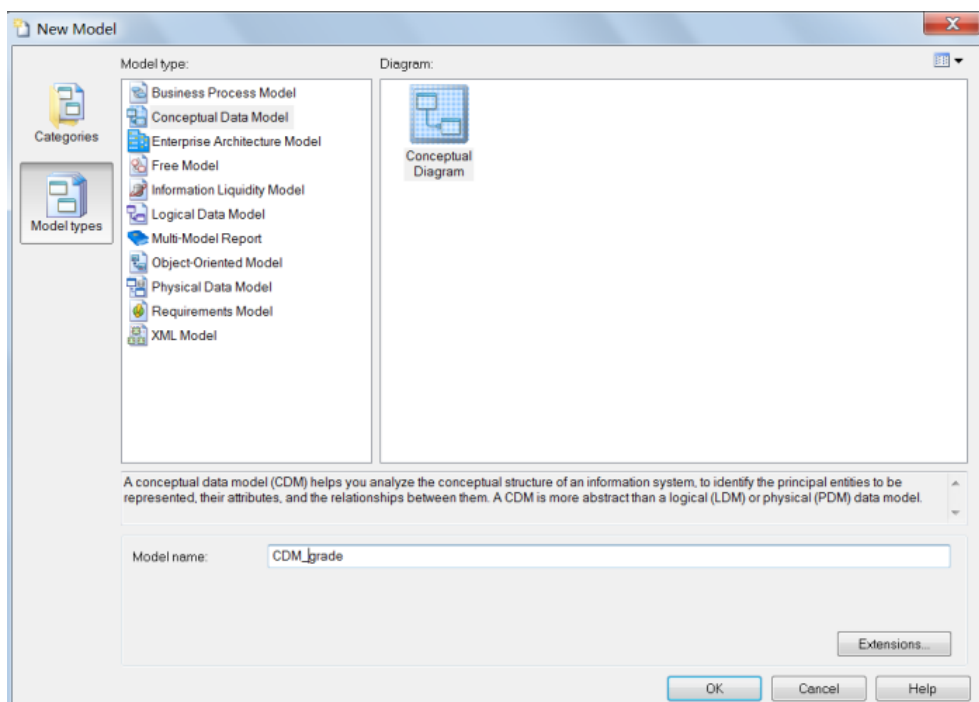


图 6-2 新建 CDM

(2) 在【对象浏览器】中选择新建的 CDM，右键单击，在快捷菜单中选择【Properties】属性项，弹出如图 6- 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信息。在【Notes】标签里可以输入相关描述及说明信息。

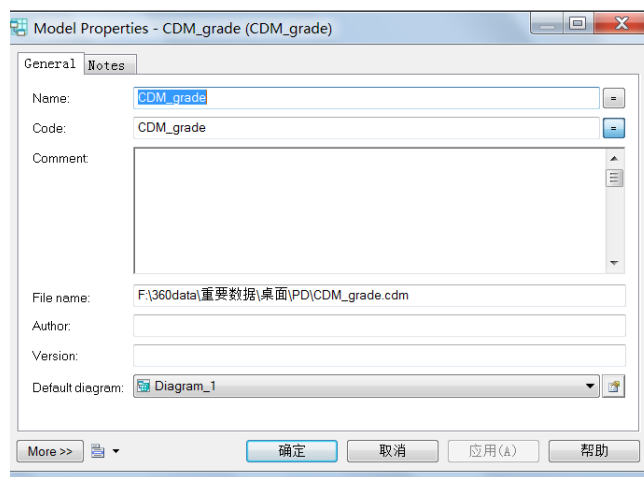


图 6-3 CDM 属性窗口

下面我们具体介绍如何创建 CDM。

图形	名称	操作	图形	名称	操作
	指针	选择符号		联合连接	插入联合连接符号
	套索	一个区域的选择符号		文件	插入一个文件符号
	整体选择	选择全部符号，一起设置大小		注释	插入注释符号
	放大	放大视野范围		连接/扩展依赖	在图表中的符号之间画一个图形连接，在注释和一个对象之间画一个注释连接，在两个支持扩展依赖的对象间画一个扩展依赖
	缩小	缩小视野范围		主题	插入主题符号
	打开包图表	显示选择包的图表		文本	插入文本
	属性	显示选择的符号属性		线条	插入一条线
	删除	删除符号		圆弧	插入一个圆弧
	包	插入包符号		长方形	插入一个长方形
	实体	插入实体符号		椭圆	插入一个椭圆
	关系	插入关系符号		圆角矩形	插入一个圆角矩形
	继承	插入继承符号		折线	插入一条折线
	联合	插入联合符号		多边形	插入一个多边形

3. 创建新实体

(1) 在 CDM 的图形窗口中，单击工具选项板上的 Entity 工具，如图 6- 所示。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，如果连续单击，即可创建多个实体。点击 Pointer 工具或右击鼠标，释放 Entity 工具。

(2) 双击已创建的实体，打开【Entity Properties】窗口，在此可以编辑实体的相关属性。在此窗口【General】标签中可以输入实体的名称、代码、描述等信息，如图 6- 所示。

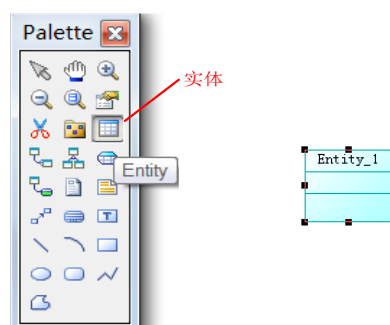


图 6-4 新建实体

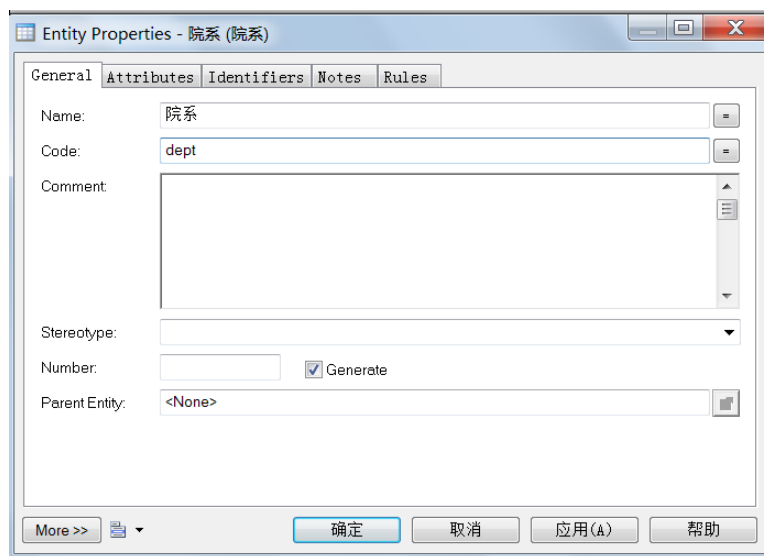


图 6-5 【General】选项页

(3) 不像标准的 E-R 图中使用椭圆表示属性，在 PowerDesigner 中添加属性只需打开【Attribute】选项页，在该窗口上可以添加该实体的属性，【Attribute】选项页如图 6- 所示。在该选项页上，可以输入每个属性的名称、代码以及数据类型等。

其中【Attributes (属性)】选项卡中主要的选项的含义如下：

- ◆ Name: 属性名; Code: 属性代号; Data Type: 数据类型; Length: 数据长度; Precision: 精度。Domain: 域，表示此属性取值的范围。
- ◆ M: 即 Mandatory，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。
- ◆ P: 即 Primary Identifier，是否是主标识符，表示实体唯一的标识符，对应我们常说的主键。当选中某属性为主标识符时，在【Identifiers】选项页中自动出现该标识符的 name 和 code。
- ◆ D: 即 Displayed，D 列表示该属性是否在图形窗口中所示。

说明：

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。

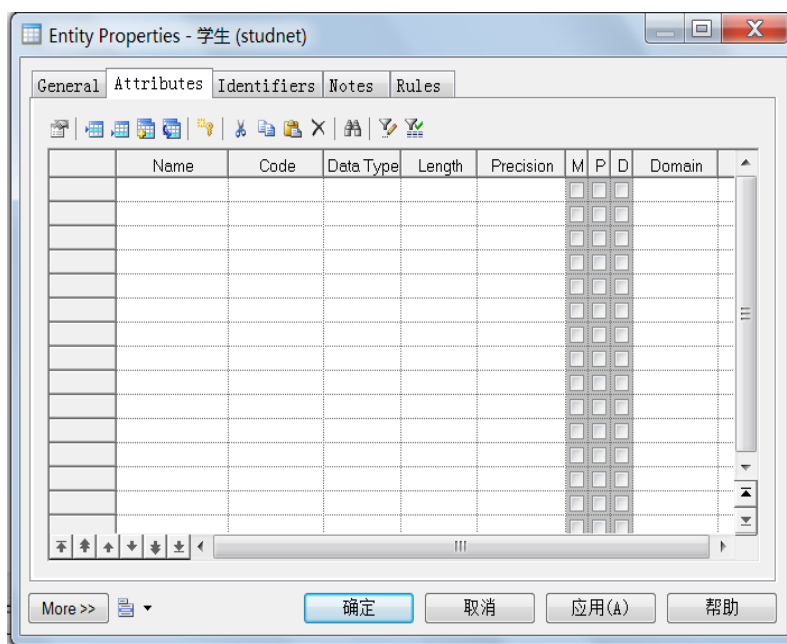


图 6-6 【Attribute】选项页

(4) 在图 6- 所示的窗口中，创建实体的各个属性。本例中创建了“院系”实体，属性有“院系编号”和“院系名”，其中“院系编号”是主标识符（图 6- 8）。在选择数据类型时，可以在对应的数据类型列中，点击省略号按钮，打开【Standard Data Types】窗口，在这个窗口中，选择相应的数据类型即可，如图 6- 所示。

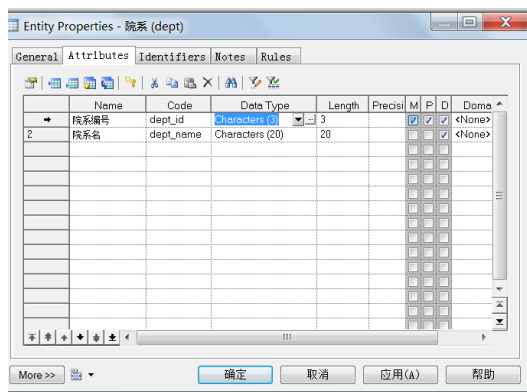


图 6-7 “院系”实体的属性

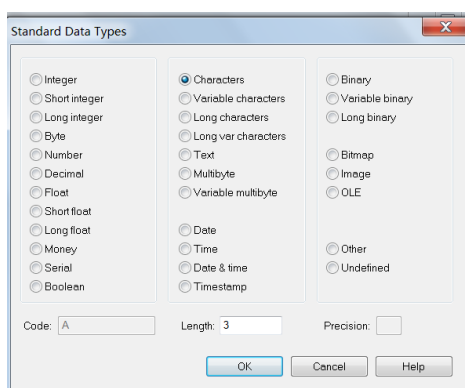


图 6-8 【Standard Data Types】窗口

类似的方法可以创建“课程”实体。已创建的“院系”和“课程”实体如图 6- 所示。

院系		
院系编号	<pi> Characters (3)	<M>
院系名	Characters (20)	
Identifier_1	<pi>	

课程	
课程编号	Characters (10)
课程名	Characters (20)
学分	Number (5,1)
学时	Integer

图 6-20 院系和课程实体

4. 创建实体间的继承联系

通过特殊化或概化方法产生的实体类型之间的联系称为继承联系。

◆ 特殊化：在实体集内部分组并把这些分组存放在不同的实体类型中的过程称为实体集的特殊化。

◆ 概化：从多个实体集的公共属性中抽象出一个公共实体类型的过程为实体集的概化。

继承联系的一端是具有普遍性的实体集，为父实体集；另一端连接的是具体特殊的一个或多个实体集，为子实体集。例如：“银行账户”是“借记卡账户”与“信用卡账户”的父实体，相应地，“借记卡账户”与“信用卡账户”是“银行账户”的子实体。

另外在继承联系中，还可以分为互斥性继承联系和非互斥性继承联系。

◆ 互斥性继承联系：父实体中的一个实例只能在一个子实体中。例如：“帐户”主实体下的“个人账户”与“公司账户”两个子实体之间的联系是互斥的。

◆ 非互斥性继承联系：父实体中的一个实例可以在多个子实体中。例如：“职工”父实体下的“干部”与“教师”子实体之间属于非互斥继承联系，教师有可能也是干部，干部有可能也是教师。

说明：如果工具栏上的 Inheritance 图标默认是禁用的。打开方法如下：**【Tools】 | 【Model Options】 | 【Model Settings】 | 【Notation】** 设为“E/R+Merise”就行了。

因此，继承允许定义一个实体为另一个更一般的特例，涉及到继承的实体间既有着共同相似的特征，也有不同的特征。

如在**【学生成绩管理】**系统中，有学生实体，有教师实体，其实他们都是人员，所以我们可以抽象出一个人员的实体，具有教师和学生的公共属性：姓名、性别、出生日期、电子邮件、家庭住址等属性。而教师实体中又具有教师编号、办公电话、职称等属性这是学生没有的，所以不能放在人员实体中；同样，学生实体中也有学号、入学年份等属性是教师没有的。

在 CDM 中创建继承的操作步骤如下：

(1) 在工具面板中左键单击继承（Inheritance）工具，在子实体上单击鼠标左键，按住不放，拖拽鼠标到父实体后才松开，这样就建立了父子实体之间的 Inheritance 关系。如图 6- 所示，人员是父实体，教师和學生是子实体。

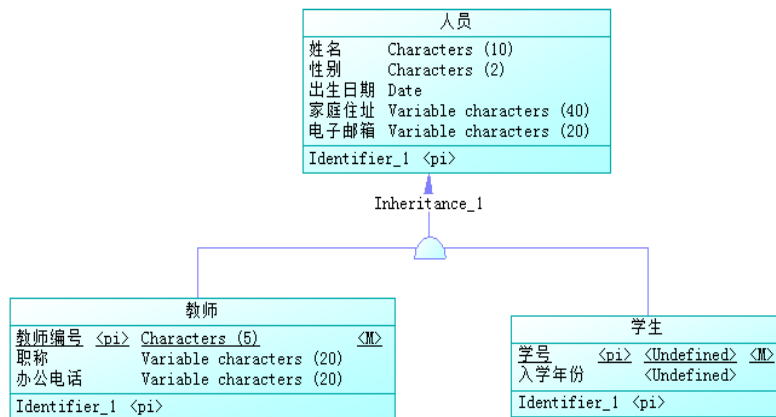


图 6-21 使用继承

(2) 双击新建的继承关系线，打开继承关系属性窗口，在此可以设置继承的相应属性。比如在【General】选项页中（图 6- ），可以输入该继承关系的 name 和 code 值，以及子实体间是否互斥：复选框 Mutually exclusive children 选中，代表是互斥性继承联系，那么在继承关系线上的半球形图标里会出现有叉叉图标。

子类可以只继承父类的主键，也可以继承所有的字段，可通过继承属性页面进程设置，双击新建的继承关系线，打开继承关系属性窗口，切换到【Generation】选项页，调整红色椭圆标注区域的单选框的选择即可，如图 6 -所示。

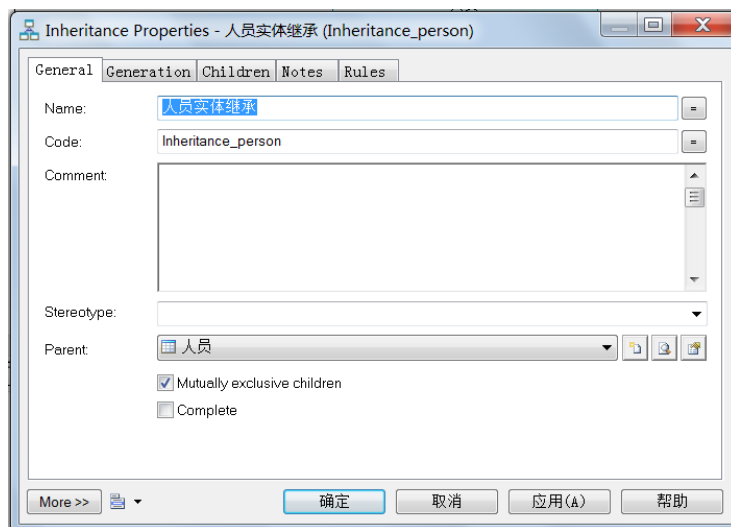


图 6-22 继承关系属性窗口的【General】选项页

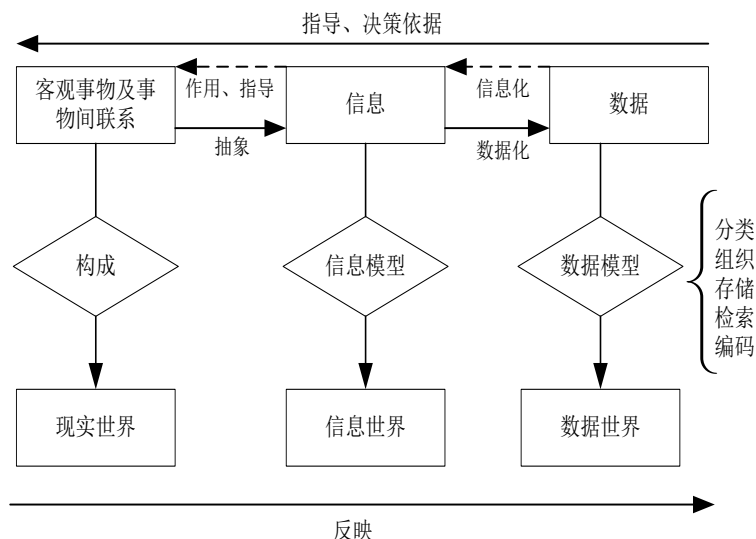


图 6-23 继承关系属性窗口的【Generation】选项页

该窗口中的部分选项说明如下：

◆ Generate parent

表示继承联系中的父实体会生成 PDM 中的表或 Class 图中的类。

◆ Generate children

① 选择 Inherit all attributes 表示继承联系中的子实体生成 PDM 中的表或 Class 图中的类，并且继承父实体中的所有实体属性

② 选择 Inherit only primary attributes 表示继承联系中子实体生成 PDM 中的表或 Class 图中的类，但只继承父实体中的标识符属性。

设置继承属性后的图如图 6- 所示。

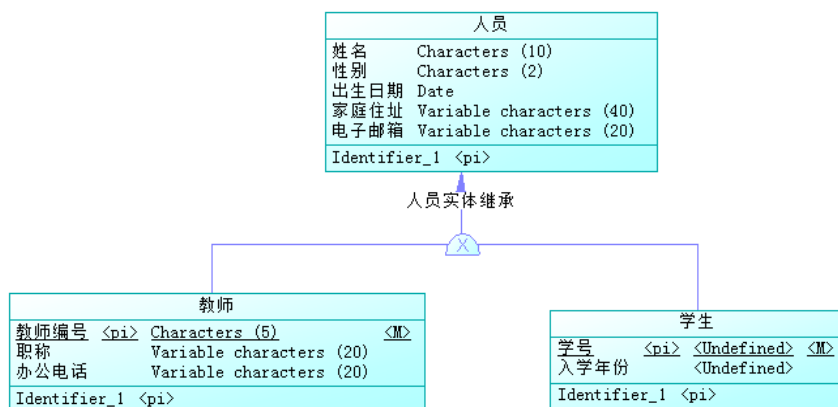


图 6-24 人员实体继承

5. 创建实体之间的联系

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，联系是具有方向性的。

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)。在 CDM 中，联系是用实体间的一条线来表示的，联系的具体含义是通过线两端的符号来表示的。如图 6- 所示了四种基本的联系。

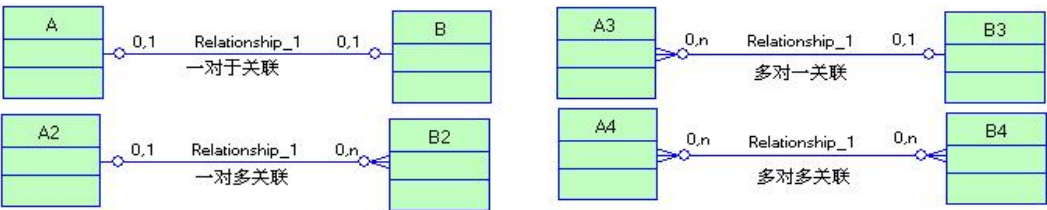


图 6- 25 四种基本的联系示意图

在两个实体间创建联系的步骤如下：

- (1) 在工具面板中左键单击联系 (Relationship) 工具。
- (2) 在实体 A 上单击鼠标左键，按住不放，拖拽鼠标到实体 B 上后才松开，这样就建立了实体 A 和实体 B 之间的 Relationship。如图 6- 所示，在院系实体和教师实体之间、院系实体和学生实体之间分别建立了联系。
- (3) 双击新建的联系，打开联系属性窗口。在【General】选项页中，输入联系 的 Name 和 Code（可以采用默认的），如图 6-27 所示。
- (4) 之后切换到 Cardinalities 标签页（图 6- ），进行详细的设置。

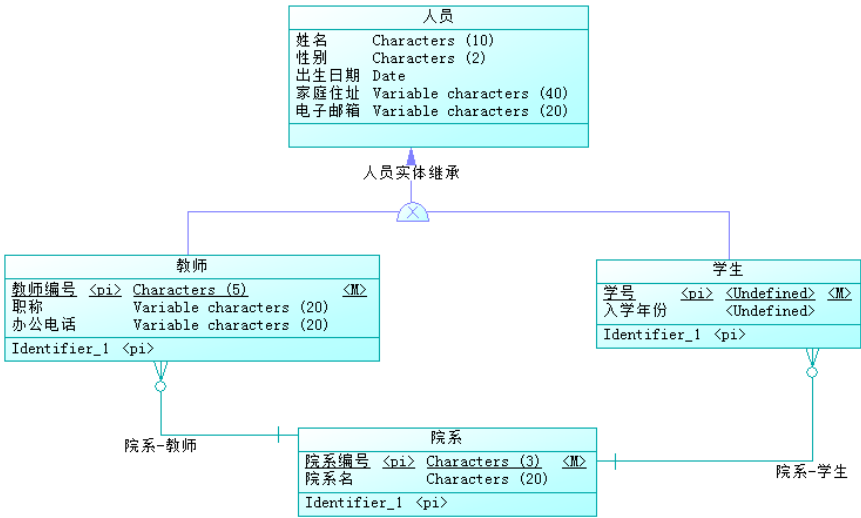


图 6- 26 建立联系

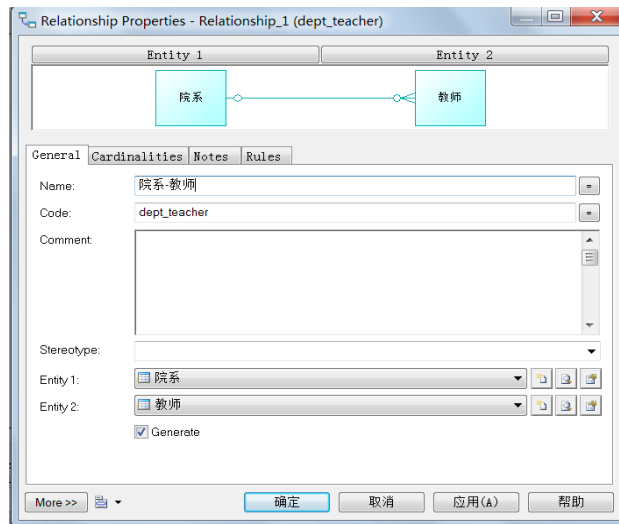


图 6-27 联系属性窗口的【General】选项页

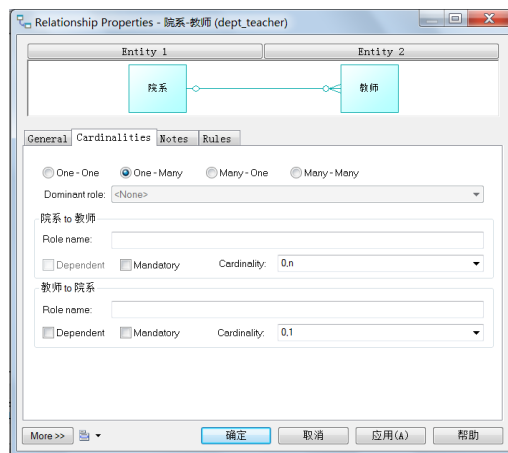


图 6-28 联系属性的【Cardinalities】选项页

该窗口中可配置的属性说明如下：

◆ one to one（一对一）， one to many（一对多）和 many to many（多对多）是最常见的联系属性，用来配置一个实体中的实例数与另一个实体中的实例数的比值关系。

◆ Dominant role：用于指明联系的主从表关系，仅作用于 one to one 的联系。在 A,B 两个实体型的联系中，如果 $A \rightarrow B$ 被指定为 dominant，那么 A 为联系的主表，B 为从表，在生成 PDM 的时候，A 的 Identifier 字段会被引用到 B 实体（如果不指定，A 的 Identifier 会被引用到 B 实体，B 的 Identifier 也会被引用到 A 实体）。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。图 6-表示了客户和合作伙伴商户之间的一对一联系。



图 6-29 一对一联系示意图

这种联系的属性页面的设置如图 6- 所示：



图 6- 30 设置 Dominant role

◆ **Mandatory:** 强制表示实体间的联系是否是可选的。在 CDM 中用穿过联系线的一条短直线表示强制，用联系线上的一个小圆圈表示可选，如图 6- 所示。图 6- 表达了两个含义：

- ① 一个用户必须归属于且只能归属于一个客户；
- ② 一个客户可以有多个用户，也可以一个用户都没有。



图 6- 31 Mandatory 的示意图

两个实体间实例的比值关系以及联系是否强制，可用出现在联系线两端的下述符号表示，如图 6-32 所示。

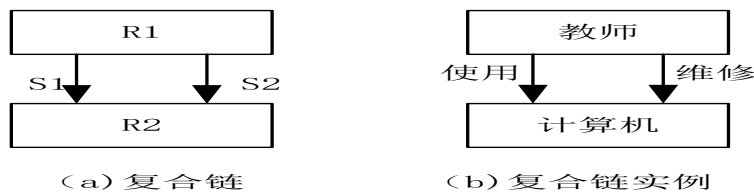


图 6-32 实体间的关系

◆ **Dependent:** 每个实体都有自己的标识符 (Identifier，用于唯一标识实体中的一条记录，由实体的一个属性字段或多个属性字段组成)，如果两个实体之间发生关联，其中一个实体的标识符所包含的属性字段是构成另外一个实体的标识符的一部分，则称后一个实体依赖于前一个实体，后一个实体部分的被前一个实体确定。在 CDM 中依赖联系用一个三角形表示，三角形的顶点指向被依赖的实体，图 6- 表示了客户联系信息和客户信息之间的依赖关系。

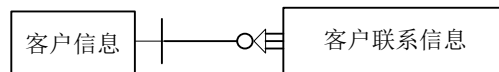


图 6- 33 依赖关系

◆ **Cardinality: 基数。**联系具有方向性,每个方向上都有一个基数。比如“系”与“学生”两个实体之间的联系是一对多联系,或者说“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”。

6. 创建关联 (Association)

Association 也是一种实体间的连接,在 Merise 模型方法学理论中,Association 是一种用于连接分别代表明确定义的对象的不同实体,这种连接仅仅通过另一个实体不能很明确地表达,而通过“事件 (Event)”连接来表示。也就是说,实体和实体之间存在着联系 (多对多),但是这种联系本身还存在其他的属性,这些属性如果作为一个明确的实体来表示又不是很合适,所以就使用了 Association 来表达,这种关系之间一般是一个“事件”虚实体,也就是说是一个动词对应的实体。

比如,在【学生成绩管理系统】中,有学生实体和课程实体,一个学生可以选择多门课程,一门课程有多个学生来上课,学生选修课程会有这门课程的成绩,因此成绩是联系的属性。我们就可以创建一个“选课”的 Association,其中记录成绩信息;同样,课程和教师也是多对多的联系,联系的属性可有上课教室等,这样,也可以建立“授课”的 Association,其中的属性是上课教室。

创建 Association 的步骤为:

- (1) 在 CDM 的图形窗口中,单击工具选项板上的 Association 工具。再单击图形窗口的空白处,在单击的位置就出现一个关联符号,如果连续单击,即可创建多个关联。点击 Pointer 工具或右击鼠标,释放 Association 工具。
- (2) 双击已创建的 Association,打开属性窗口,可以编辑其属性。在【General】选项页中可以输入该关联的 name 和 code 信息,如图 6- 所示。在【Attributes】选项页中以添加该关联的属性,如图 6- 所示。

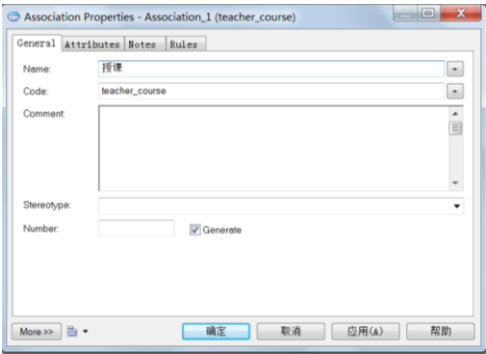


图 6- 34 关联属性的【General】选项页
属性修改后的关联如图 6- 所示。

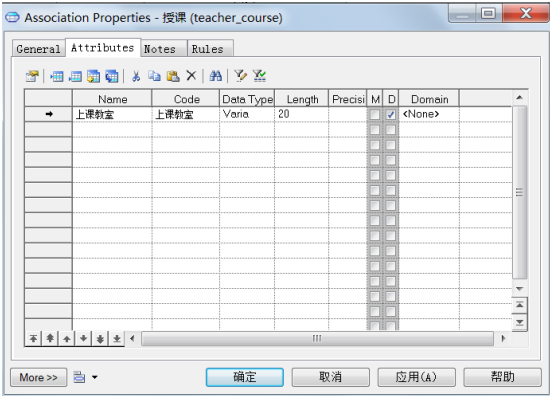


图 6- 35 关联属性的【Attributes】选项页

授课

上课教室 Variable characters (20)

选课

成绩 Number (6, 1)

图 6- 36 “授课”和“选课”关联

7. 添加关联连接

在工具面板上选中 Association Link，从实体拖动到已创建的关联，即可建立一个关联连接，如图 6-所示。

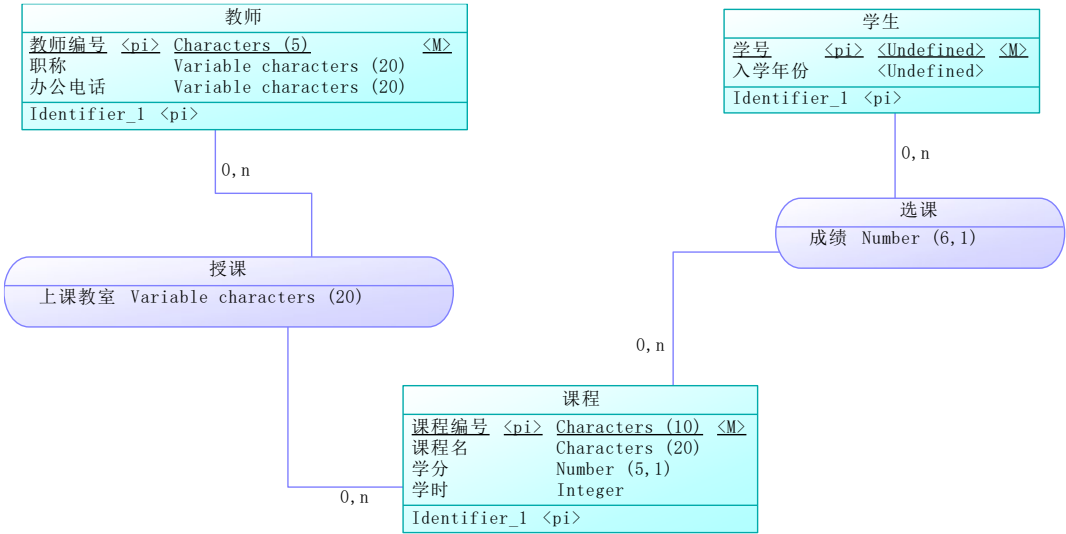


图 6-37 建立关联连接

经过以上的介绍，我们可以完整的创建出【学生成绩管理系统】的 CDM，如图 6- 所示。

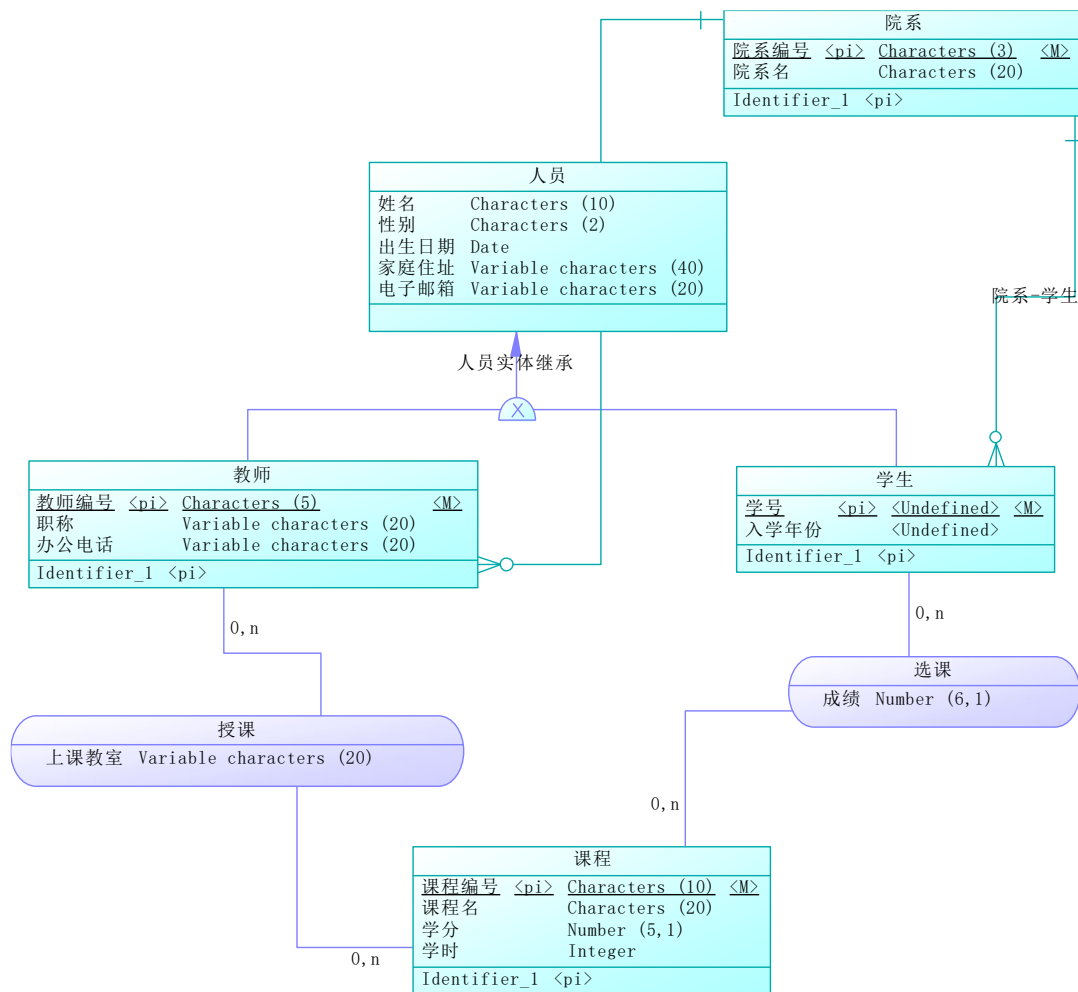


图 6-38 学生成绩管理系统的 CDM

6.7.3 CDM 生成 PDM

由创建好的 CDM 可以直接生成 PDM。操作步骤如下：

(1) 选择菜单栏上的【Tools】|【Generate Physical Data Model】，打开 PDM Generation Options 窗口，选择【Generate new Physical Data Model】，DBMS 项选择对应的数据库（本例中选择 SQL Server 2008），在 Name 和 Code 文本框中输入生成的物理模型的 Name 和 Code。如图 6- 所示。

(2) 切换到 Detail 选项页，可以设置生成的各类索引的命名规则，可根据具体项目的命名规则更改；选中 Check model，模型将会在生成之前被检查；如果不想被检查，取消选中即可。如图 6- 所示。

(3) 切换到 Selection 选项页，列出 CDM 的所有对象，可以选择对哪些对象进行转换，一般默认全部选中，如图 6-41 所示。

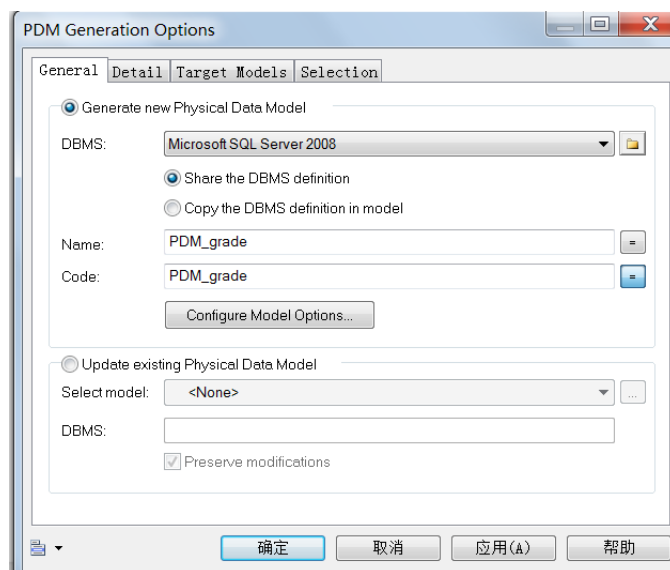


图 6-39 生成 PDM 的【General】选项页

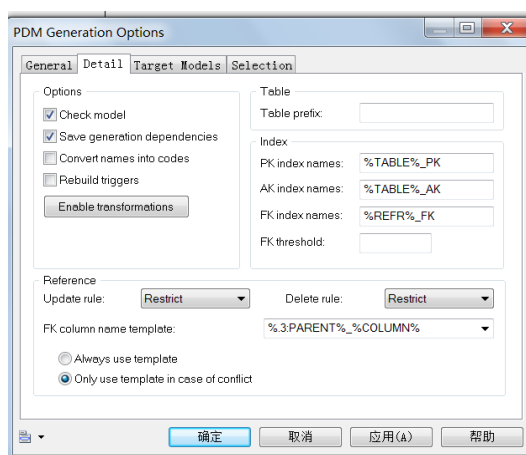


图 6-40 Detail 选项页

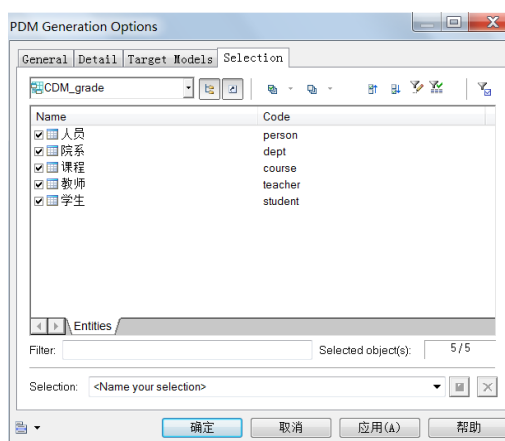


图 6-41 Selection 选项页

(4) 确认各项设置后，点击“确认”按钮，即生成相应的 PDM 模型。由于在第二步中选中了 Check model，所以在生成 PDM 时，会先对 CDM 进行检查，检查结果可能会包含 ERROR，

这样就不能成功生成 PDM，需要先将错误修改。

图 6- 显示了【学生成绩管理系统】的 PDM。

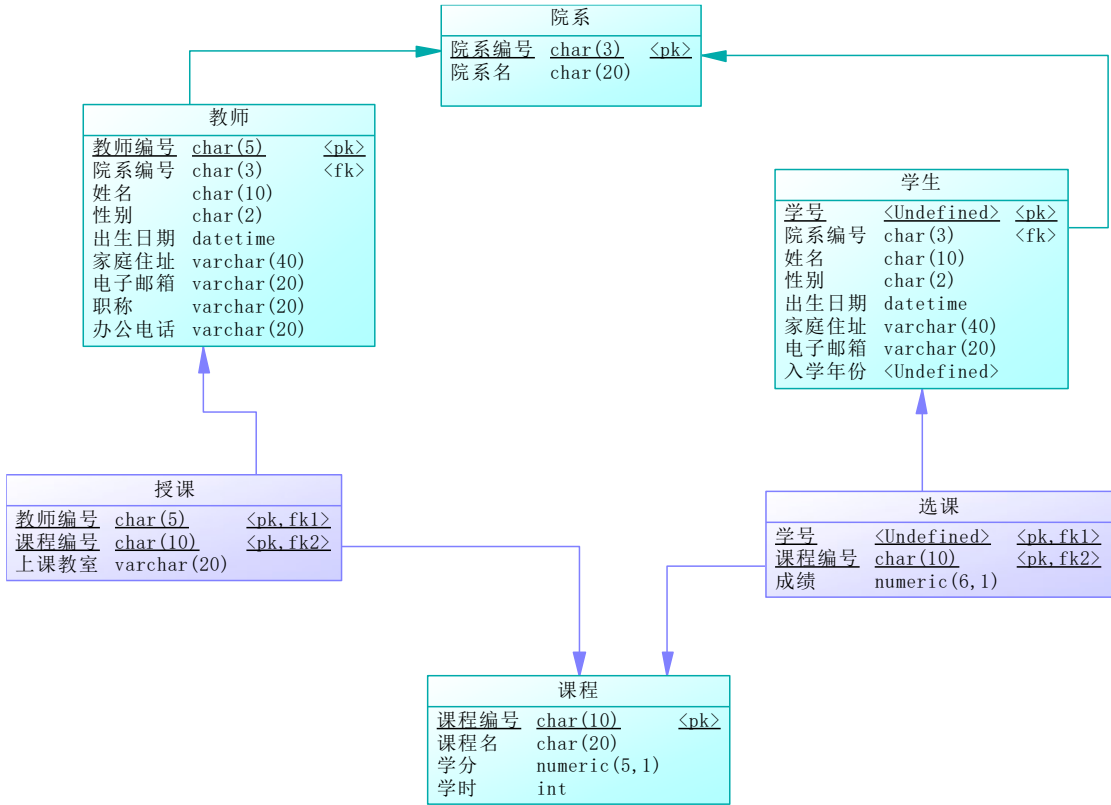


图 6- 42 学生成绩管理系统的 PDM

6.7.4 生成数据库

PDM 生成数据库建表脚本的步骤如下：

(1) 选择菜单栏【Database】|【Generate Database】，打开 Database Generation 窗口（图 6- ），其中包括生成数据库的各种参数选项。

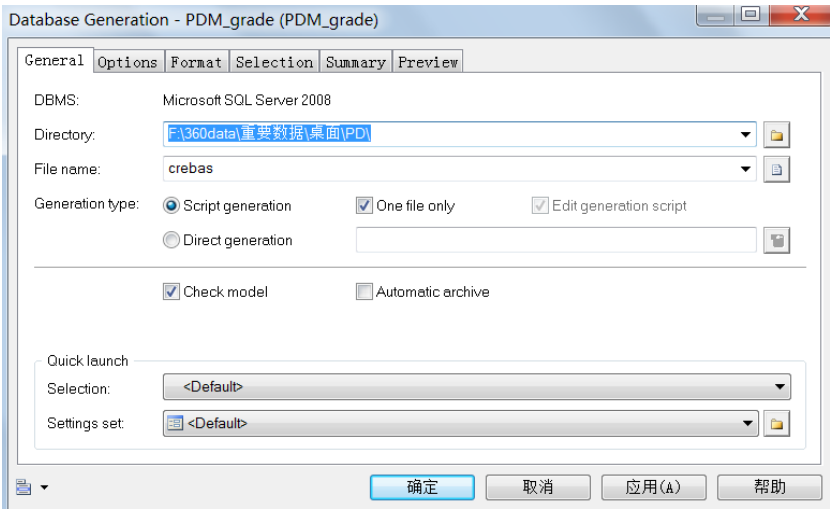


图 6- 3Database Generation 窗口

(2) 在 Director 后选择脚本文件的存放目录，并在 File 文本框输入脚本文件名称。勾选上 One file on，表示所生成脚本将包含于一个文件中。在 Generation 选项栏中选择 Script general 单选框，确认生成数据库方式为直接生成脚本文件。

(3) 点击【确定】按钮后，系统在 check model 没有错误的情况下，即可在指定路径下生成一个脚本文件。

(4) 在 SQL Server 2008 中，选择要建表的数据库，本例为 grade 数据库。打开脚本文件，执行后即可完成在 SQL Server 中的建表。如图 6- 所示。

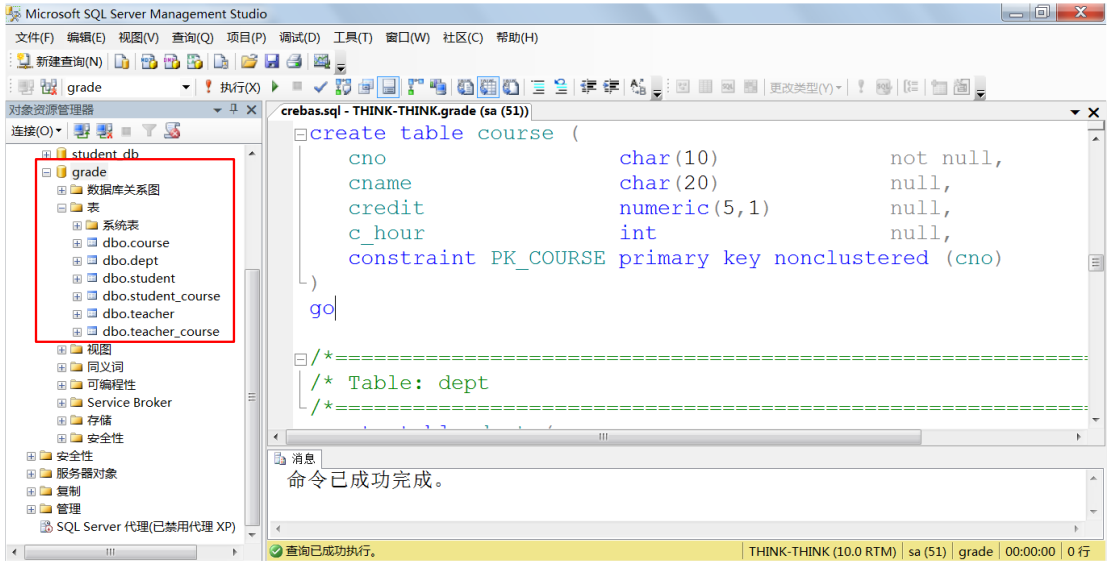


图 6- 44 在 SQL Server 中执行脚本文件

6.7.5 生成 REPORT

- 从 CDM 或 PDM 都可以生成 HTML 格式或 RTF 格式的数据字典。以 RTF 为例，步骤如下：
- (1) 右键单击 PDM 工程名称，选择【New】|【Report】，新建一个 Report 文件。
 - (2) 在出现的【新建 report】中，选择 report name，language 和 report 模版，如图 6- 所示。
 - (3) 在出现的窗口中，可以调节在 report 中显示的内容。从 Available items 中拖拽需要显示的部分到 Report items 窗口，可以修改输出选项的内容。
 - (4) 选择某一项，并右击鼠标，在快捷菜单中选择【layout】，在 list layout 窗口中可以修改输出内容的展示格式，如图 6- 所示。

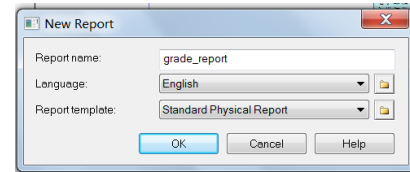


图 6- 9 新建 report

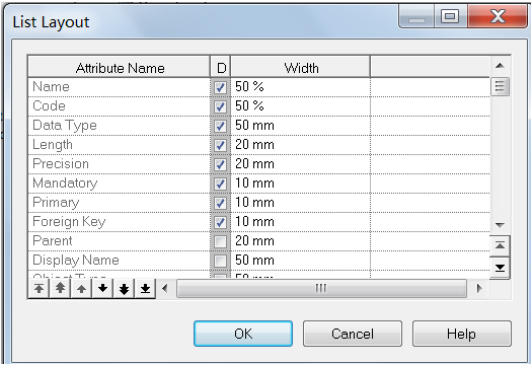


图 6- 10 list layout 窗口

(5) 确认各种设置完成后, 右键单击新建的 Report 文件名, 选择【Generate】|【RTF】即可生成 RTF 格式的 Report 文件, 如图 6-47 所示。

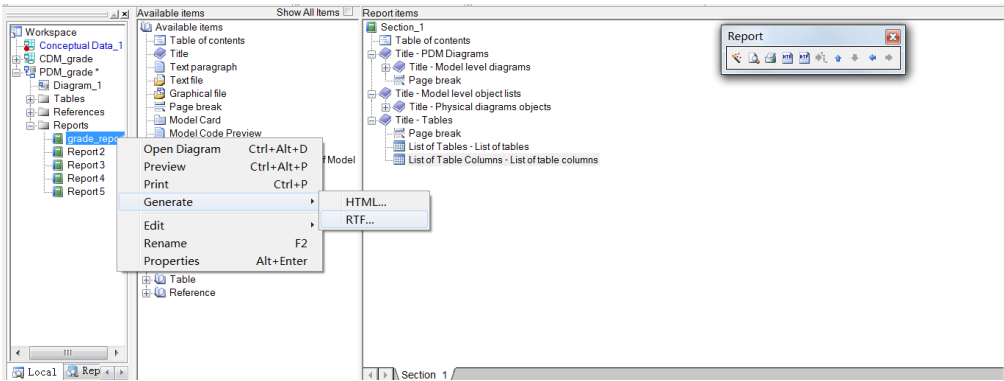


图 6-11 生成 RTF 文件

以下列出了 report 中的各表结构, 如表 6-1——表 6-6 所示。

表 6-1 学生表

Name	Code	Data Type	Length	Mandatory	Primary	Foreign Key
入学年份	enroll_date	Date				
出生日期	birthdate	datetime				
姓名	name	char(10)	10			
学号	sno	Char(10)		X	X	
家庭住址	address	varchar(40)	40			
性别	sex	char(2)	2			
电子邮箱	e-mail	varchar(20)	20			
院系编号	dept_id	char(3)	3	X		X

表 6-2 教师表

Name	Code	Data Type	Length	Mandatory	Primary	Foreign Key
出生日期	birthdate	datetime				
办公电话	phone	varchar(20)	20			
姓名	name	char(10)	10			
家庭住址	address	varchar(40)	40			
性别	sex	char(2)	2			
教师编号	tno	char(5)	5	X	X	
电子邮箱	e-mail	varchar(20)	20			
职称	rank	varchar(20)	20			
院系编号	dept_id	char(3)	3	X		X

表 6-3 授课表

Name	Code	Data Type	Length	Mandatory	Primary	Foreign Key
上课教室	上课教室	varchar(20)	20			
教师编号	tno	char(5)	5	X	X	X
课程编号	cno	char(10)	10	X	X	X

表 6-4 课程表

Name	Code	Data Type	Length	Mandatory	Primary	Foreign Key
学分	credit	numeric(5,1)	5			
学时	c_hour	int				
课程名	cname	char(20)	20			
课程编号	cno	char(10)	10	X	X	

表 6-5 选课表

Name	Code	Data Type	Length	Mandatory	Primary	Foreign Key
学号	sno	Char(10)		X	X	X
成绩	grade	numeric(6,1)	6			
课程编号	cno	char(10)	10	X	X	X

表 6-6 院系表

Name	Code	Data Type	Length	Mandatory	Primary	Foreign Key
院系名	dept_name	char(20)	20			
院系编号	dept_id	char(3)	3	X	X	