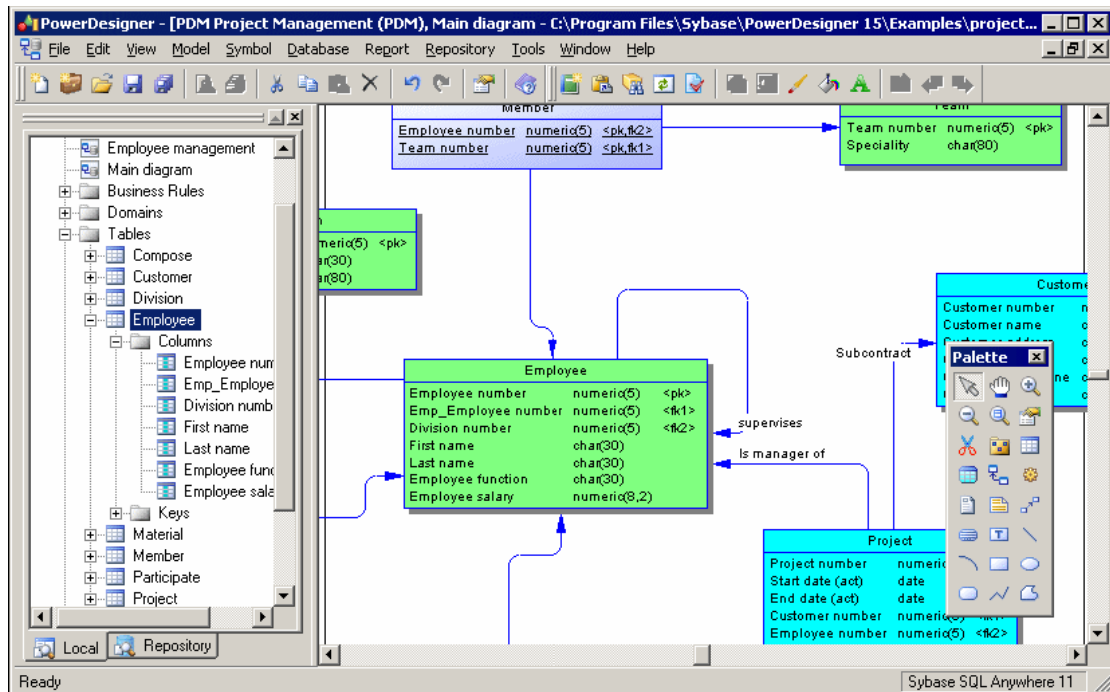


目录

1	PowerDesigner 简介.....	3
2	使用 PowerDesigner 环境.....	5
2.1	对象浏览器中的对象.....	5
2.2	PowerDesigner 的几种模型文件:	6
3	概念数据模型 (CDM)	9
3.1	创建概念图.....	10
3.2	创建及删除实体 (Entity) 及关系 (Relationship)	11
3.3	定义和使用域 (Domain)	14
3.4	建立数据项 (Data Item)	17
3.5	定义业务规则 (Business Rule)	19
3.6	定义和使用继承.....	20
4	物理数据模型 (PDM)	26
4.1	新建 PDM	27
4.2	将 CDM 对象转换成 PDM 对象	28
4.3	使用物理数据模型创建数据库脚本	35
4.4	逆向工程.....	37
5	生成数据库报表文件.....	42
6	多维图	46
6.1	创建多维图.....	47
6.2	获得多维对象.....	48
6.3	建立 Cubes	49
6.4	创建 Cube 数据	50

1 PowerDesigner简介

PowerDesigner是Sybase公司的CASE工具集，是图形化、易于使用的企业建模环境。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。利用PowerDesigner可以制作数据流程图、概念数据模型、物理数据模型，可以生成多种客户端开发工具的应用程序，还可为数据仓库制作结构模型，也能对团队设计模型进行控制。它可与许多流行的数据库设计软件，例如：PowerBuilder、Delphi、VB等相配合使用来缩短开发时间和使系统设计更优化。



PowerDesigner 主要包括以下功能：

- 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)

- **自动生产代码，包括：**

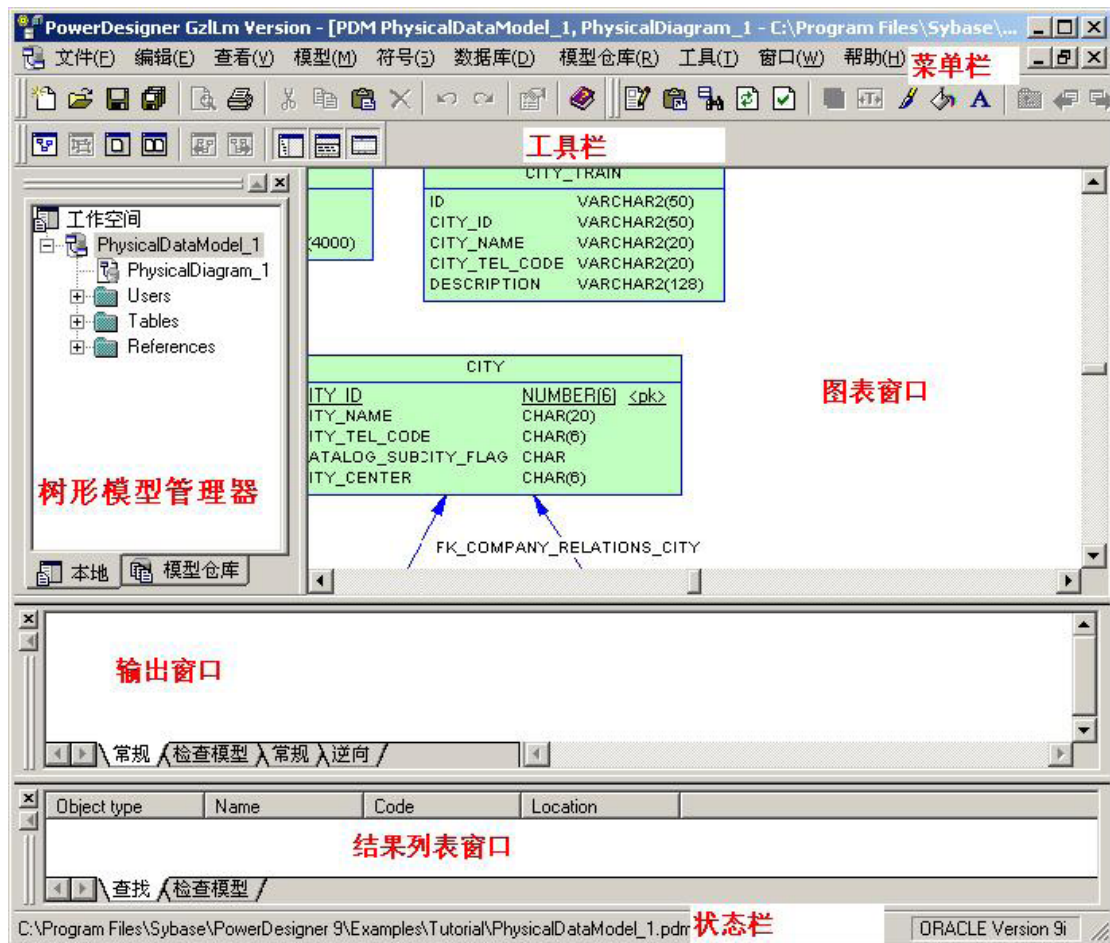
SQL (支持多于50种数据库系统)

Java

.NET

-
- 强大的逆向工程能力
 - 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 使用PowerDesigner环境



- **对象浏览器**

对象浏览器可以用分层结构显示你的工作空间，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。

- **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。

- **结果列表**

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。

2.1对象浏览器中的对象

- **工作空间 (Workspace)**

它是浏览器中树的根，是组织及管理所有设计元素的虚拟环境。你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)**

作为容纳你所有开发出的制品的容器，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，你有两个独立的项目，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。

- **模型对象 (Model Object)**

模型对象泛指与模型相关的所有术语。一些模型对象（比面向对象模型中的类）图形化符号，但另一些对象（比如业务规则）无法显示在图中，只能通过对象浏览器或对象列表访问。

2.2 PowerDesigner的几种模型文件：

- **概念数据模型 (CDM)**

概念数据模型(CDM)帮助你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。概念数据模型(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。一

个概念模型经常包括在物理数据库中仍然不实现的数据对象。CDM可生成LDM、PDM和OOM

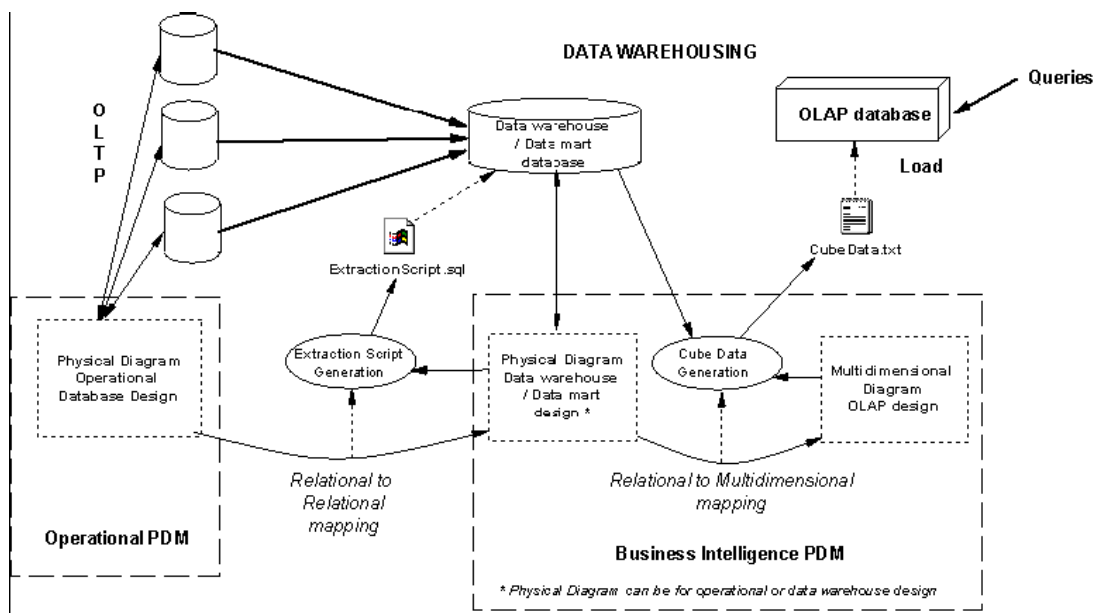
- **逻辑数据模型 (LDM)**

逻辑数据模型(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比概念数据模型(CDM)具体，但不允许定义视图、索引以及其他在物理数据模型(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 (PDM)**

PDM帮助分析数据库中的表、视图及其他对象，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。



- **面向对象模型 (OOM)**

一个OOM包含一系列包、类、接口和他们的关系。这些对象一起形成所有的(或部份)一个软件系统的逻辑的设计视图的类结构。一个OOM 本质上是软件系统的一个静态的概念模型。

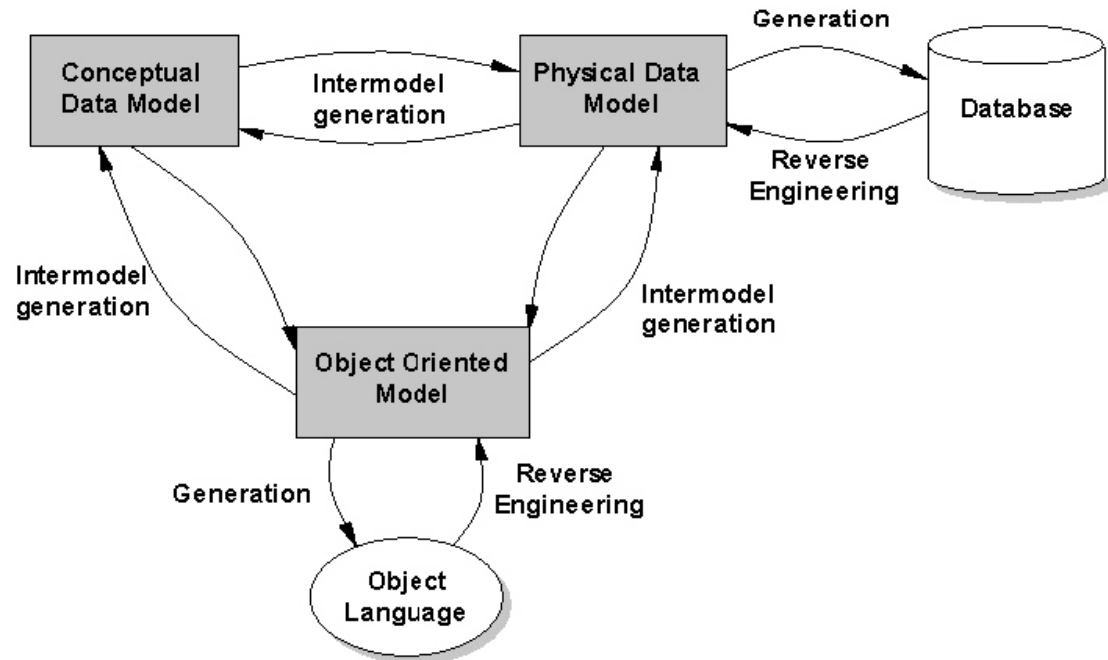
使用PowerDesigner面向对象模型建立面向对象模型(OOM)，能为纯粹地面向对象的系统建立一个OOM，产生Java 文件或者PowerBuilder 文件，或使用一个来自OOM的物理数据模型(PDM)对象，来表示关系数据库设计分析。

- **业务流程模型 (BPM)**

BPM 描述业务的各种不同内在任务和内在流程，而且客户如何以这些任务和流程互相影响。

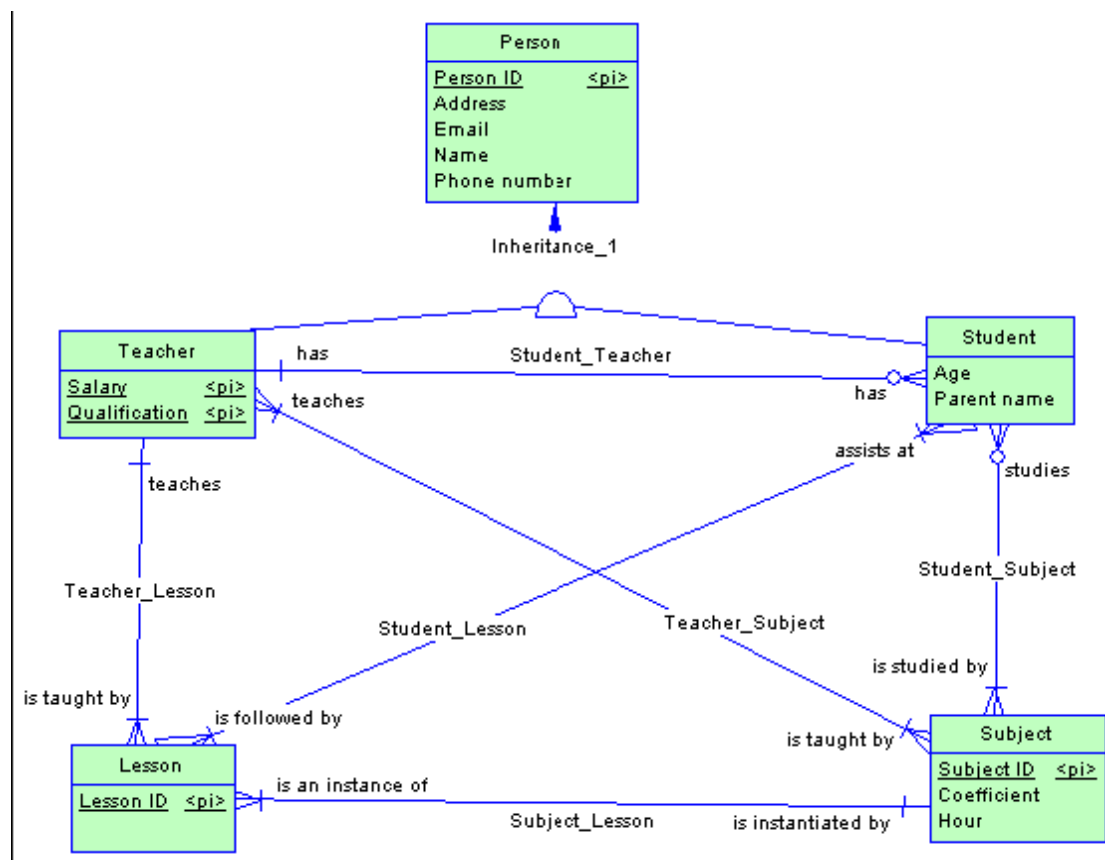
BPM 是从业务合伙人的观点来看业务逻辑和规则的概念模型，使用一个图表描述程序、流程、信息和合作协议之间的交互作用。

以下是其中的 CDM、PDM、OOM 三者的转换关系



CDM、PDM 和OOM 之间的关系

3 概念数据模型 (CDM)











上图含义:



























- 每个teacher可以讲授多个subject, 每个subject可以被多个teacher讲授 (many-to-many)。
- 每个teacher可以讲授多个lesson, 每个lesson只能由一个teacher讲授 (one-to-many)。
- 每个student可选多个lesson, 每个lesson可被多个student选修 (many-to-many)。
- 每个student可以选择多个subject, 每个subject可被多个student选修 (many-to-many)。

概念图中的对象:

名称	工具图标	图形符号	描述
域 (Domain)			数据项有效值的集合
数据项 (Data Item)			基本数据项
实体 (Entity)			人们感兴趣的概念, 如人、地点、事物等
属性 (Entity Attribute)			附属于Entity的信息
标识符 (Identifier)			用于区分实体的属性

			或属性集合
关系 (Relationship)			实体之间的命名连接
继承 (Inheritance)			用于表示实体间一般/特殊关系
关联 (Association)			
关联连接 (Association Link)			Association与Entity之间的连接

概念图的工具面板：

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

3.1 创建概念图

方法1：创建概念模型时可默认创建概念图：

选择File > New Model ，从Model type创建概念数据模型conceptual data model，选择概念图Conceptual Diagram，点击OK，

方法2: 如果已有概念数据模型, 则:

- 在对象浏览窗口中, 在概念数据模型上右键选择 New > Conceptual Diagram。
- Right-click the background of any diagram and select Diagram > New Diagram > Conceptual Diagram。

3.2 创建及删除实体 (Entity) 及关系 (Relationship)

1 按工具面板的实体工具。

当光标移动进图中的时候, 变成实体的形状。

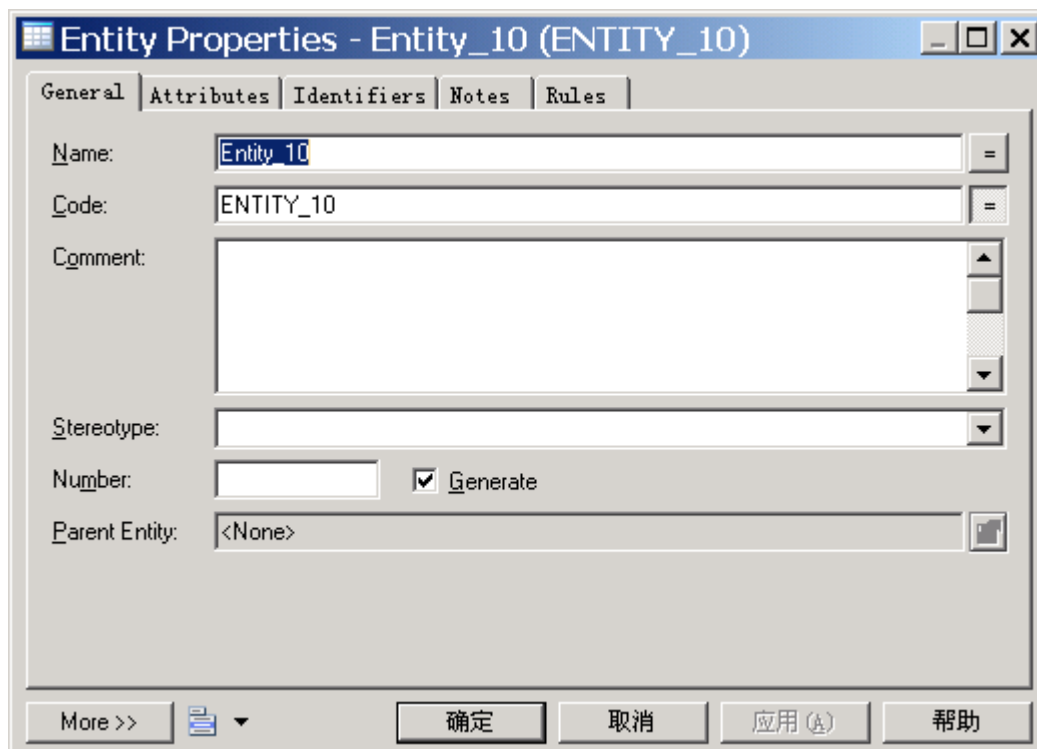
2 在 CDM 图中点击任何一处。

一个实体符号在点击位置出现。实体名字为 Entity_n,n是一个创建对象的次序编号。

3 实体工具仍然是可使用的, 因此点击再一次在CDM图表中产生另外的一个实体。

现在有 CDM 图表的二个实体。

双击实体, 可以看到实体的属性。



其中：**Name**表示该实体的显示名称，是给非技术人员看的。

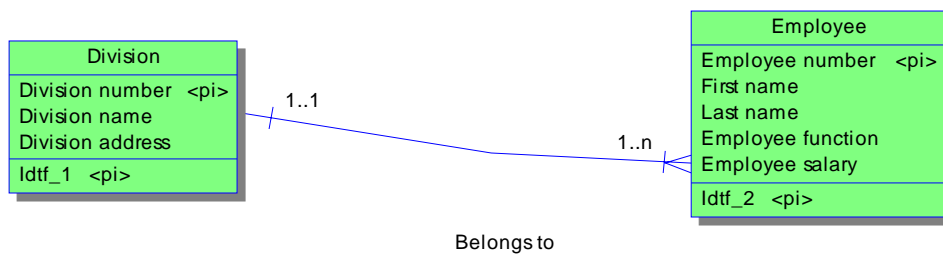
Code是该实体的技术名称，用于生成代码。

Number表示实体数量，也就是未来数据表中有多少条记录。

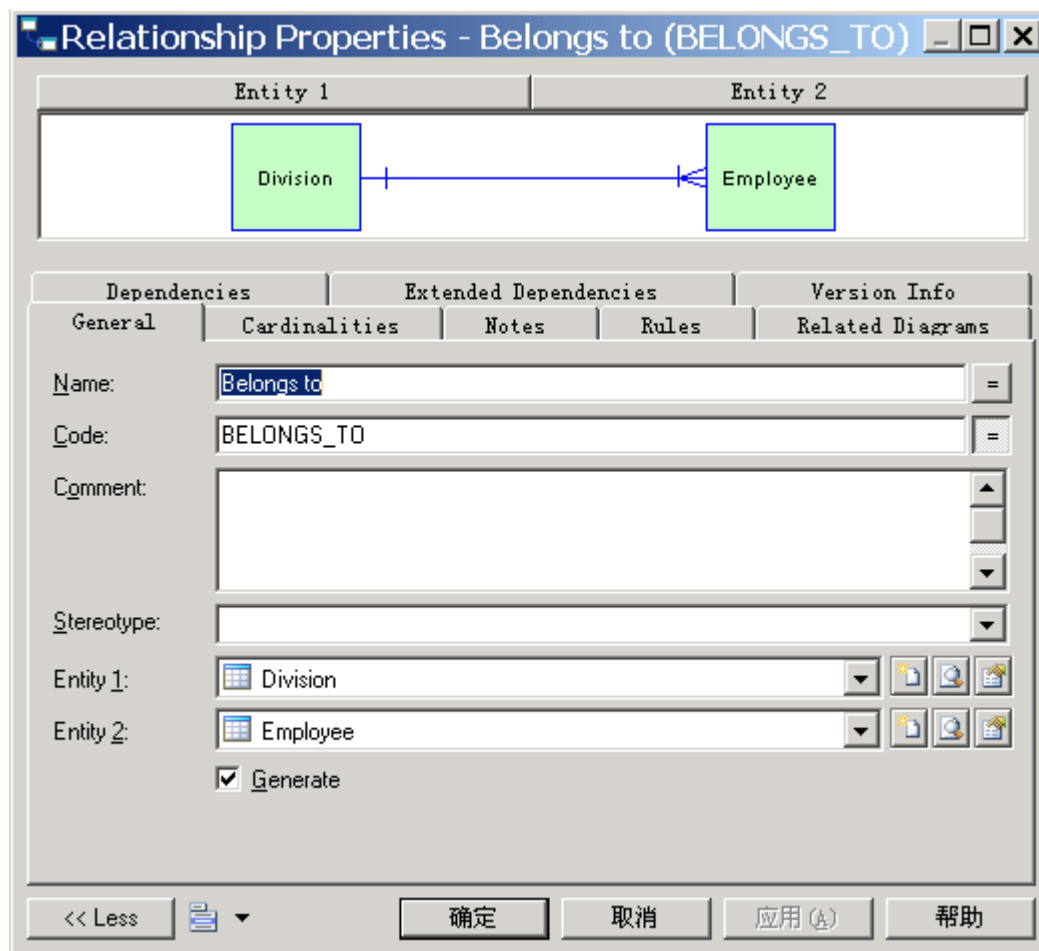
4 点击工具面板的关系工具。

实体工具被现在释放，而且关系工具是可使用的。

5 点击第一个实体之，并且在继续按着鼠标按钮的时候，拖拉光标到第二个实体。在第二个实体之内放开鼠标按钮。这样可以产生关系。



双击关系，可以看到关系的属性。

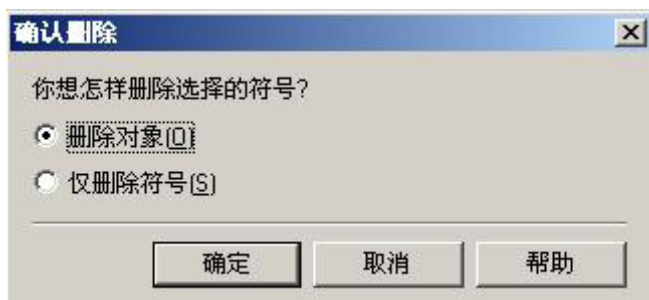


6 点击鼠标右键，释放关系工具。

释放一个工具

一个工具保持可使用直到释放它。要想释放一个工具，可以选择另外一个工具或按鼠标右键。默认地，当按鼠标右键，指针工具被激活。

- 7 点击面板的文本工具。文本工具是现在可使用。
- 8 在关系下面点击光标，一些文本在被长方形指出的区域中出现。
- 9 点击鼠标右键，你释放文本工具。
- 10 双击文本，一个文本框出现。
- 11 在文本框中输入短文本。
- 12 点击OK，文本在图中出现。
- 13 点击文本框的一个柄，按着鼠标左键，拖拉光标到右边直到所有的文本出现，放开鼠标按钮，在图的背景上点击，文本框柄消失。
- 14 点击面板的指针工具。你将使用这个工具选择并且删除图中的符号。
- 15 在实体符号上点击，选择想删除的对象。
- 16 按键盘上的DEL键，确认信息对话框出现，询问如何删除符号。

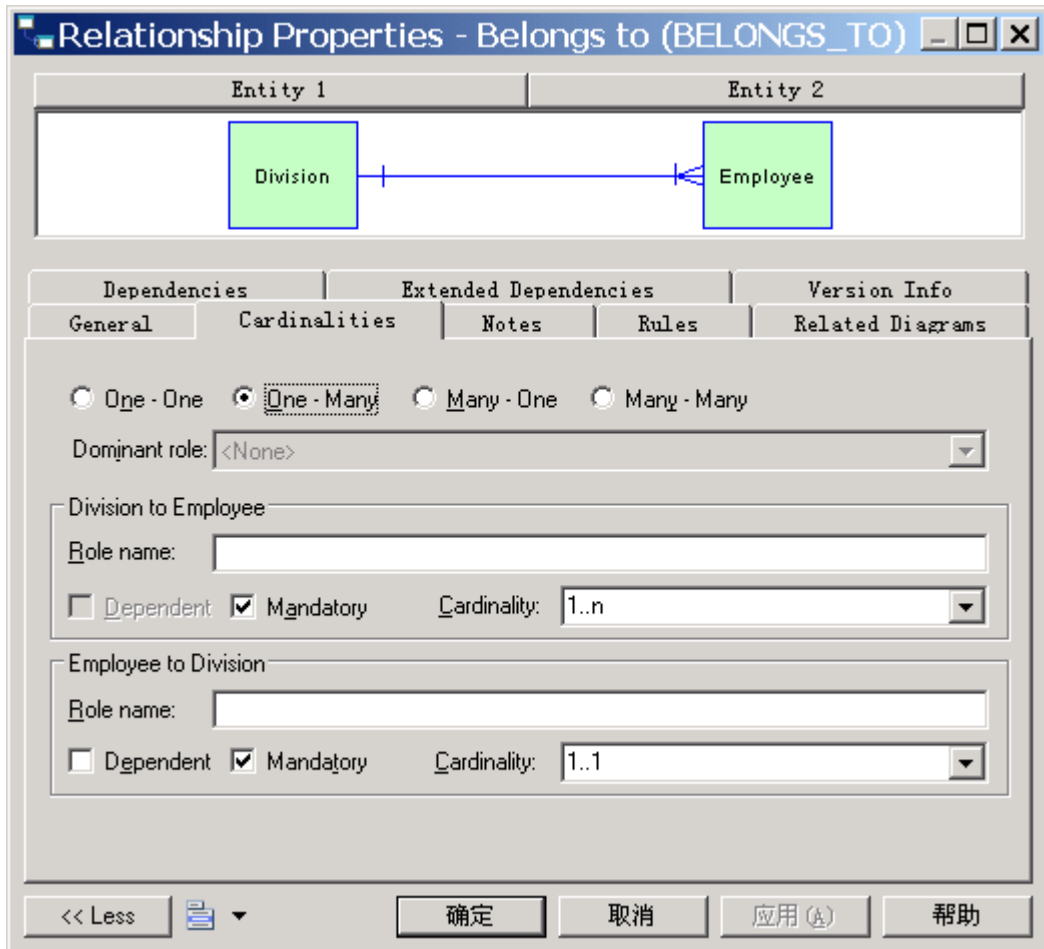


删除对象

如果选择删除对象，将删除图标符号并且删除模型中的对象。如果你只选择删除符号，将只删除图标符号，但是保存模型中的对象。

- 17 点击OK，图表中的实体和联合的关系被移动。对象也从模型删除。
- 18 点击剩余的实体，当点击文本的时候，按着键盘的SHIFT键，两个对象将被选择。
- 19 按DEL键，并且在删除信息出现的时候点击OK，剩余的实体和文本被删除。

- 讨论与讲解：关系（Relationship）的性质。



3.3 定义和使用域 (Domain)

- 关于域

一个域定义能适用于多个数据项 (Data Item) 的标准数据结构。当修正一个域时，将更新全部与域关联的数据项目。当作任何变化的时候，导致数据一致化特性比较容易实现。

- 创建一个新的域

下面将会创建两个域，为钱数定义一个标准化的域，为模型中的百分比定义一个域。

- 1 选择菜单栏的模型域，Model > Domains。

域对话框的列表显示已存在的域。

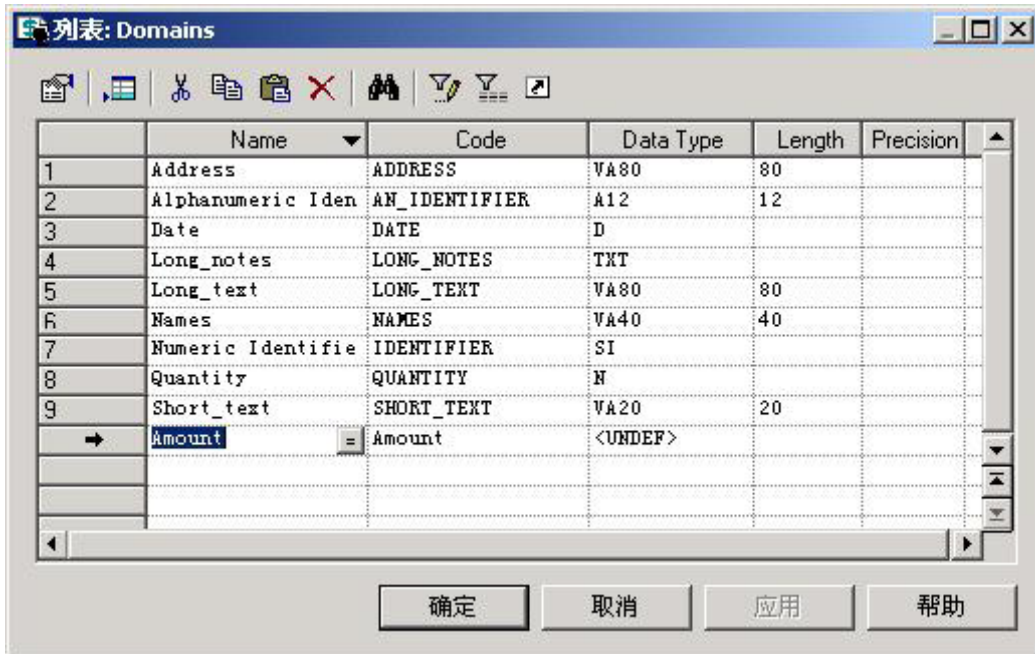
- 2 点击增加一行(add a row)工具。

一支箭头在第一空行的开始处出现，一个默认的名字和代码被进入。

3 在名字列输入Amount。

这是域的名字。相同的代码自动地在代码列中被进入。

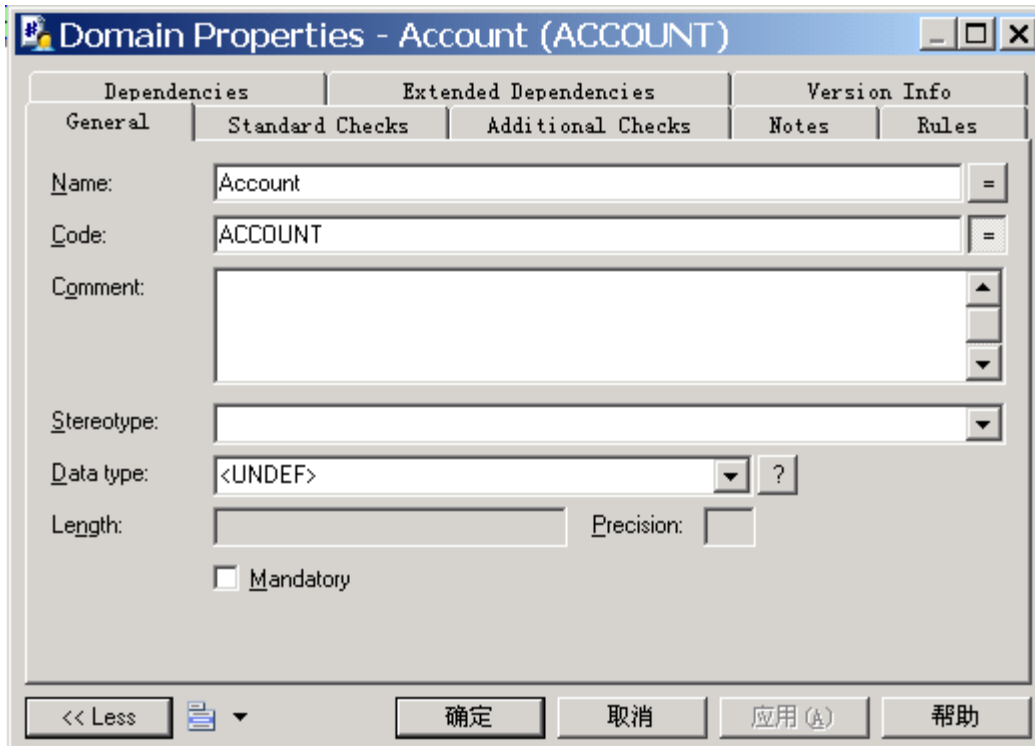
4 点击应用。新创建的域被提交。



5 点击新的域行。

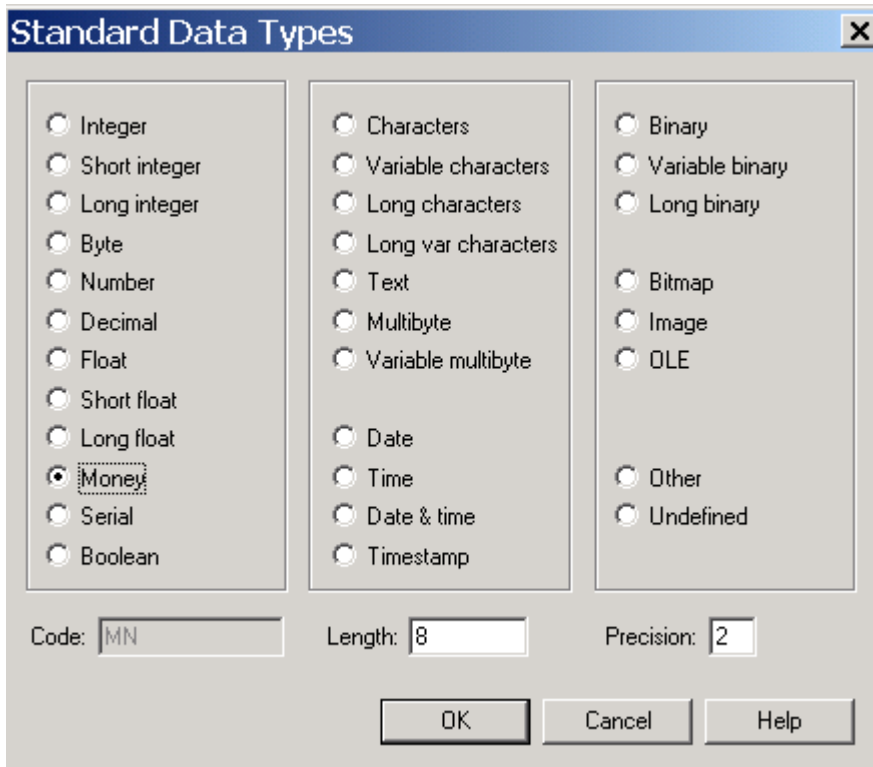
6 点击属性工具。

7 点击紧邻数据类型下拉列表的问号按钮。



8 点击Momey单选按钮。

域现在有一个Money数据类型。一个Money数据类型用一个固定长度的数字来储存数字。然后，当应用这个域用来储存Money的数量数据项目（Data Item）的时候,将会使他们继承这一个数据类型。



- 9 在长度框中打字8。附到这个域的数据项目的最大的长度将会是8。
- 10 在精密框中打字2。附到这个域的数据项目在小数点之后能带二个位置。
- 11 点击OK。

回到域特性页。值MN8,2在数据类型列表框中出现。MN是一个Money数据类型的代码。8指出Money类型能有8个长度。2指出数量有2个小数位精度。

- 12 点击OK，回到域的列表。

- 13 点击增加一行工具。

一支箭头在第一个空白行的开始处,一个默认的名字和代码被输入。

- 14 名称列键入Percent。

这是域的名称。相同的代码自动地在代码列中被进入。

- 15 点击应用。新创建的域被提交。

- 16 点击新的域行。一个箭头在行开始处出现。

- 17 点击属性工具。或在行开始处双击箭头。属性页为新创建的域。

- 18 点击问号按钮旁边数据类型下拉列表框。

28	Sale Terms	SALE_TERMS	VA80	80	
29	State	STATE	A2	2	
30	Store Address	STOR_ADDRESS	VA80	80	
31	Store ID	STOR_ID	A12	12	
32	Store Name	STOR_NAME	VA40	40	
33	Title ISBN	TITLE_ISBN	A10	10	
34	Title Notes	TITLE_NOTES	TXT		
35	Title Price	TITLE_PRICE	MN5, 2	5	2
36	Title Publication	TITLE_PUBDATE	D		
37	Title Text	TITLE_TEXT	VA80	80	
38	Title Type	TITLE_TYPE	VA20	20	
→	TitleAuthor Per	TitleAuthor Perce	<UNDEF>		

确定 取消 应用 帮助

标准数据类型对话框出现。使用这个对话框指定被域影响的数据形式。

19 点击短整数型前面的单选按钮。

代码SI指出Percent域有短整数型数据类型。因为不能为短整数型数据类型指定长度和精度，所以长度和精度框是不可用的。

20 在每一个对话框中点击OK。

3.4 建立数据项（Data Item）

- 关于数据项

一个数据项是一个基本的信息。

- 创建一个新的数据项目

要管理多个作家的多本书，需要对每位作家版税的百分比创建数据项并且为作家列表中的作家名称排序。

1 选择菜单栏的模型数据项，Models > Data Items。数据项的列表显示已存在的数据项目。

2 点击增加一行工具，Add a row。第一个空白行可编辑。

3 键入名称列的TitleAuthor Percent

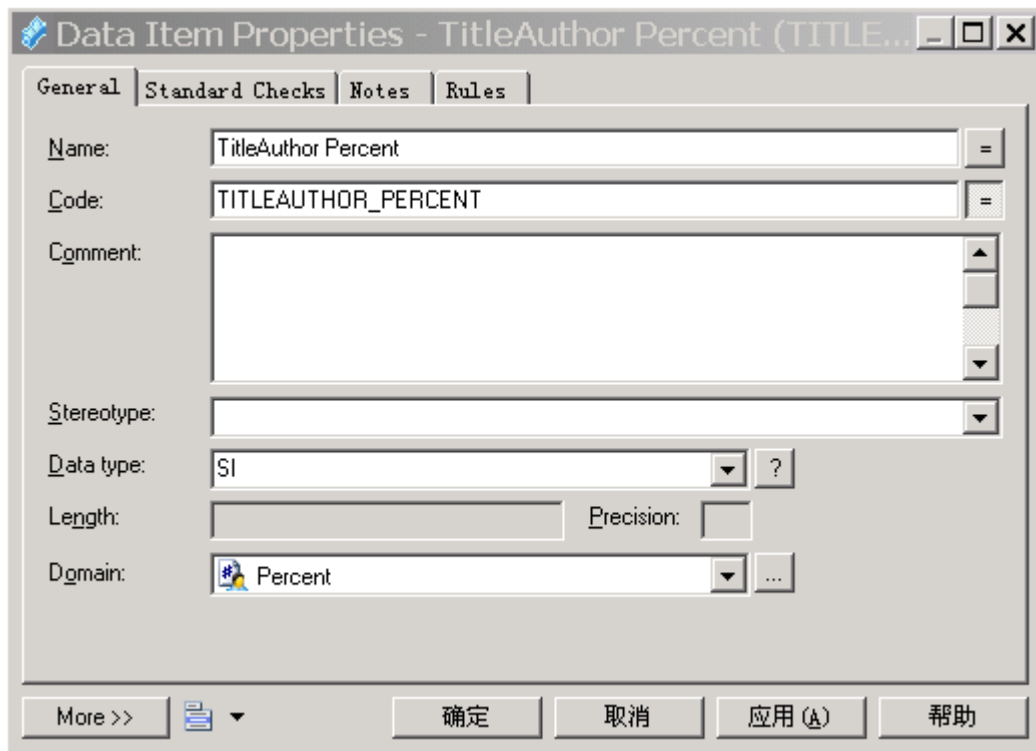
这是数据项的名称。相同的代码自动地在代码列中被输入。

4 点击应用。

新创建的数据项被提交。名称按字母顺序分类，当按应用或列表的OK按钮的时候，所有的名称按字母顺序被分类。列表的名称次序将会以其中任何一个操作而改变。

5 点击刚才新创建的数据项行。一个箭头在行开始处出现。

6 点击属性工具。或在行开始处双击箭头。出现属性页。



7 选择对话框底部域下拉列表框中的Percent.

Percent域的数据类型应用到新的数据项。在数据类型列表中，SI自动被选择。

8 点击确定回到数据项列表。

9 点击增加一行工具。

一个箭头在第一个空白行的开始，一个默认的名称出现。

10 输入TitleAuthor Order作为数据项名称，相同的代码自动在代码列中被输入。

11 点击应用。新创建的数据项被提交。

12 点击新的数据项。一个箭头在行开始处出现。

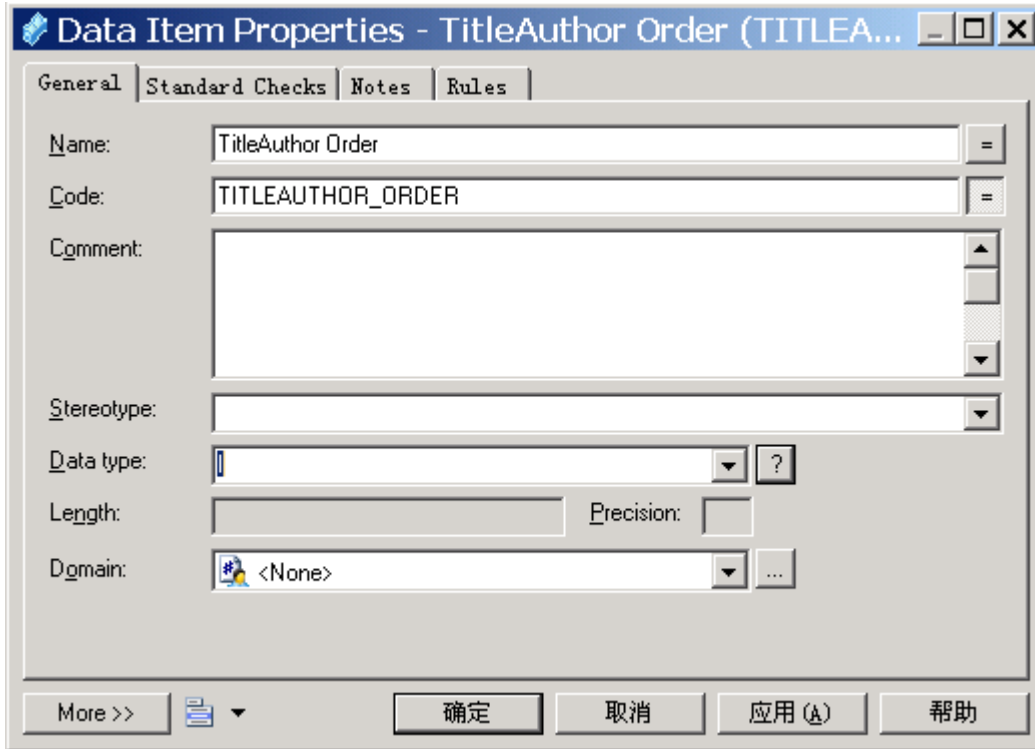
13 点击属性工具。或在行开始处双击箭头。属性页出现。

14 点击问号按钮旁边数据类型下拉列表框。

标准数据类型对话框出现。

15 点击整数型前面的单选按钮。

这给TitleAuthor Order数据项一个整数型数据类型。



16 在每个对话框中点击OK。回到模型的窗口。

3.5 定义业务规则 (Business Rule)

- 关于业务规则

业务规则是业务活动中必须遵循的规则，是业务信息之间约束的表达式，它反映了业务信息数据之间的完整性约束。每当实体中包含的信息发生变化的时候，系统都会检查这些信息是否违反特定的业务规则。

业务规则有的六种类型：事实，定义，公式，确认，需求和约束：

业务规则类型	业务规则说明	业务规则举例
事实	信息系统中存在的事实	一个出版者可能出版一或多个的主题的图书
定义	信息系统中对象的特性	一位作家被一个名字和一个住址识别
公式	信息系统中的计算公式	总金额为所有订单金额的总和
确认	信息系统中需要的确认	支付所有作家一本书的版税百分

	认	比必须为版税的100%
需求	信息系统中功能的详细说明	模型被设计以致版税的总数量不超过总售卖的10%
约束	信息系统中数据之间的约束	销售开始日期必须迟于出版日期

当你使用业务规则约束 CDM 的时候,他们不被转变为可执行的代码。业务规则可能被实现

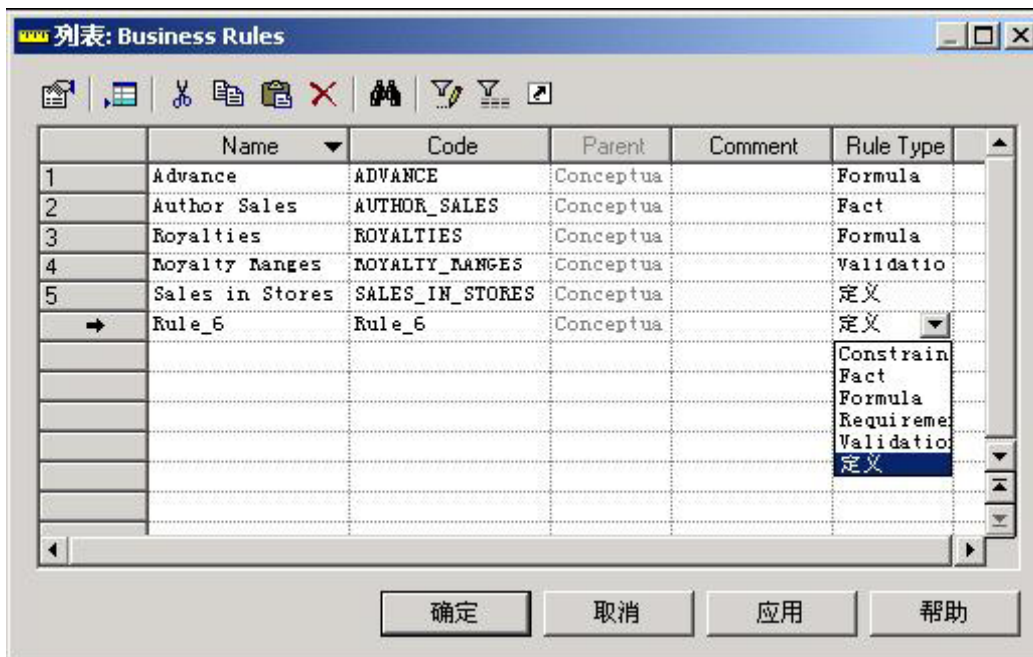
为PDM的约束。

产生一条新的业务规则

下面将会产生一条业务规则指出该如何将版税分配给作家。

- 1 选择模型的业务规则，Models>Business Rules。
- 2 点击增加一行工具，Add a row。

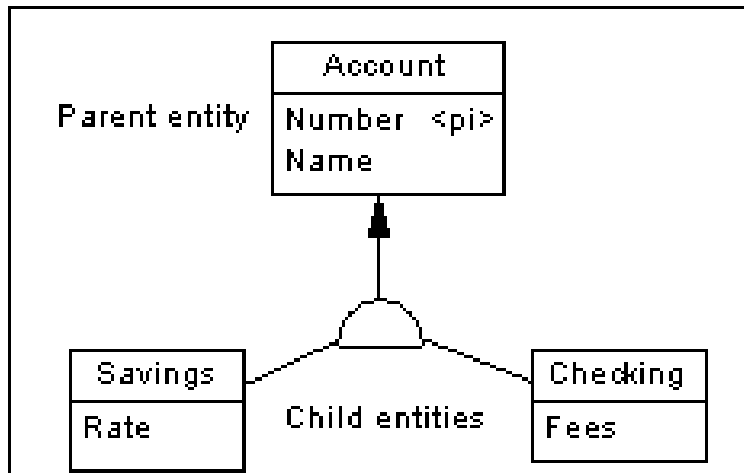
一个箭头在新空行的开始处出现，并且一个默认的名字和代码被输入。默认为定义类型的业务规则



- 3 输入业务规则的名称并选择类型。

3.6 定义和使用继承

- 继承举例



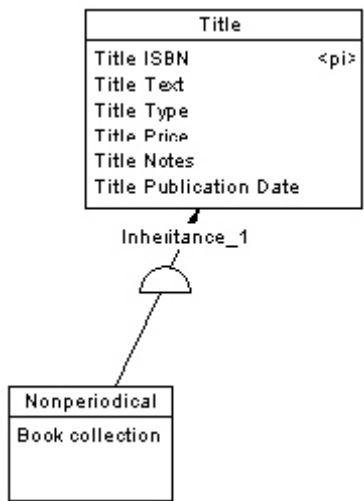
● 继承类别

图形	含义
	标准继承
	互斥继承
	完全继承
	互斥且完全继承

● 创建一个继承连接

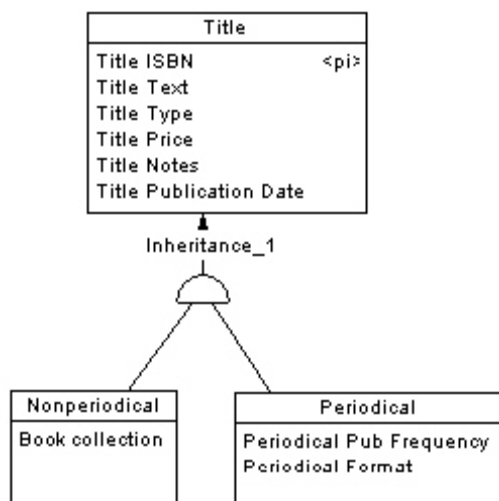
从子实体到父实体创建一个继承连接。你将会从PERIODICAL和NONPERIODICAL实体到TITLE实体定义一个继承。

- 1.选择工具面板的继承工具。
- 2.在NONPERIODICAL实体之内点击，按住鼠标左键不放,拖拉光标到 TITLE 实体。在TITLE实体之内放开鼠标按钮。



一个继承连接出现在这些实体之间。连接中间有一个半圆形，和一个指向TITLE父实体的箭头。NONPERIODICAL是子实体。子实体继承它的父实体。

3 选择工具板上的继承工具，点击半圆形，按住鼠标左键不放，拖拉光标到PERIODICAL实体。在PERIODICAL实体内放开鼠标按钮。继承符号因此改变。



● 定义继承属性

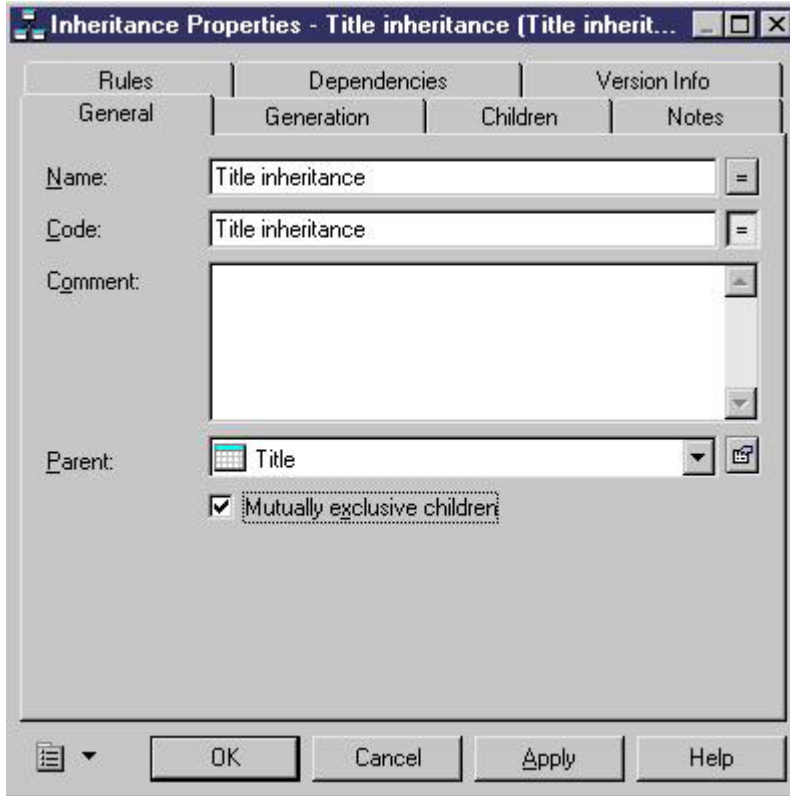
你将会分配一个主题到继承，使它互斥，而且定义它的生成模态。

- 1 点击工具面板的指针工具。
- 2 双击继承连接的中央半圆形。

继承特性页出现。

- 3 类型名称在名称框中继承。这是继承的名称。
- 4 选择子对象互斥的复选框。

因为主题是一份期刊或一份非期刊,不能两个兼有,所以是互斥的。

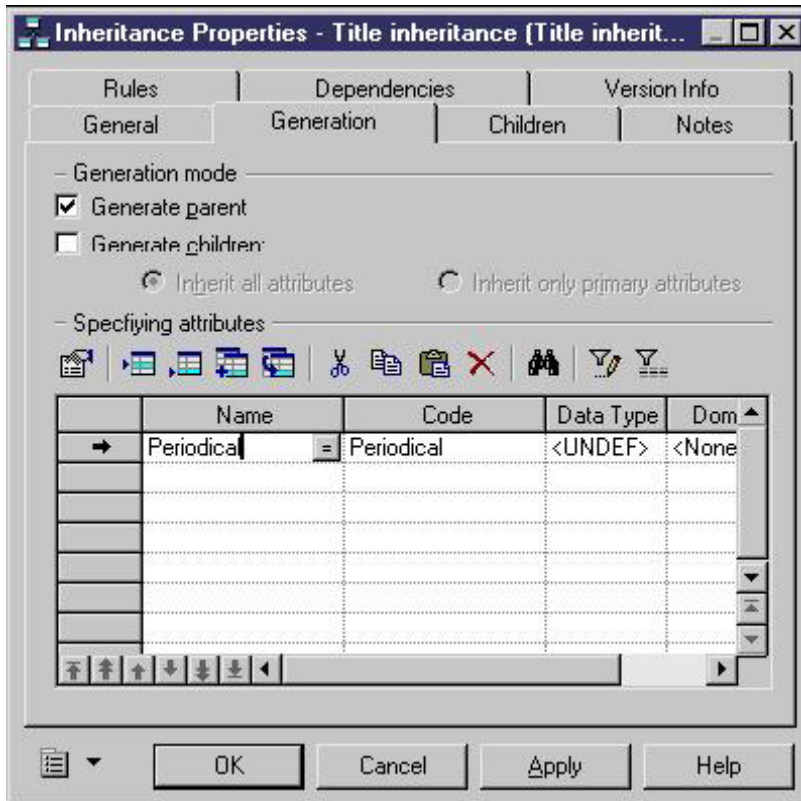


5 点击生成定位键。生成页出现。

6 在生成模式分组框中确定产生子对象复选框没有被选择。

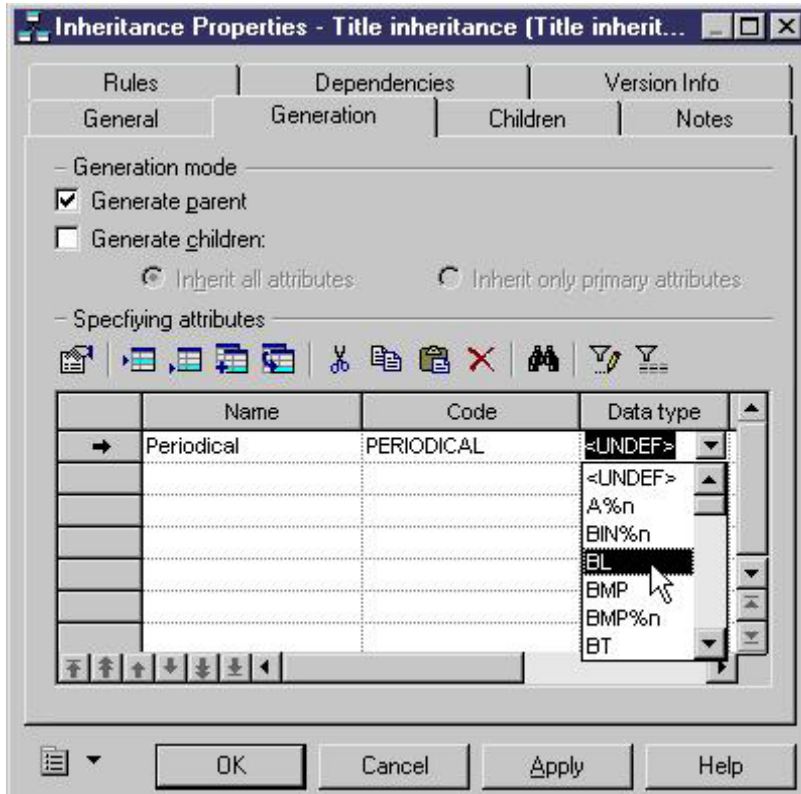
这里的选择将影响继承如何产生物理数据模型PDM。仅仅生成父对象意味着只有一个表将被生成，在这里例子中，你只需要知道在个子对象中的属性是不同的。

7 输入Periodical到名称列中。



- 8 在数据类型列中点击。
- 9 点击下拉列表箭头。一个下拉列表框出现。
- 10 选择来自下拉列表框的 **BL** 。

你为那个指定属性定义 **Boolean(BL)** 数据类型。

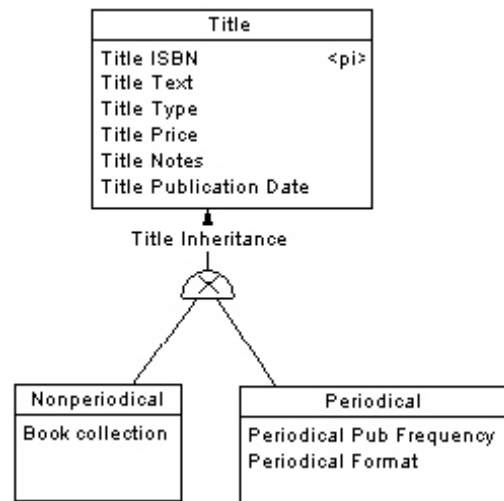


11 移动到右边，知道出现M复选框。

这是强制性的复选框。当选择的时候，它指出当Periodical列被产生的时候不能包含NULL价值。

12 选择M复选框。

13 点击确定。



十字架在半圆形中出现，指出继承是互斥的。




4 物理数据模型（PDM）







物理数模型(PDM)是用于详细定义物理结构和数据查询的数据库设计工具。可以在PDM中使用不同类型的图，这取决于所要设计的目标数据库的类型。而PowerDesigner支持各种数据模型，包括操作型数据库、数据仓库或数据集市，OLAP等类型数据库系统。

● PDM与DBMS

PowerDesigner能够用于创建多种不同类型的DBMS，对于每种类型的DBMS，都包含一个标准定义文件用于在PowerDesigner和DBMS中确定关联而提供一套接口。可以修改装载在PowerDesigner中DBMS，对于每个将要修改的初始DBMS，都可以创建一个相应的新DBMS。

物理图中的对象：

名称	工具图标	图形符号	描述
Table	Table		Collection of rows (records) that have associated columns (fields).
Column			Data structure that contains an individual data item within a row (record),model equivalent of a database field.
Primary key			Column or columns whose values uniquely identify each row in a table, and are designated as the primary identifier of each row in the table.
Alternate key			Column or columns whose values uniquely identify each row in a table,and which is not a primary key
Foreign key			Column or columns whose values depend on and migrate from a primaryor alternate key in another table.
Index			Data structure associated with one or more columns in a table, in whichthe column values are ordered in such a way as to speed up access to data.
Default			[certain DBMSs] A default value for a column.
Domain			Defines valid values for a column.
Sequence			[certain DBMSs] Defines the form of incrementation for a column.
Abstract data type			[certain DBMSs] User-defined data type.
Reference			Link between a primary or an alternate key in a parent table, and a foreign key of a child table. Depending on its selected properties, a reference can also link columns that are independent of primary or alternate key columns.

View			Data structure that results from a SQL query and that is built from data in one or more tables.
View Reference			Link between a table and a view.
Trigger			A segment of SQL code associated with a table or a view.
Procedure			Precompiled collection of SQL statements stored under a name in the database and processed as a unit.
Database			The database of which the PDM is a representation.
Storage			A partition on a storage device.
Tablespace			A partition in a database.
User			A person who can log in or connect to the database.
Role			A predefined user profile.
Group			Defines privileges and permissions for a set of users.
Synonym			An alternative name for various types of objects.
Web service			Collection of SQL statements stored in a database to retrieve relational data in HTML, XML, WSDL or plain text format, through HTTP or SOAP requests.
Web operation			Sub-object of a Web service containing a SQL statement and displaying Web parameters and result columns.

4.1 新建PDM

可以通过三种方式新建PDM

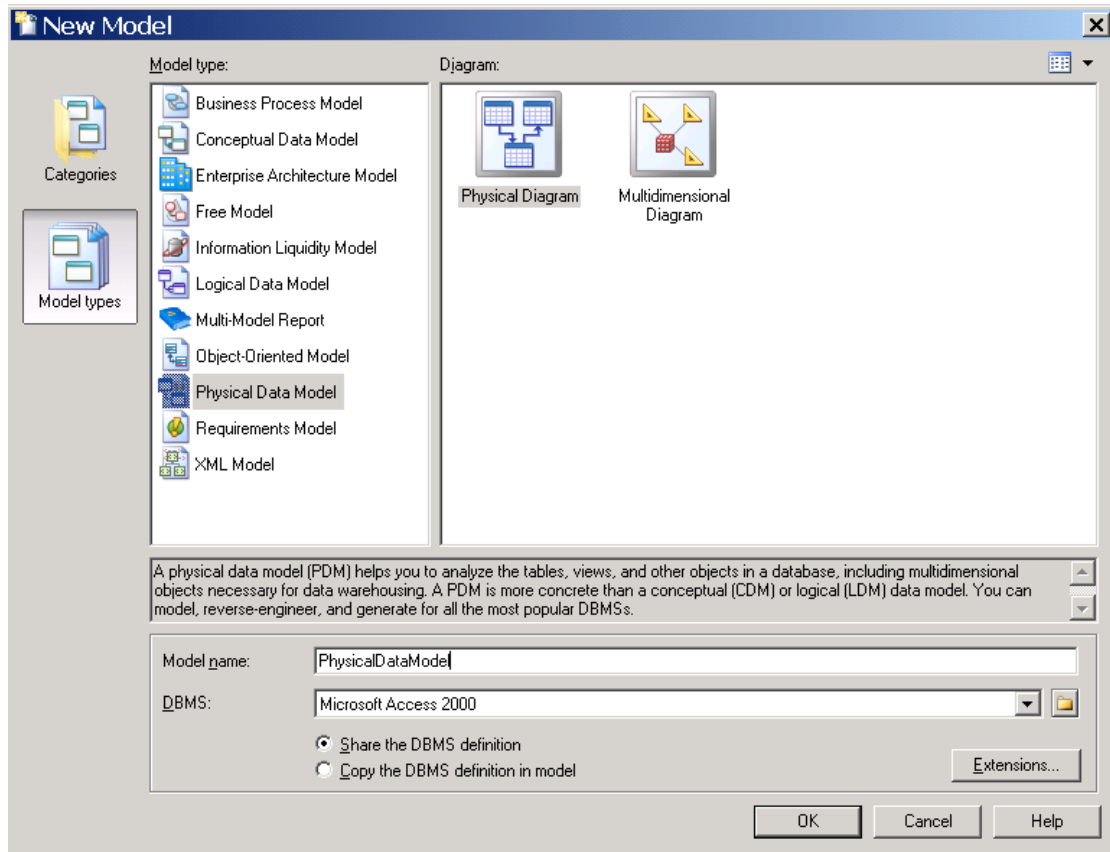
1. 直接创建新PDM

2. 使用模板创建新PDM

3. 通过现有基础创建新PDM，现有元素包括：数据库的反向工程，引入一Erwin模型，从现有CDM或OOM自动生成，从低版本模型迁移等。

下面只简要讲解概述其中一种PDM的创建过程：

(1) 选择New，即打开创建模型选项窗口，如下图：



(2) 选择New model单选框。

(3) 选择左边模型列表中Physical Data Model，同时在DBMS下拉列表中选择相应类型DBMS（当然你也可以在后面的过程中更改DBMS类型），

(4) 在First diagram中选择Physical Diagram,其中列表中Multidimensional Diagram选项用于创建多维（Multidimensional）数据模型。

(5) 单击“确认”，即完成PDM创建过程。

4.2将CDM 对象转换成PDM 对象

当从一个CDM生成PDM时，PowerDesigner将CDM中的对象和数据类型转换为PDM对象和当前DBMS支持的数据类型。

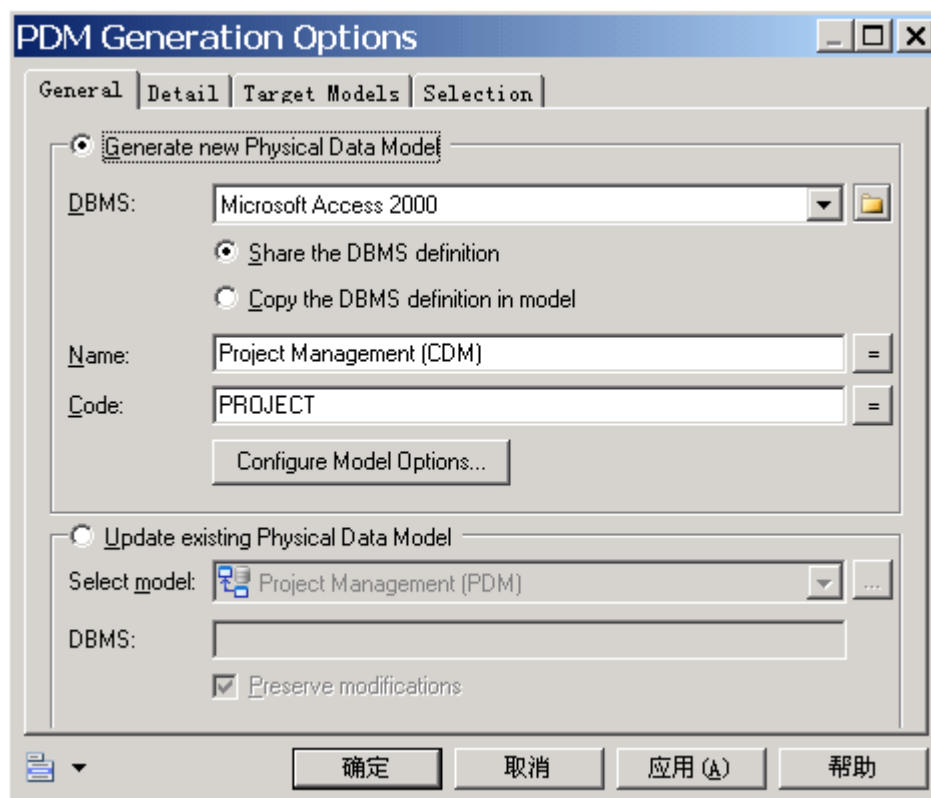
PDM转换概念对象到物理对象的对象关系如下表：

CDM 对象	在 PDM 中生成的对象	备注
实体 (Entity)	表 (Table)	
实体属性 (Entity Attribute)	列 Table Column)	
主标识符 (Primary Identifier)	根据是否为依赖关系确定是主键或外键	
标识符 (Identifier)	候选键 (Alternate key)	
关系 (Relationship)	引用 (Reference)	

同一个表中的两列不能有相同的名称，如果因为外键迁移而导致列名冲突，PowerDesigner会自动对迁移列重命名，新列名由原始实体名的前三个字母加属性的代码名组成。主标识符在生成PDM中的主键和外键，非主标识符则对应生成候选键。

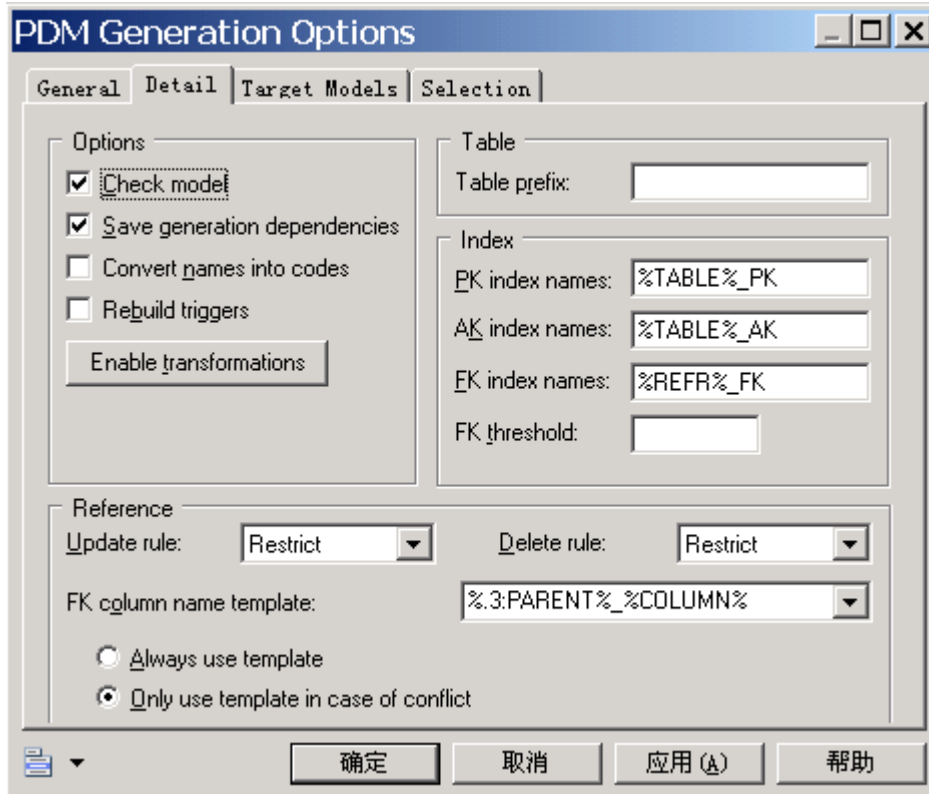
在PDM中生成的键类型取决于CDM中用于定义一个Relationship的基数和依赖类型。

- 1 选择Tools—>Generate Physical Data Model。PDM生成选项对话框出现。
- 2 从数据库管理系统下拉列表框选择一种数据库系统系统。
- 3 选择 Share 单选按钮。



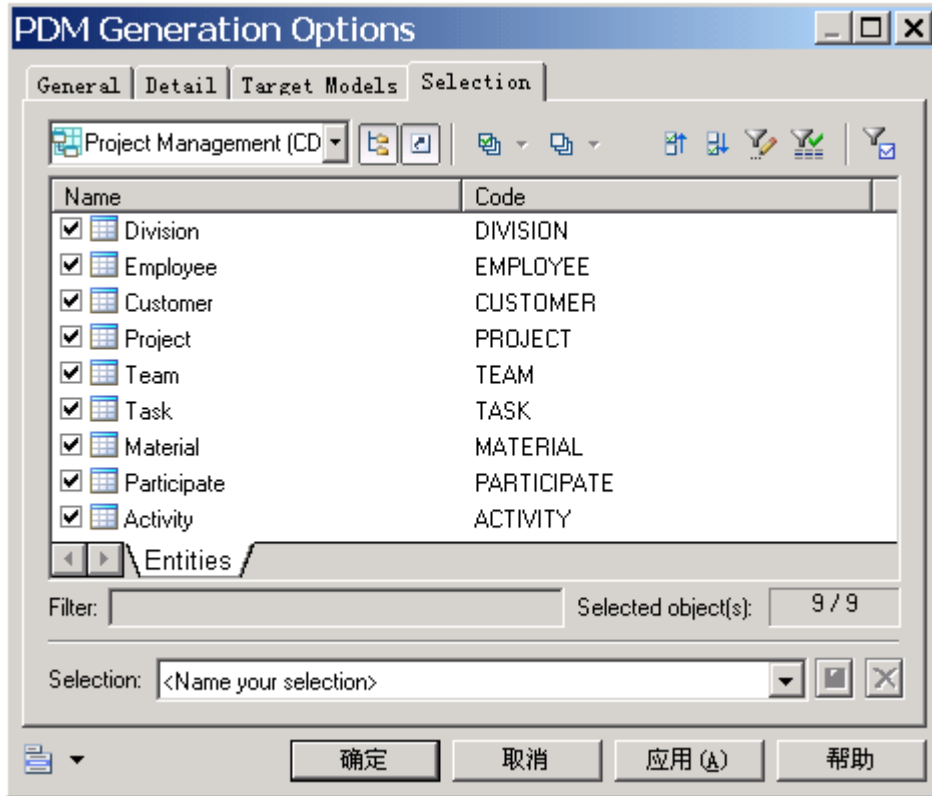
指出使用数据库管理系统库中储存的数据库管理系统定义文件。

- 4 点击detail页面。细节页出现。
- 5 挑选出来的或清除下列各项选项:



如果选择了**Check Model**，模型将会在生成之前被检查。**Save GenerationDependencies** 选项决定PowerDesigner 是否为每个模型的对象保存对象识别标签，这个选项主要用于合并由相同CDM生成的两个PDM。

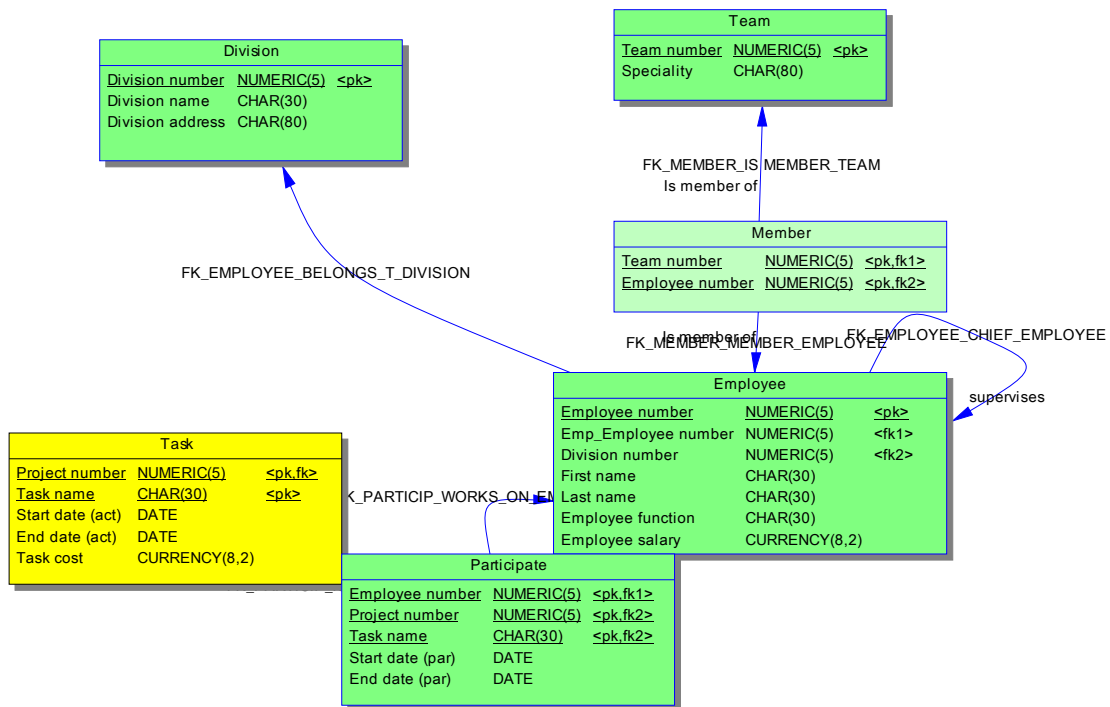
6 选择**Selection** 定位键。它列出 CDM 的所有对象。默认地，所有的对象被选择。



7 点击确定。

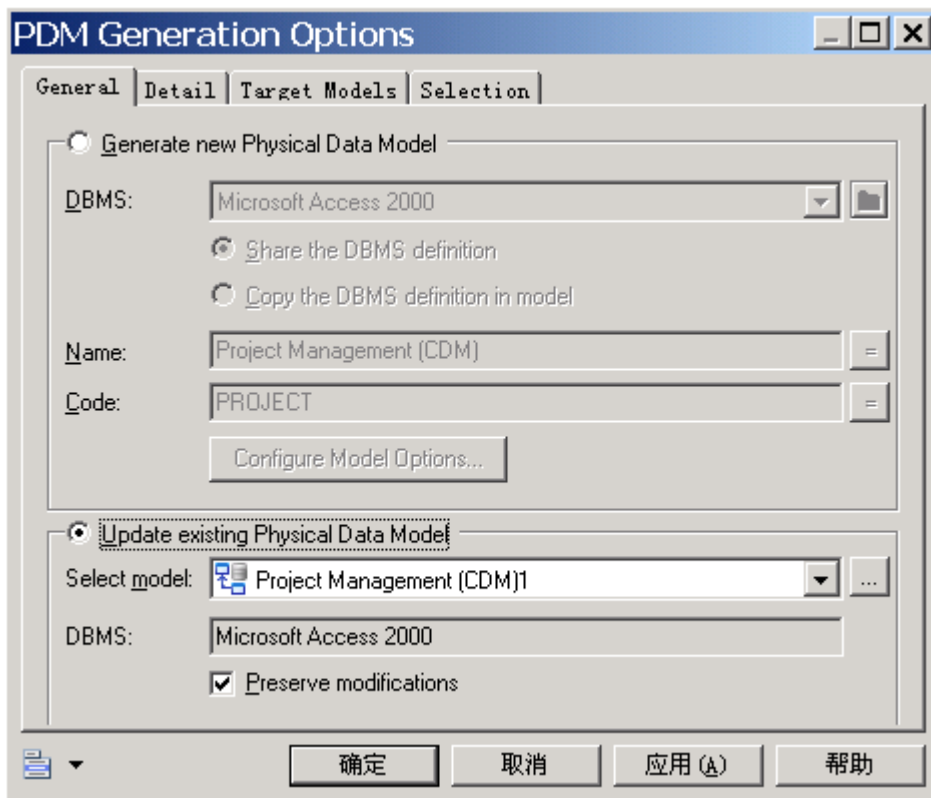
8 当程序已经完成生成时，关闭结果窗口。

PDM 在模型的窗口中出现。



生成PDM后，可能还会对前面的CDM进行更改，若要将所做的更改与所生

成的PDM保持一致，这时可以对已有PDM进行更新。这时操作也很简单，Tools->Generate Physical Data Model，在打开的PDM Generation Options窗口中选择Update existing Physical Data Model，并通过Select model下拉框选择将要更新的PDM。如下图：



最后我们在CDM部分的工作应该就是根据所建立的概念模型生成文档了，文档是作为设计成果的输出，也用于开发小组成员交流的媒介，其重要性不能忽视。

在PDM中生成的键类型取决于CDM中用于定义一个Relationship的基数和依赖类型。

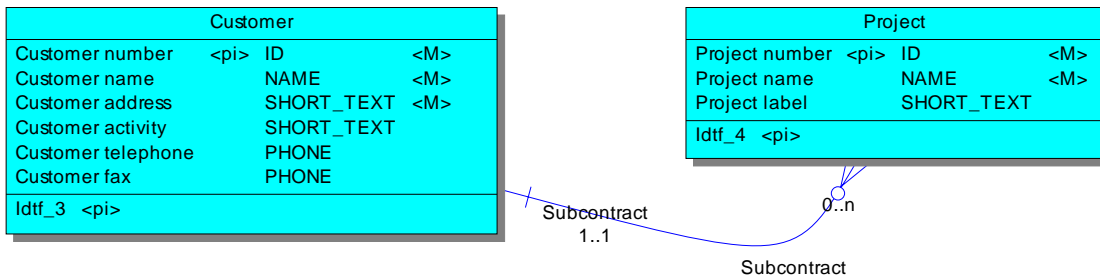
- **非依赖性一对多关系 (Independent one-to-many relationships)**

在非依赖性关系中，“一”端的实体主标识符将转化为：

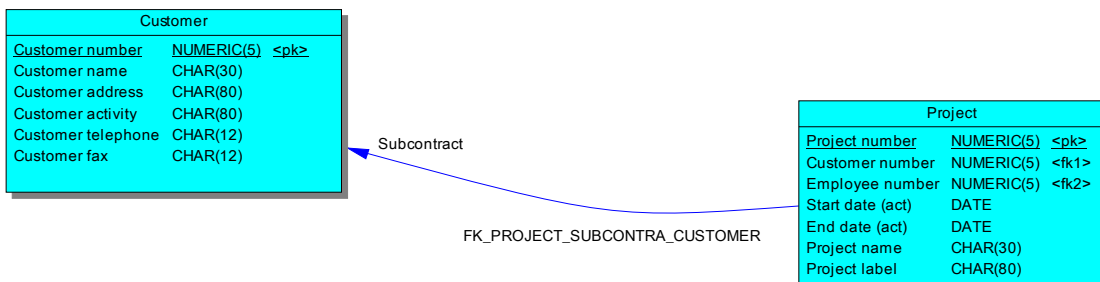
由关系中“一 (one)”端的实体生成的表的主键 (Primary key)

由关系中“多 (many)”端的实体生成的表的外键 (Foreign key)。

如下图所示：



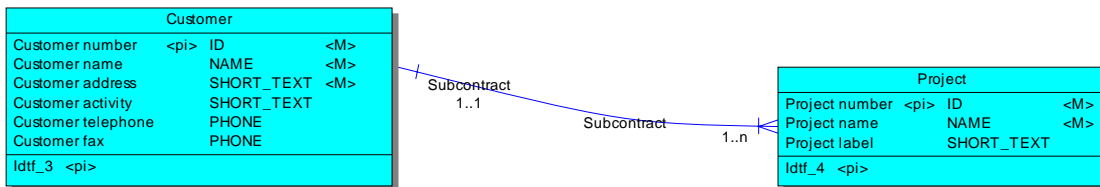
CDM中Independent one-to-many relationship



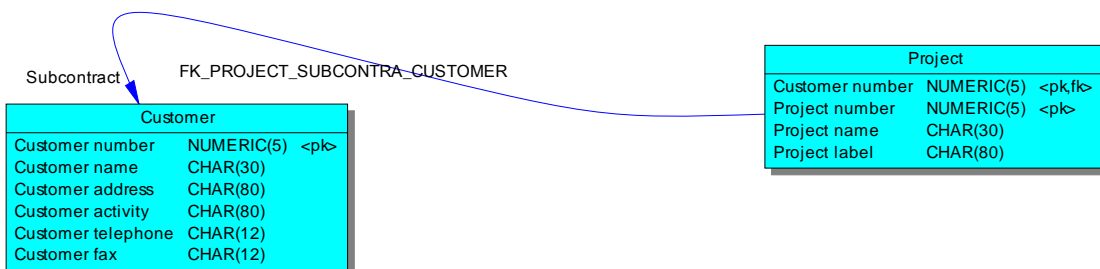
生成的PDM中的Independent one-to-many relationship

● 依赖性一对多关系 (Dependent one-to-many relationships)

在依赖性关系中，被依赖端的主标识符转化为主键，依赖端则产生一个与被依赖端主标识符同名称的字段同时作为依赖端的主键和外键，如果依赖端实体中已经存在主标识符转化为主键，则该键同主键共同组成主键，同时作为外键。



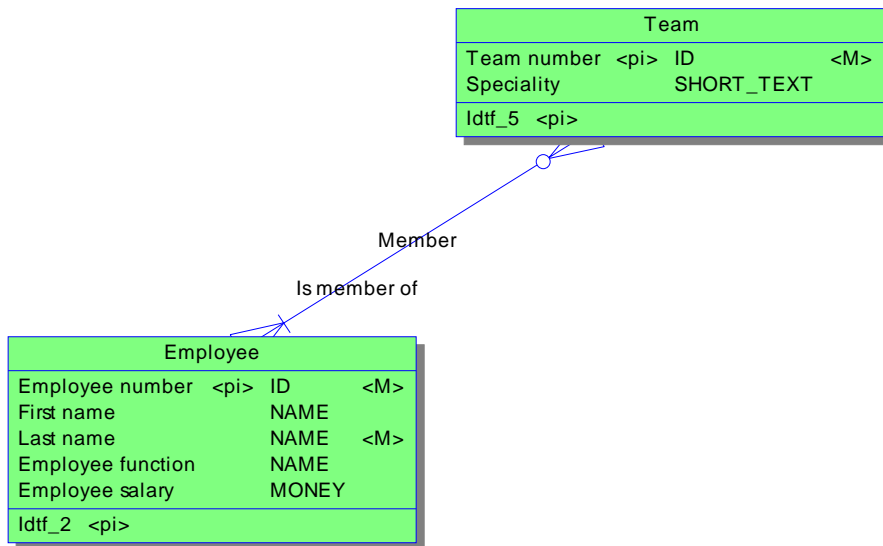
CDM中Dependent one-to-many relationship



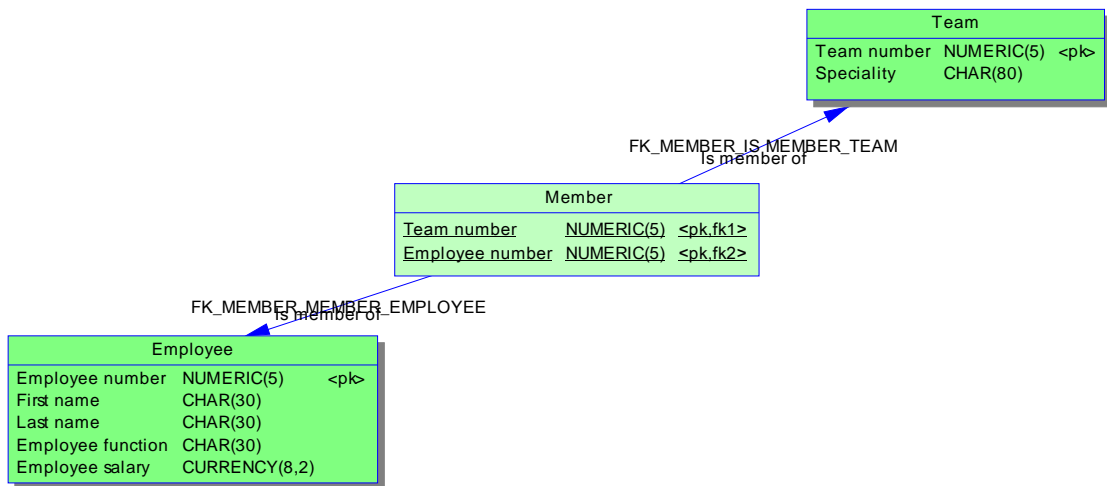
生成的PDM中的Dependent one-to-many relationship

● 非依赖性多对多关系 (Independent many-to-many relationships)

在非依赖性多对多关系中，各实体的主标识符（**Primary key**）迁移至一个新生成的连接表中都作为外键，同时共同组成这个新连接表的主键，各实体的主标识符也转化为其所生成表的主键（**Primary key**）。下图所示CDM，每个雇员可以是一个或多个团队的成员，同时每个团队也可能包含一个或多个的雇员。



CDM中Dependent one-to-many relationship



生成的PDM中的Dependent one-to-many relationship

● **非依赖性一对一关系（Independent one-to-one relationships）**

在非依赖性一对一关系中，如果没有定义支配角色（**Dominant role**）的方向，则各实体的主标识符均自动迁移转化为另一实体生成的表的外键。

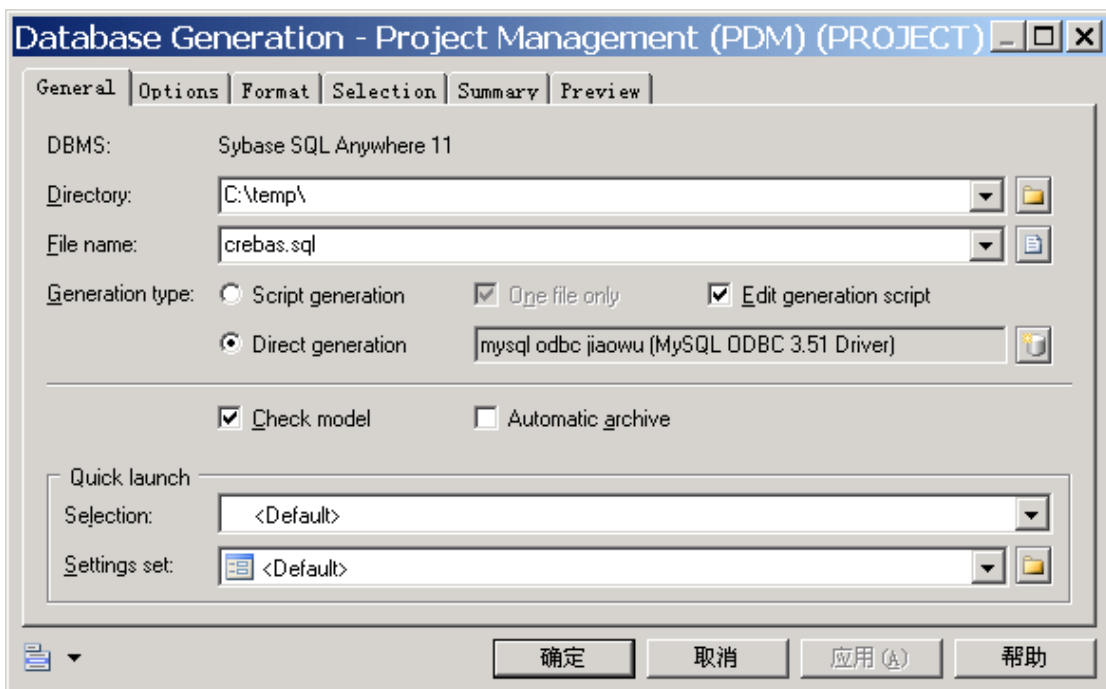
4.3使用物理数据模型创建数据库脚本

可以直接地从PDM产生一个数据库，或产生一个能在特定的数据库管理系统环境中运行的数据库脚本。

默认是生成与PDM相同数据库的脚本，但是也可以产生其他数据库的脚本产生一个数据库创建脚本

1 选择Database—> Generate Database。

数据库生成对话框出现。 它显示生成参数。 默认参数已经被选择。



2 SQL的文件名称框中键入脚本文件名。

3 在目录框中，输入或选择一条路径。

4 选择生成脚本的按钮。

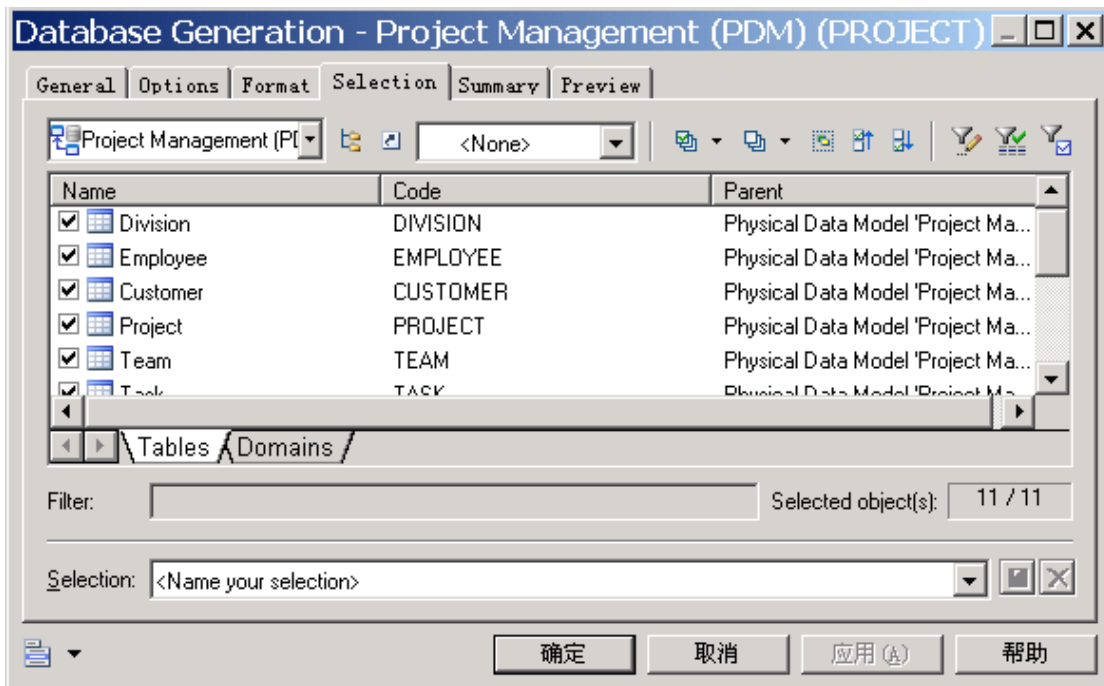
5 选择仅仅生成一个文件。

6 点击Selection 定位键。

7 点击底部表定位键。

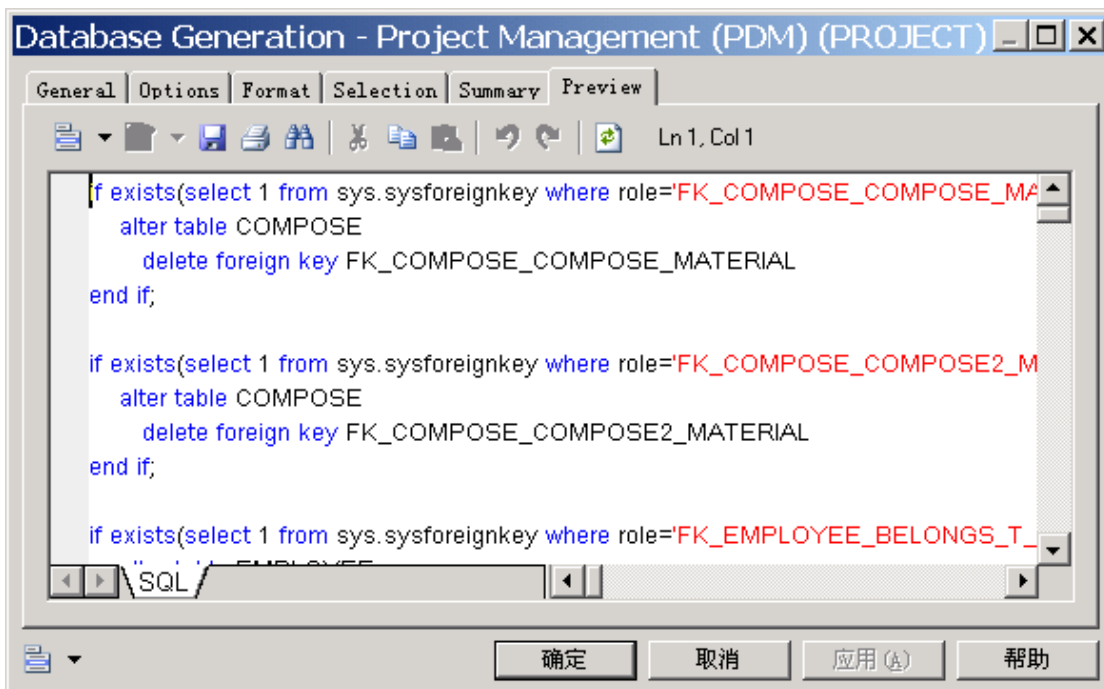
表页列出模型中选择可用的所有数据库表。

8 全部点击选择工具。这选择所有的表复选框。



9 切换Views和Domains选择需要的视图和域。

10 点击Preview, 可预览脚本。



11 点击确定。

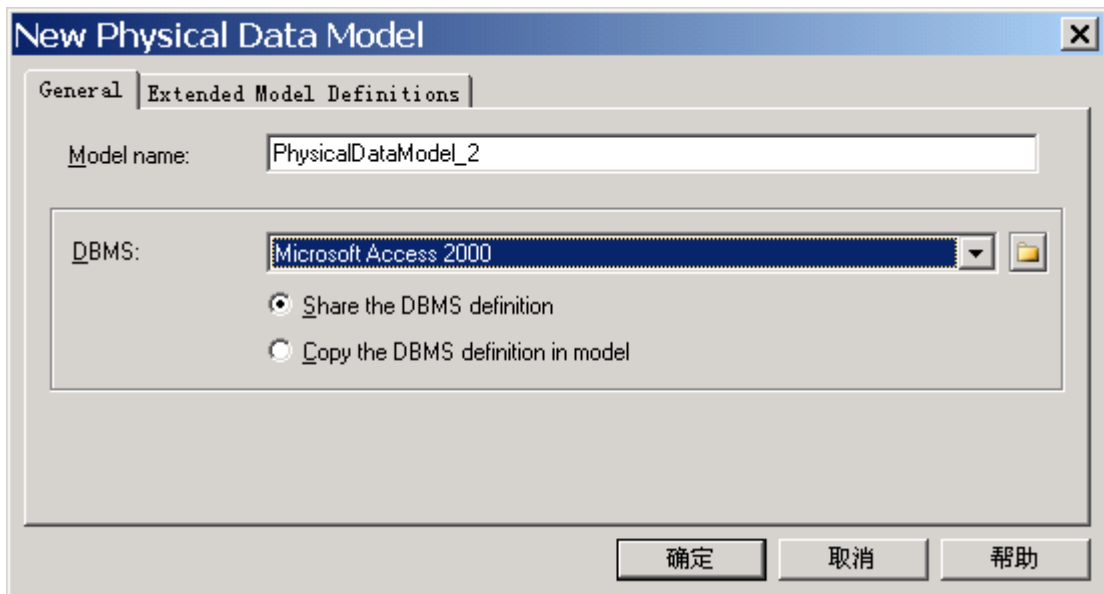
可以生成数据库脚本, 如果选择ODBC方式,则可以直接连接到数据库,从而直接产生数据库表以及其他数据库对象。

4.4 逆向工程

Powerdesigner可以逆向工程已存在的数据库，加进新的PDM之内。数据来源可以是脚本文件或一个ODBC数据源。当你逆向工程使用脚本的时候，可以使用单一脚本文件或多个脚本文件。

- 从脚本文件逆向工程到PDM

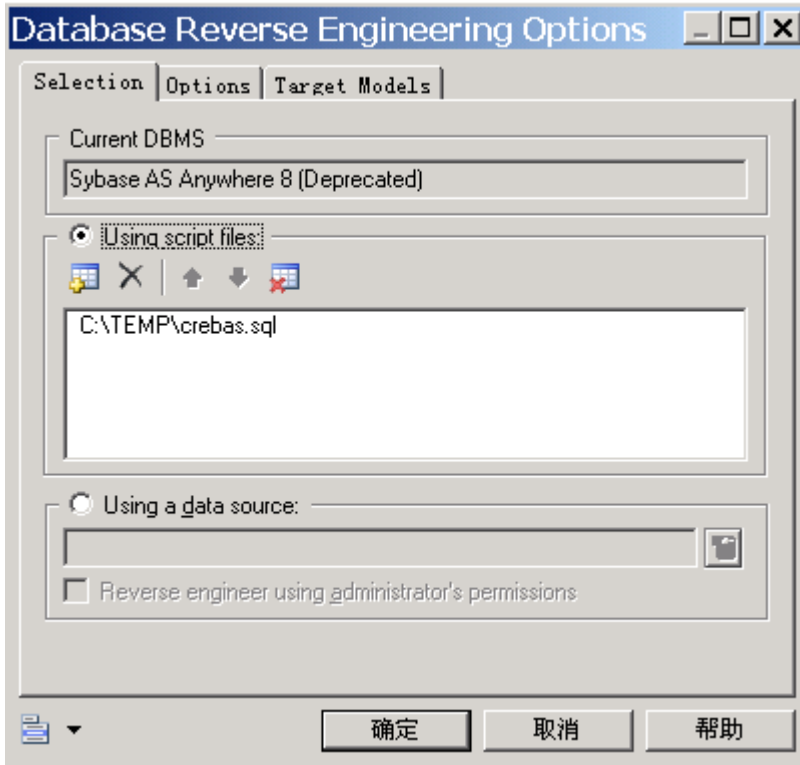
1 选择File->Reverse Engineer->Database 显示新的物理数据模型的对话框。



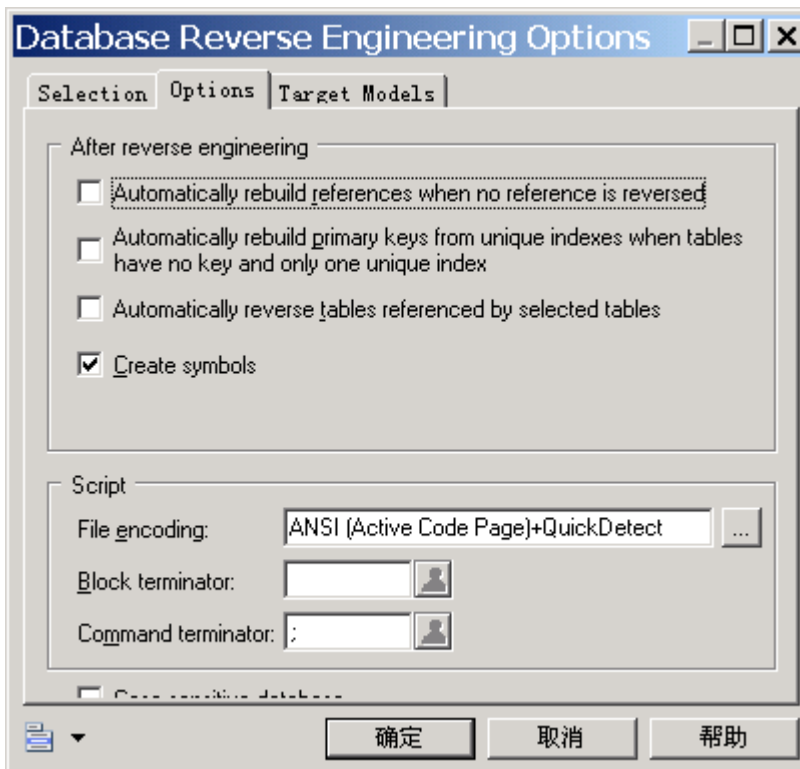
2 输入数据模型名称。

3 选择下拉列表框的一个数据库管理系统。

4 点击确定。数据库逆向工程对话框出现。



- 5 点击使用脚本文件的 Radio 按钮。
- 6 浏览适当的目录选择脚本文件。
- 7 点击Options选项页，进行详细设置。

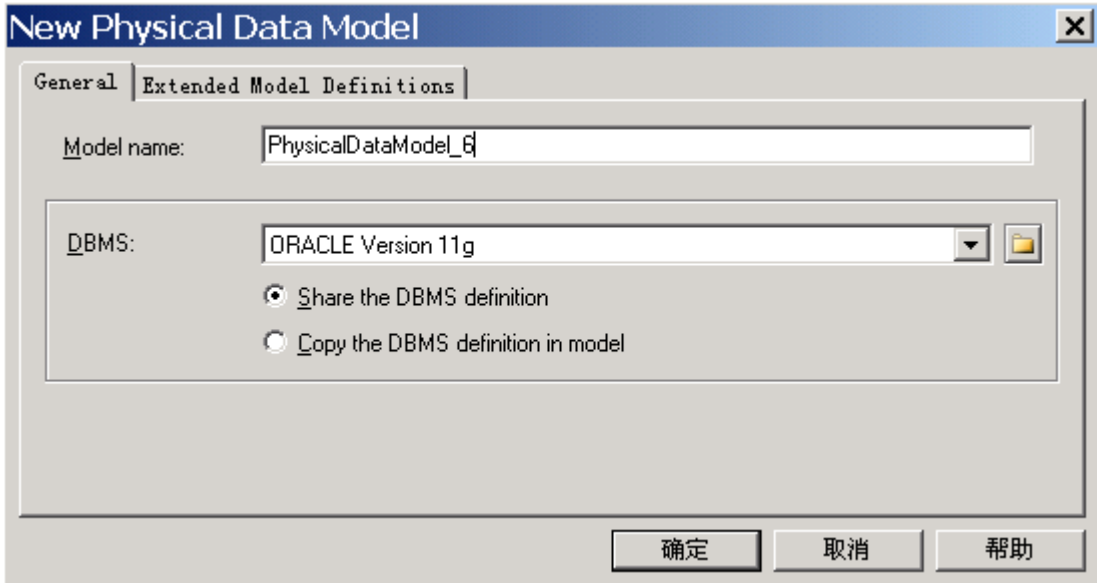


- 8 点击确定。

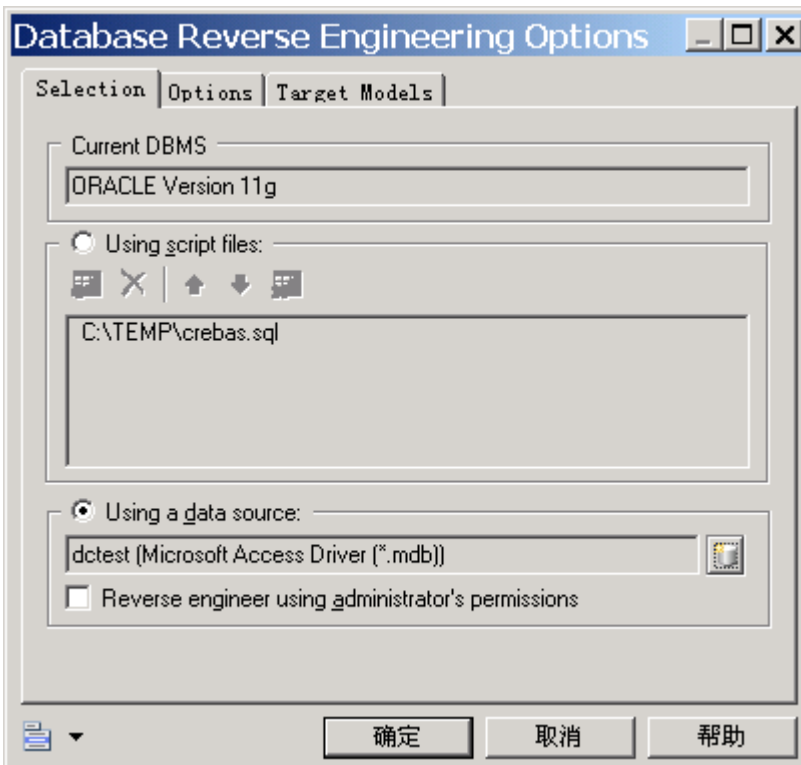
输出窗口的信息指出被指定的文件完全逆向工程。

● 从ODBC逆向工程到PDM:

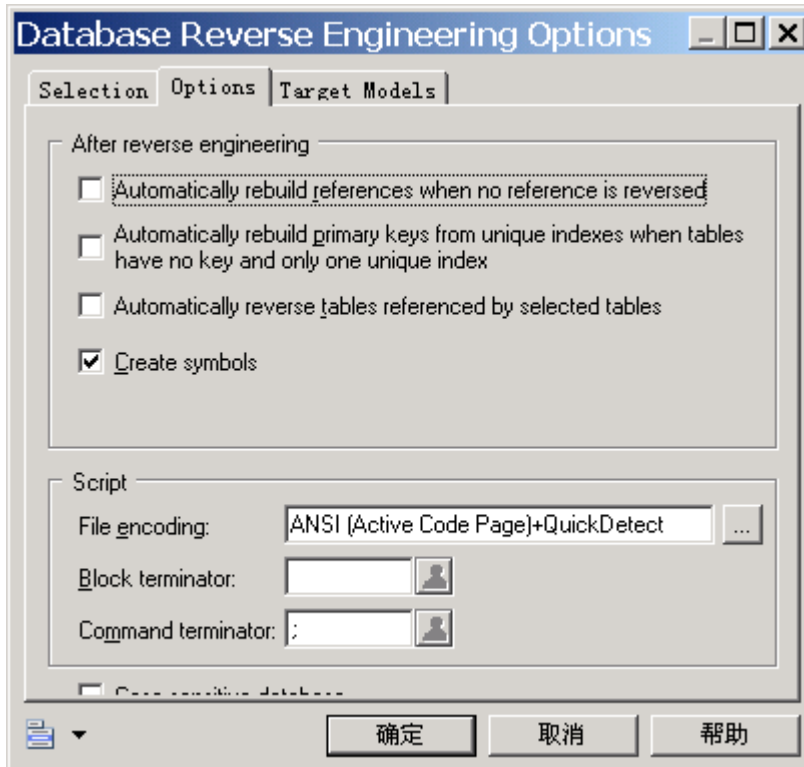
- 1 选择File->Reverse Engineer->Database 显示新的物理数据模型的对话框。
- 2 点击Share 单选框。
- 3 选择下拉列表框的一个数据库管理系统。



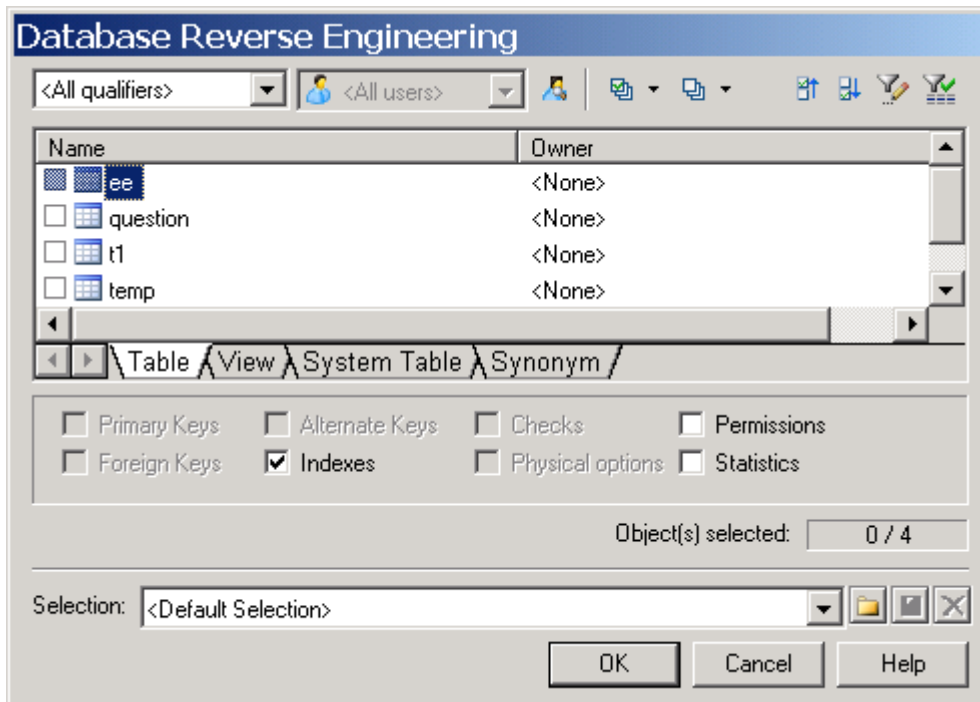
- 4 点击确定。数据库逆向工程对话框出现。



- 5 点击使用一个ODBC 单选框,选择一个ODBC。
- 6 点击选项定位键显示选项页。
- 7 选择逆向工程选项。



- 8 点击确定。ODBC逆向工程对话框出现。



- 9 点击确定。

输出窗口的信息显示哪些表被转换而且指出数据库成功逆向工程。

- **逆向工程进入已存在的 PDM 之内**

你也能逆向工程进已存在的 PDM 之内。 逆向工程的这一个类型依赖现在的数据库管理系统的数据库管理系统定义。

数据来源可以是脚本文件或一个ODBC。 当你逆向工程使用脚本的时候,你能使用一个单一脚本文件或一些脚本文件。

- **合并 PDM**

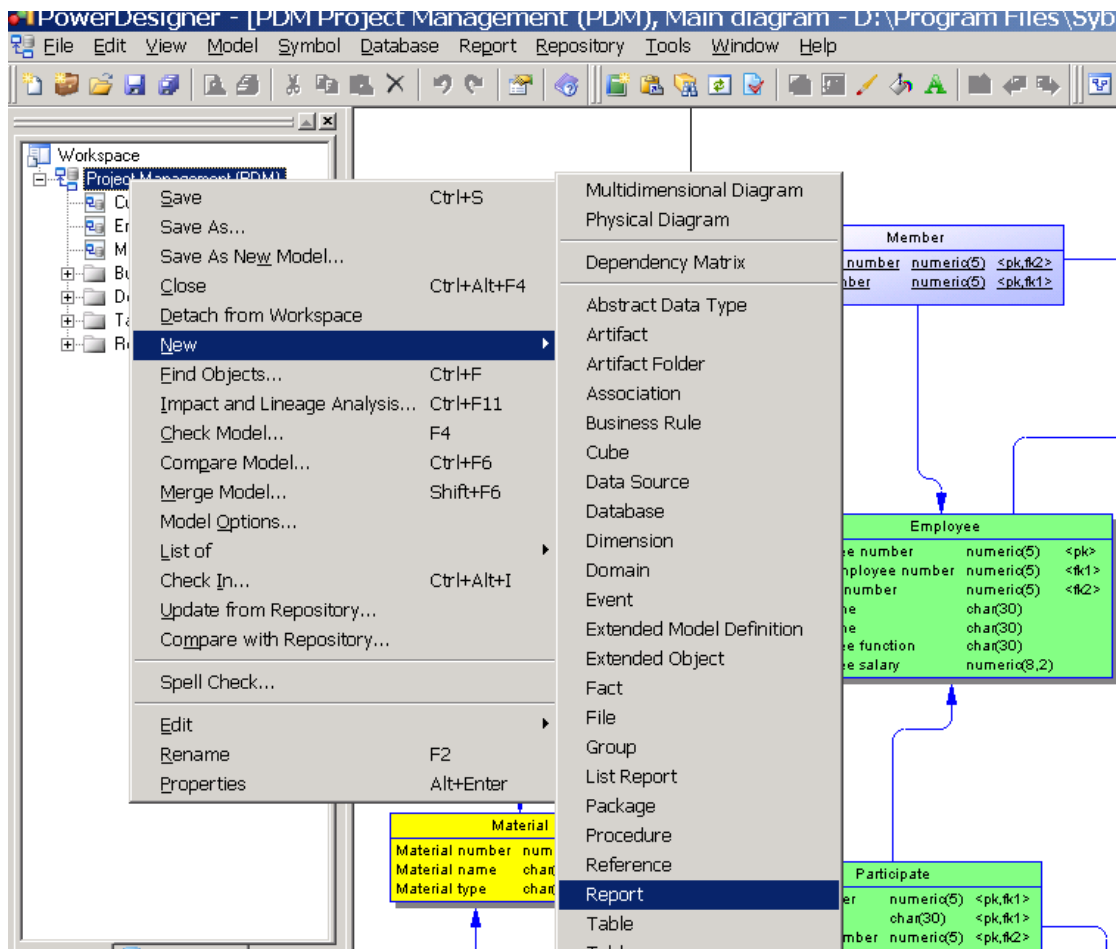
当你逆向工程进已存在的 PDM 之内的时候, 一个模型合并窗口出现逆向工程程序完毕之后。你然后能使用**模型的合并功能**把当前和新的逆向对象整合到一个模型。

5 生成数据库报表文件

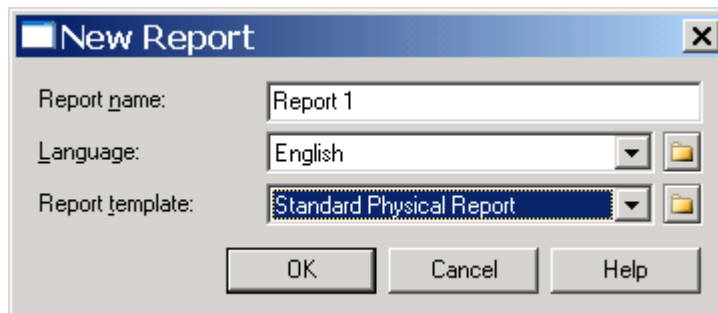
当设计出数据库以后，有一份正规的DOC文件是很必要的，下面讲解如何生成报表。

1.新建立报表文件

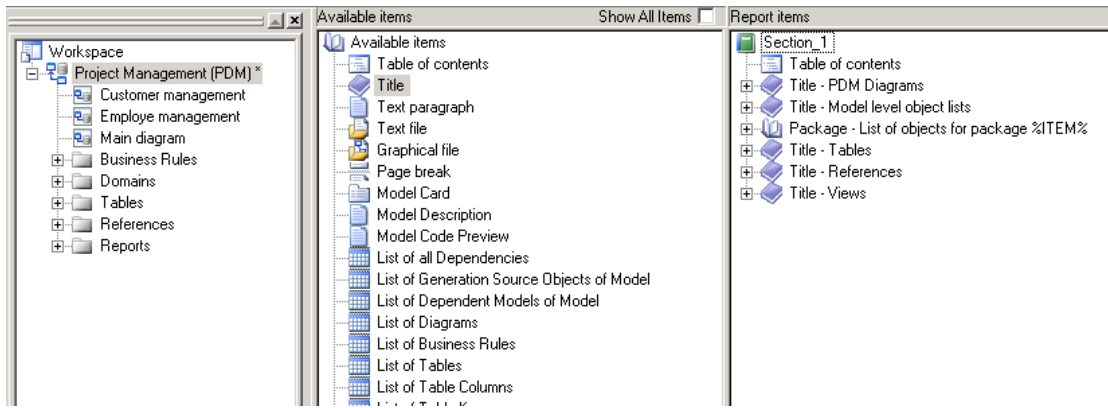
右键单击模型，选择New->Report，弹出如下界面



输入报表名称，选择报表模板。

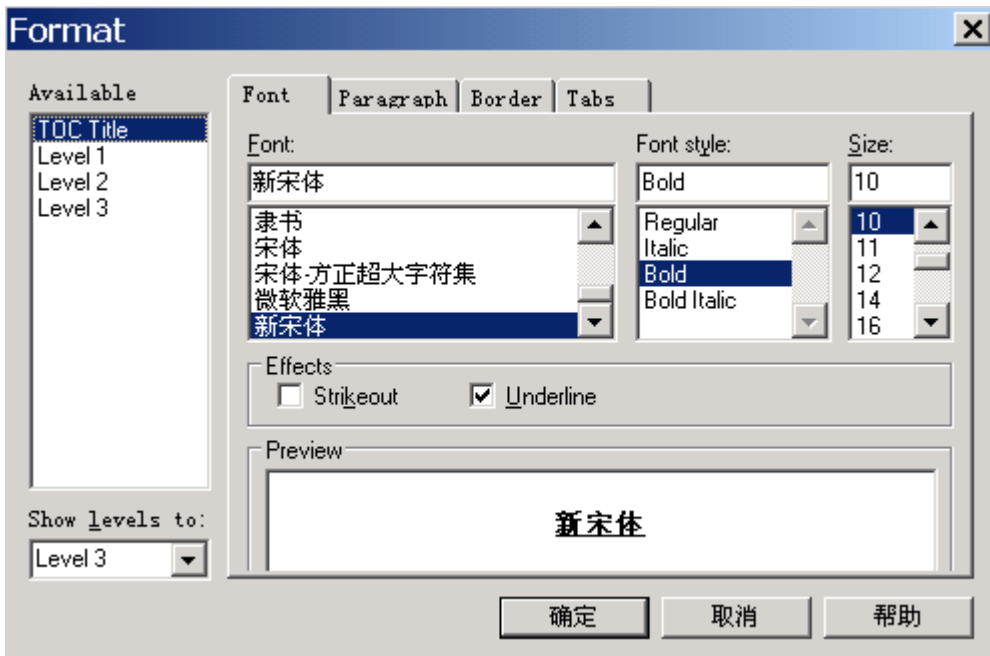
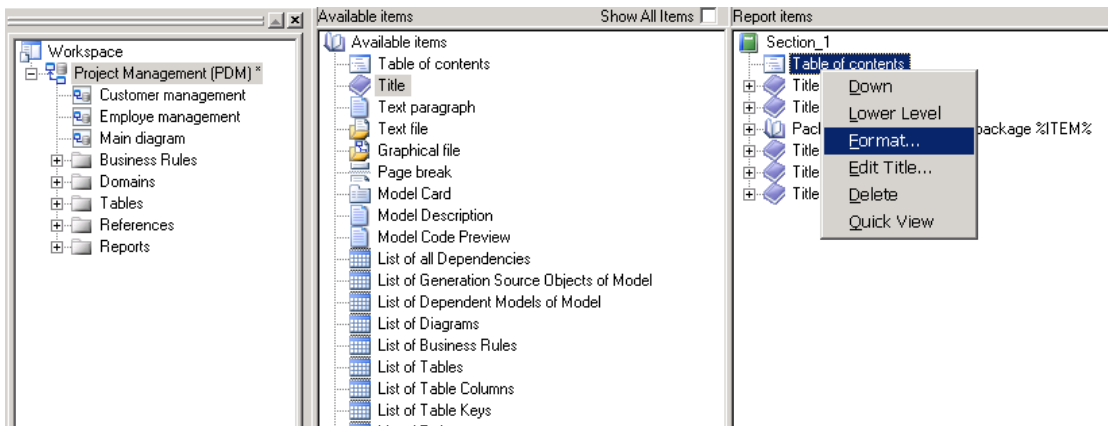


2.拖拉需要显示的部分到报表输出选项

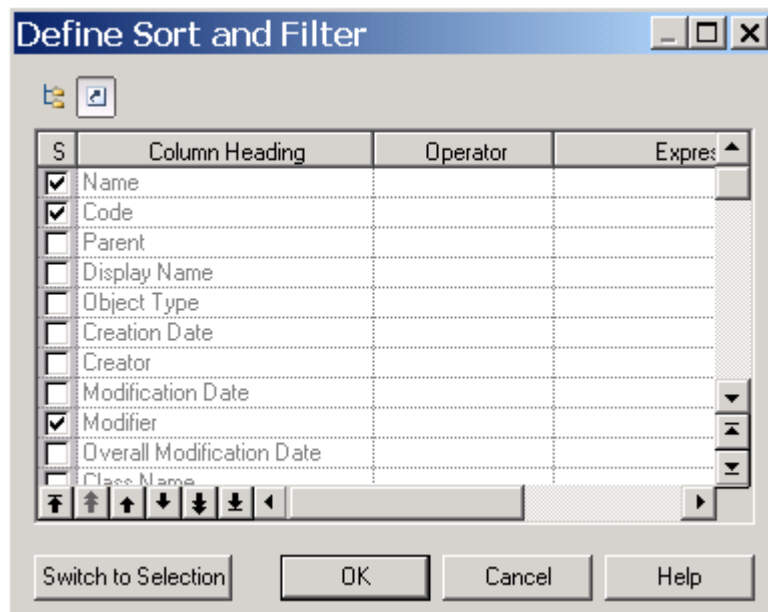
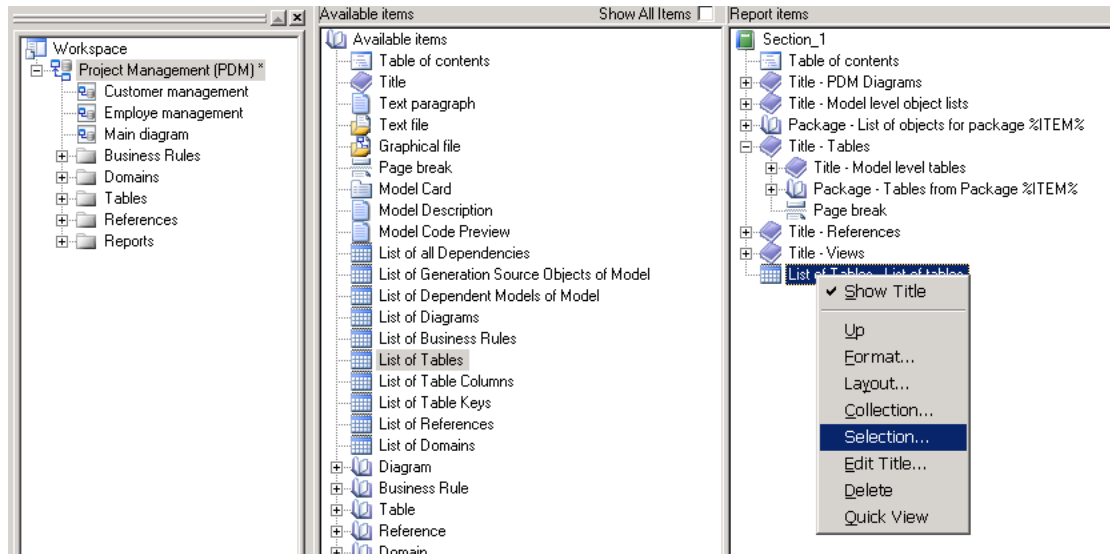


3.修改输出选项的内容

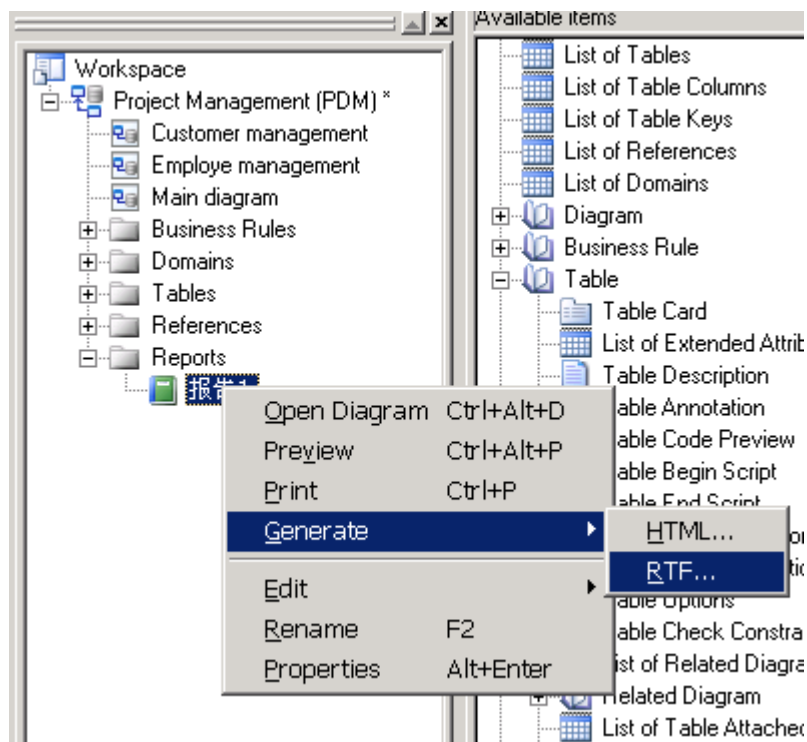
在内容上单击右键，选择Format，可以修改相应项目的标题、字体、格式等。



4.修改列的宽度和需要显示的内容

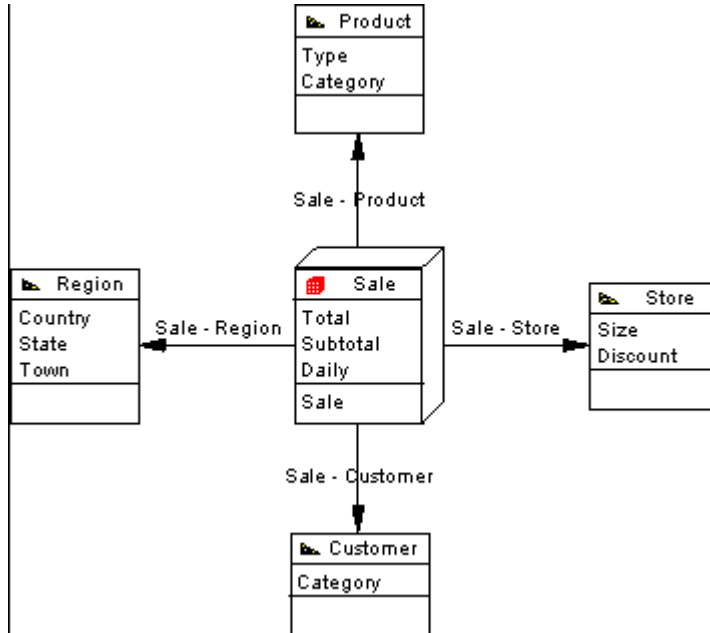


5.生成RTF或者其他的类型的报表文件.



6 多维图

多维数据图和数据集市提供了图形化的数据展现，帮助识别事实表(fact)、多维数据集(cube)以及维度(dimension)。



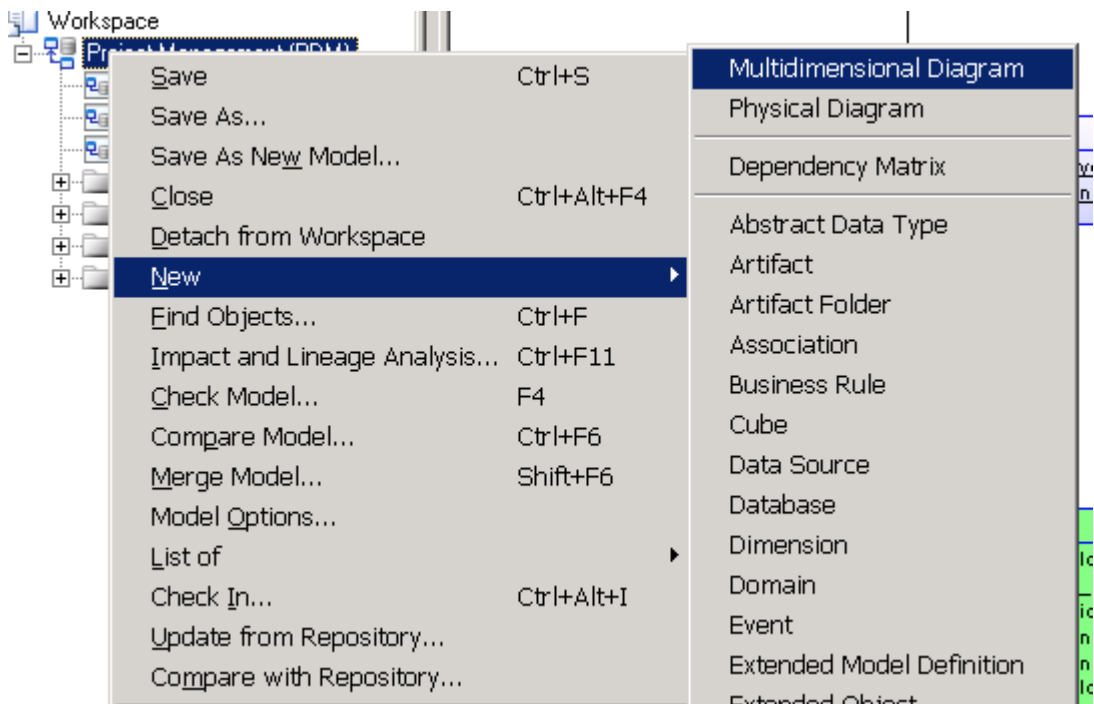
多维图中的对象:

名称	工具图标	图形符号	描述
多维数据集 (Cube)			多维数据集
维度 (Dimension)			观察数据集的角度
属性 (Attribute)			用于细化维度, 如时间维中的年、月、日属性
事实 (Fact)			数据集中的度量组
度量 (Measure)			数值型变量
层次 (Hierarchy)			用于描述维的钻取路径
关联 (Association)			数据集与维之间的关联

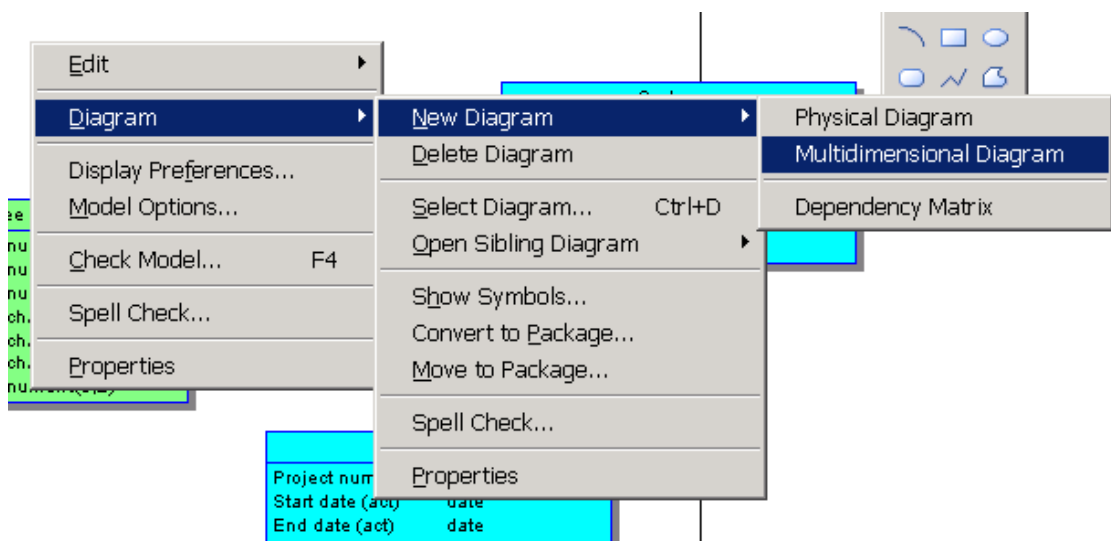
6.1 创建多维图

1. 在已有的PDM中创建多维图的方式如下：

右键单击已有PDM模型，在上下文菜单中选择 **New > Multidimensional Diagram**。

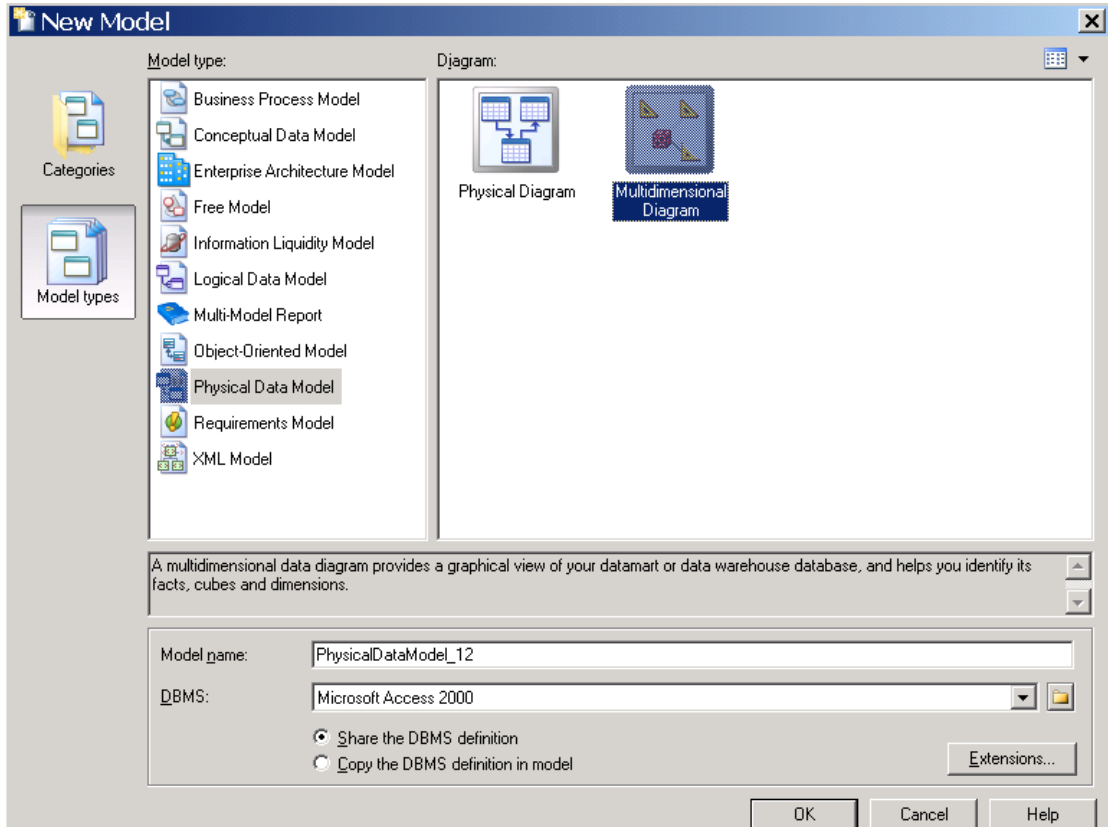


或者右键单击图的背景，在上下文菜单中选择 **New > Multidimensional Diagram**。



2. 新创建一个带多维图的PDM

选择File > New Model，在模型类型列表中选择physical data model，选择Multidimensional Diagram作为初始图形，点击OK。



6.2 获得多维对象

1. 在已有的PDM中：

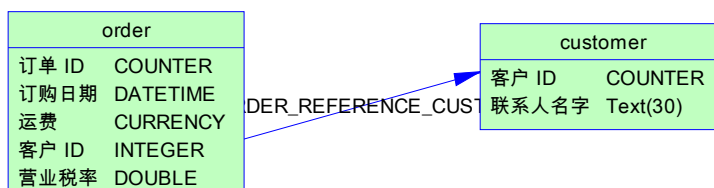
选择Tools > Multidimension > RetrieveMultidimensional objects...

2. 指定获得Facts和Dimensions

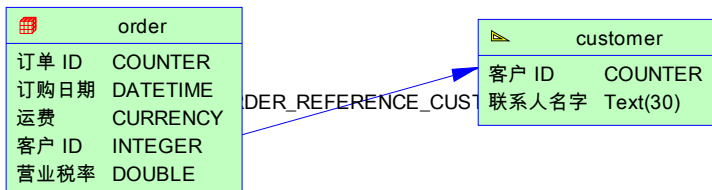
3. [可选] 点击“Selection tab”指定需要获取哪些表

4. 点击“OK”

执行前：



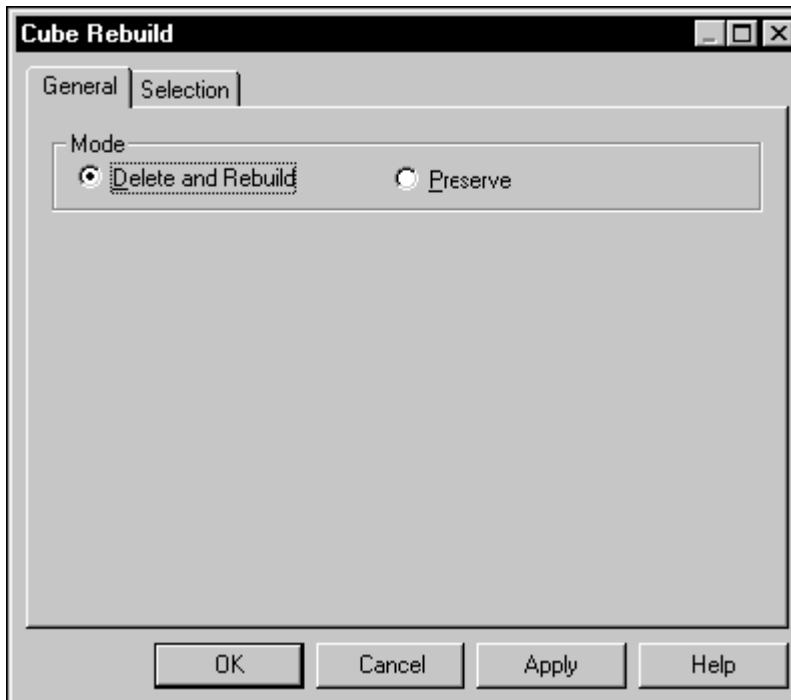
执行后：



6.3 建立Cubes

1. 在已有的PDM中：

选择Tools > Multidimension > RebuildCubes启动向导



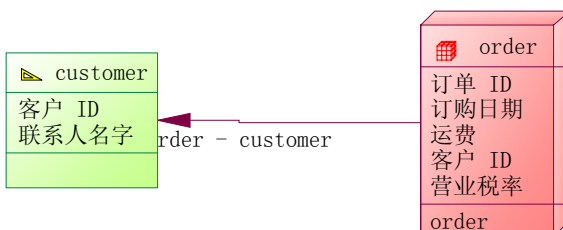
2. 指定重建Cubes的模式。可选如下两种：

- Delete and Rebuild –删除所有已有的Cubes.
- Preserve – 只重建没有手动修改过得的Cubes

3. [可选] 点击“Selection tab”指定哪些表或视图用于重建Cubes

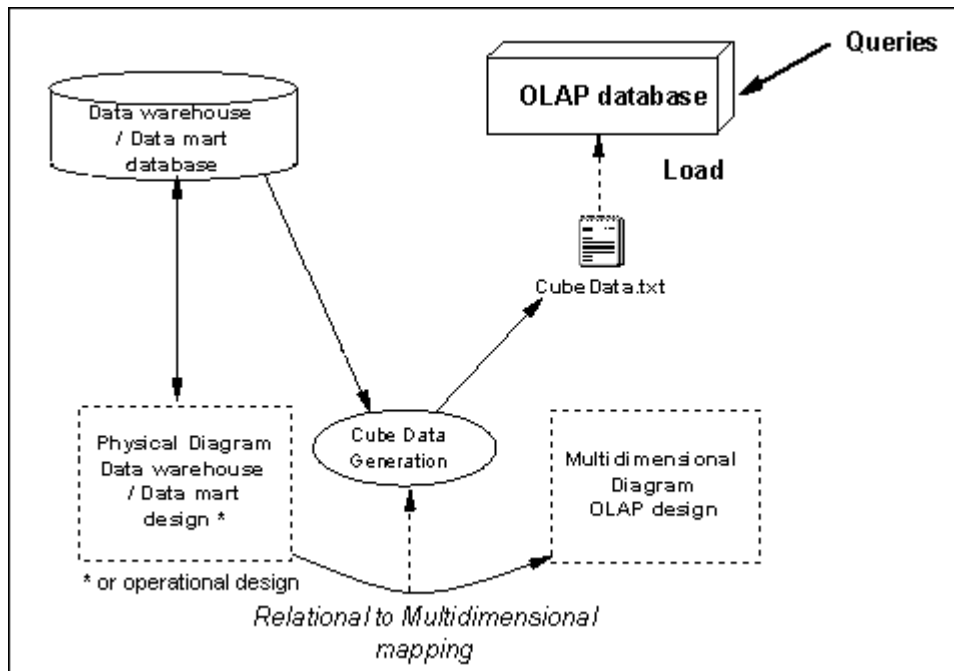
4. 点击“OK”

执行后生成如下多维结构：



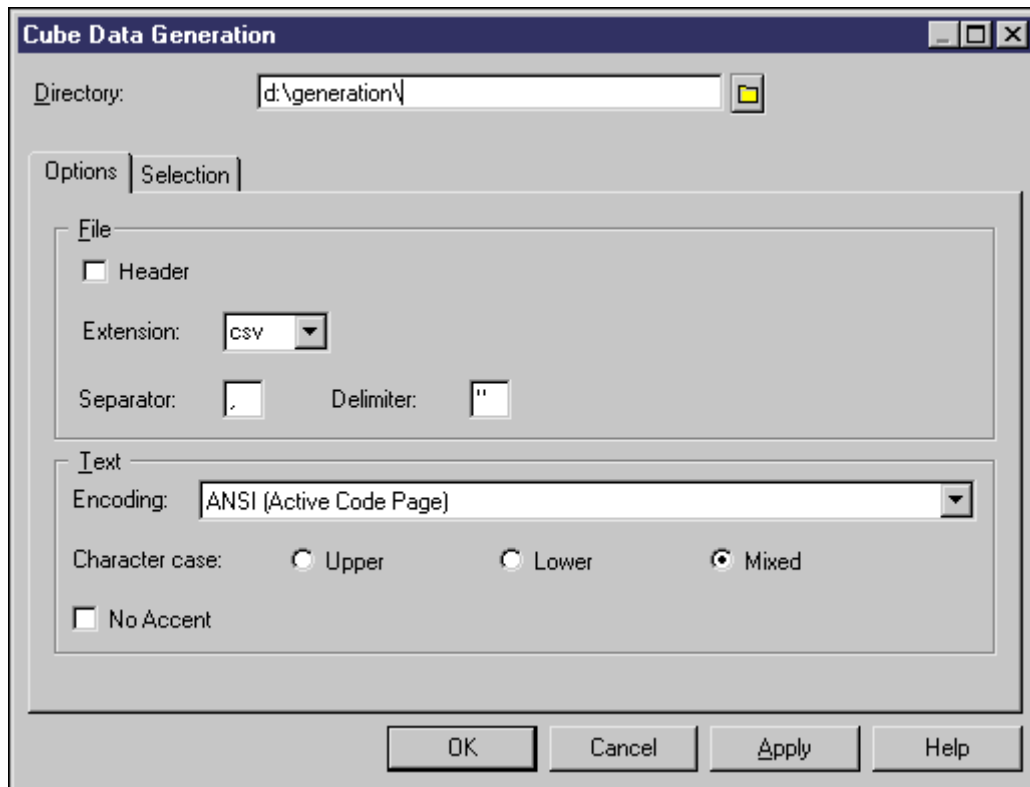
6.4 创建Cube数据

OLAP中的Cube必须被填充数据才能使用，Cube中的数据来自数据仓库、数据集市或关系型数据库。Powerdesigner允许生成OLAP服务器所需的以文本方式存储的Cube。



1.在已有的多维图中：

选择Tools > Generate Cube Data .向导界面如下：



2.定义目标文件存放的路径

3.选择或填写可选参数

4.点击“OK”

生成的Cube数据文件将存放于指定目录下。