

前言

在 CSDN 上转悠经常看到有网友寻求 PowerDesigner 相关资料的帖子，Baidu,Google 上找找还真很少；同时也有不少网友发来 Email 询问相关 PowerDesigner 问题或索要相关资料的，故下定决心制作本文档。折腾二十多天，终于输出了现在的文档，其中绝大部分内容都是依照 PowerDesigner 自带的帮助文档翻译过来，乐意啃英文的朋友最好还是看其“原汁”教程，同时本文档仅用于帮助分析设计人员更快熟悉掌握 PowerDesigner 的使用方法，不包含分析设计方面的理论，所以要作好系统的分析设计工作还是需要用户深厚的项目实践功底。

起初想尽量按照 PowerDesigner 自带帮助文档完整地进行，尝试了一上午的工作之后这种方案马上就被我否决，原因有二：1.内容太多，工作量太多。2.原帮助文档特别周全，个人觉得可以在内容上作很大程度的压缩。姑决定按原帮助文档写，同时加入自己目前正在做的技术论坛分析设计过程以便于理解。

对本文档内容的几点说明：

1. 本文档只包括 PowerDesigner 部分内容（RQM,Report,CDM,PDM），内容不够全面。
2. 内容尽量简略，一些相同或类似操作过程尽量不再重复。
3. 部分术语参考了飞思科技产品研发中心监制电子工业出版社的《PowerDesigner 数据库系统分析设计与应用》。
4. 暂时没有包含 OOM,XML,BPM,ILM 等模型内容，我将会在后期陆续更新。

版本说明：我使用的是 PowerDesigner Trial 11英文版，因此文档中一些菜单，按钮名称也用英文写出（因当心自己译出的名称和中文版上的名称不一致而造成理解不便），若是给使用中文版的朋友带来不便，我在这说声“抱歉”了！同时由于各版本不同部分操作可能会有所区别。

这里要感谢在我进行翻译工作期间给我发送 Email 关注的网友，感谢一直支持我的朋友们！由于第一次做翻译工作，限于水平有限，文档中肯定存在很多不足和错误之处，衷心欢迎各位网友指点迷津，期望得到您的指导！

Email:dingchungao@gmail.com dingchungao@126.com

QQ:330982401

Blog:<http://feiren1421.cnblogs.com>

Slash

2006.8.31

需求模型

PowerDesigner11.0.0.1363评估版

为了更好的将原文含义再现，不加入我个人语言习惯，我尽量按照原文档内容翻译。

环境简介

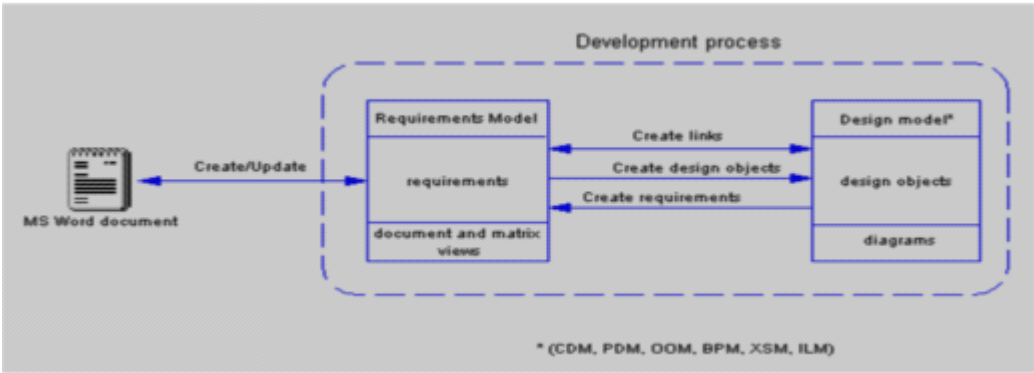
Requirements Model 以下面两种视图呈现（而不是以图表形式）：

- 需求文档视图 对一系列公共属性进行编号
- 可编辑行矩阵 单元格代表了当前需求与设计对象，外部文件或其它需求的联系

Requirements Model 允许你可以：

- 对一结构化技术文档建立需求模型
- 检查现有或引入的模型
- 对需求 and 设计对象(其它类型模型)建立联系
- 对其它设计对象建立需求模型，或反之通过需求模型建立其它设计类型
- 从需求模型生成或更新 MS Word 文档,提供用户一符合需求模型的 MS Word 文档
- 从现有 MS Word 文档生成或更新相应的需求模型

各对象之间关系如下图所示：

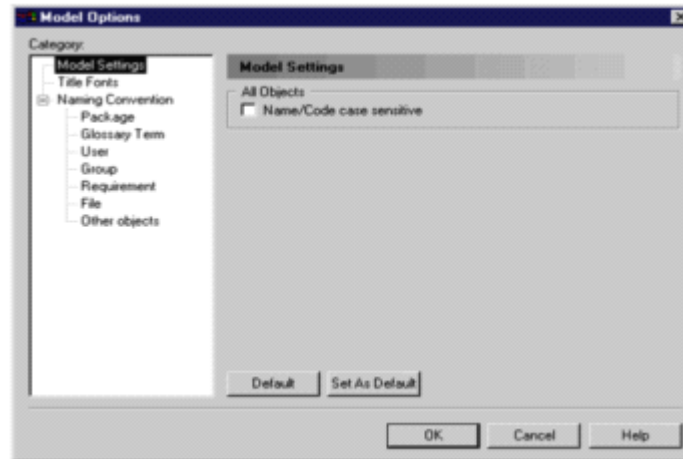


Requirements Model 应该包括如下特定对象(Object)：

Object	Description
Requirement	功能行为的名称或内容，可以是父级或子级需求的一部分，它应该在被指派给用户或群(Groups)前被准确定义说明
Glossary term	用于需求模型中的词汇，它应该被正确定义说明以避免误解，建立一定的通用规则
User	至少与某一需求有关的个人实体
Group	至少与某一需求有关的用户(user)群体

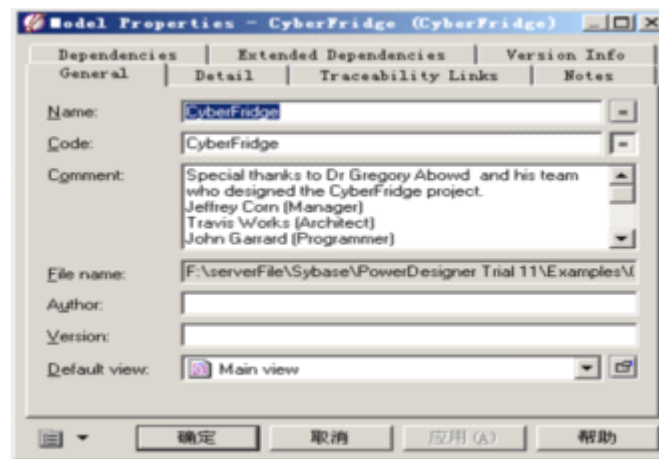
由于 Requirements Model 中没有图表，以上各对象均没有与之对应的图象符号。需求是以图表视图形式列出，可编辑矩阵视图显示出需求和各设计对象，外部文件或其它需求之间的联系。

需求建模环境包括一系列定义不同模型内容和行为的参数和设定选项，你可以通过在建立模型时，使用默认选项建立模型后或建立模型模版时进行设置。
菜单栏—>选择”Tools”—>Model Options，可见以下模型选项对话框，现在可以进行你喜欢的设置了。



定义模型属性

在打开相应模型文件后，选择菜单栏中 Model->Model Properties,或在左边树性对象浏览器中选中对应模型，双击/右键->选择 Properties，均可进入 Properties 设置区间，如下图：



接下来就可以进行你想要的设置了！

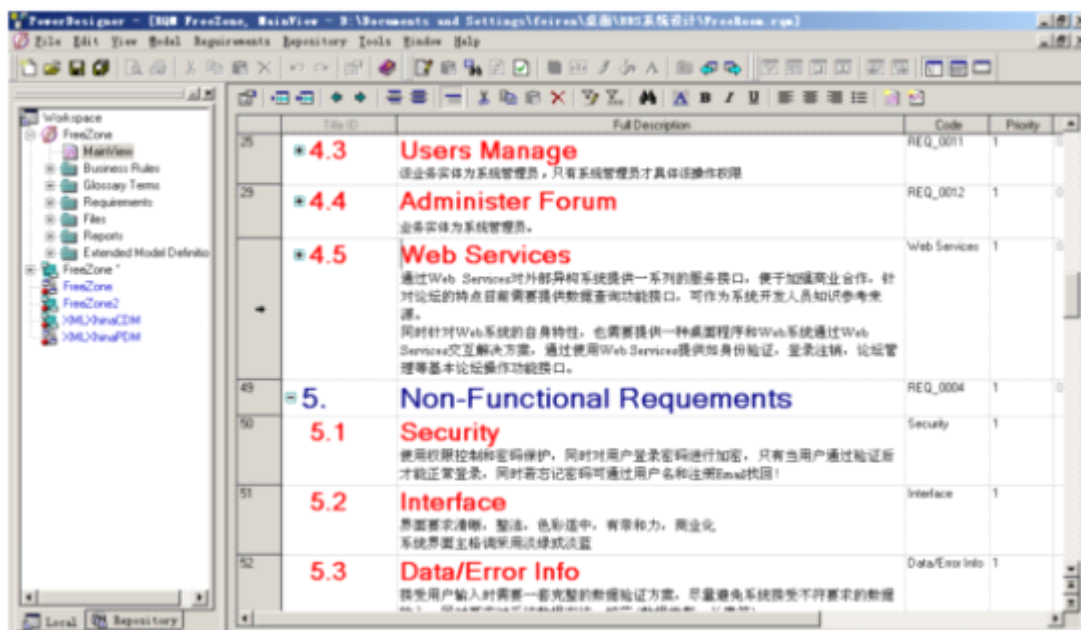
新建 Requirements Model

下面以我自己最近的项目过程为例逐步讲解各过程：

项目简介：这是个类似动网或 CSDN 的论坛系统，参考了它们的功能设计，主要用于本人练习 N 层架构的学习。

建立需求模型：

建立完成的需求视图

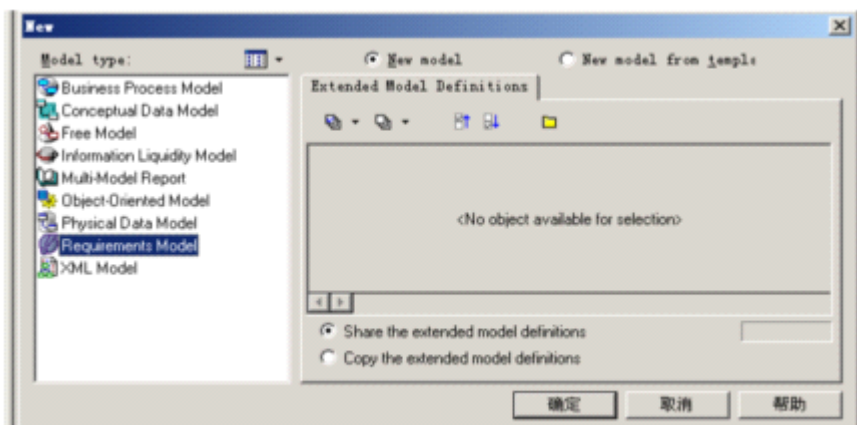


首先我们要新建一 Workspace 作为整个系统各种模型，文档与报告信息集合。

启动 PowerDesigner,这时会默认打开一个 Workspace,单击鼠标右键->选择”Close”,这样我们完成了关闭原来 Workspace,同时新建 Workspace 的工作。接下来就是在其中添加各种模型了。

新建 Requirements Model

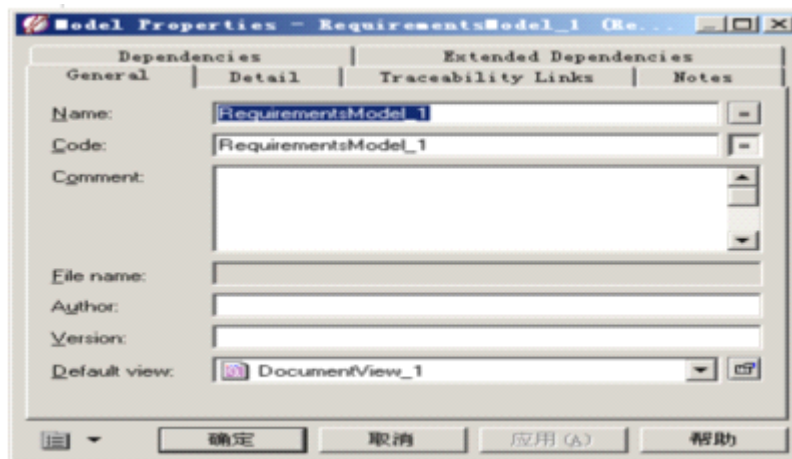
点击 File->New 或鼠标右键单击 Workspace->New->Requirement Model 可以看到新建模型属性选项框如下：



选择左边 Requirements Model,其它为默认设置，确定,OK！

下面我们对新建的 RQM 进行先进行一些基本属性设置：

在资源浏览窗口中右键单击新建好的 RQM->Properties 或直接双击对应 RQM,直接进去模型属性设置 Model Properties,如下图所示：



现在你可以进行自己想要设置了。这里我们将 Name,Comment 分别进行基本设置,同时系统默认 Name 和 Code 是一致的, Name 用来进行分析描述,为了形象明了可以使用中文,而 Code 则和后期的具体设计有关,如用于编码设计,一般多用英文加数字等标准命名(仅供参考)。

同时我们可以看到在新建 RQM 时也自动建立了一个模型视图(View),接下来我们就要对该视图(View)进行编辑以建立需求模型,根据前面需求模型简介介绍的相关 RQM 视图知识,需求模型可以用文档视图的形式表示,后续的大部分工作只有对 View 进行编辑就 OK 了! 先看看完成后的需求视图吧!

	Title ID	Full Description	Code	Priority	
1.		Purpose 该文档为FreeRoom(自由空间)论坛的系统需求说明书,主要对该系统的功能需求做详细描述,同时也作为后述开发的主要参考文档之一。	REQ_0001	1	
2.		Introduction FreeRoom主要目的是为了创建一个自由,轻松的网上技术交流社区,方便志同道合的朋友进行技术交流,讨论。该系统的设计也是用于学习构建多层架构系统,学习应用WebServices应用。	REQ_0002	1	
3.		Terminology	REQ_0006	1	
4.		Functional Requements	REQ_0003	1	0
4.1		Use of Forum	REQ_0009	1	0
4.1.1		User	REQ_0021	1	0
4.1.1.1		Number of Users-online 查看在线用户数量信息,对所有用户均适应	REQ_0017	1	
4.1.1.2		Look/Edit users' self info 查看/编辑已经登录用户的个人信息,只有已经注册且成功登录用户才具有此功能权限	REQ_0016	1	
4.1.1.3		Look the Info of Users 查看已注册用户个人信息,只有已经登录用户才能查看	REQ_0015	1	
4.1.1.4		User Login/Logout 根据登录与否限制用户权限,登录注销功能实现两种权限的切换!	REQ_0014	1	

这里的各系统需求是按层次排列的,这样也使需求文档视图能和标准的层次化 Word/rtf 文档能进行相互转换。可以通过视图上方的工具栏进行全面的需求模型建设。

添加需求(Requirement):

点击需求文档视图工具栏上”Insert a Row”工具或点击需求文档视图的空白区

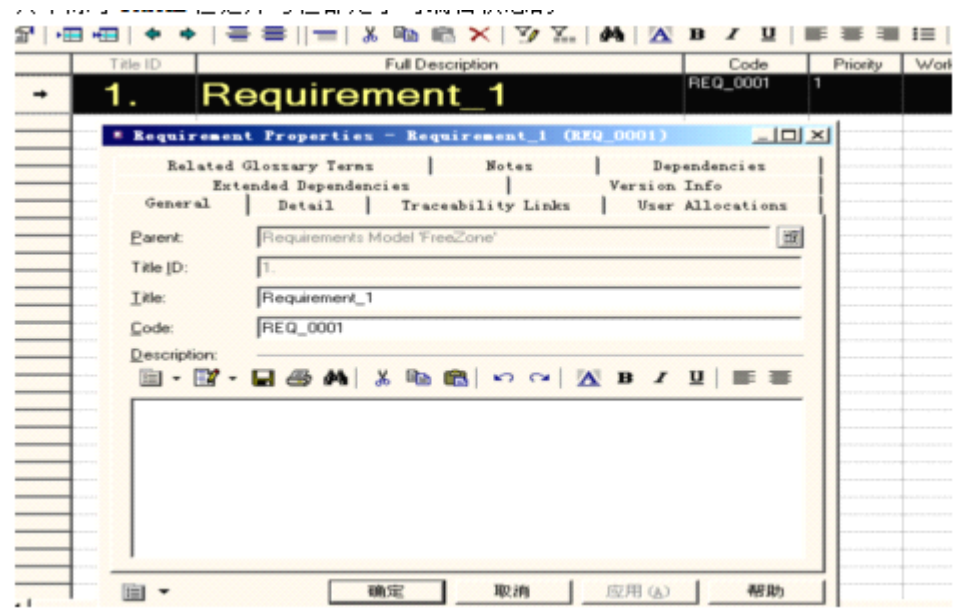
这样一个预先默认自定义的需求已经添加在文档视图中,如下所示:

	Title ID	Full Description	Code	Priority	Workload	Risk	Status
1.		Requirement_1	REQ_0001	1		Undefined	Draft

编辑需求属性

双击需求 TitleID 左边的箭头(arrow)或单击需求文档视图工具栏最左边的 Properties 工具即进入属性属性编辑。

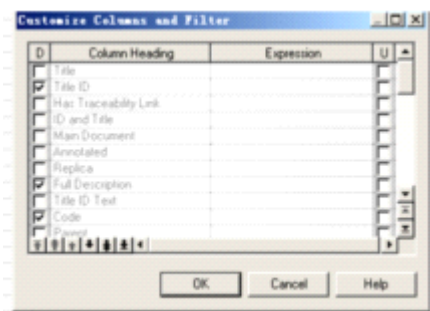
其中除了 TitleID 栏之外每栏都处于可编辑状态的。



注：箭头所在行为选中行

属性各栏目对应着文档视图中的各可编辑栏。这里我们可以设置各需求的详细内容和描述信息，比如标题(Title),需求描述(Description),优先级(Priority),风险(Risk),状态(Status),工作量(Workload)等详细内容。详细设置信息请参考示例文件。

若要更改文档视图中的可见栏目，可以通过单击需求文档视图工具栏中 Customize Columns and Filter 工具，进入



现在可以选择您想要显示的栏目了。

这样我们就基本上完成了系统需求的设计过程，依此多次操作完成如下系统需求文档视图基本框架：

Title ID	Full Description	Code	Priority	Workload	Risk	Status
1.	Purpose 该文档为FreeRoom(自由空间)论坛的系统需求说明书,主要对该系统的功能需求详细描述,同时也作为后续开发的主要参考文档之一。	REQ_0001	1		Undefined	Draft
2.	Introduction FreeRoom主要目的是为了创建一个自由、轻松的网上技术交流社区,方便志同道合的朋友进行技术交流、讨论。该系统的设计也是属于学习构建多层架构系统,学习应用WebServices应用。	REQ_0002	1		Undefined	Draft
3.	Terminology	REQ_0003	1		Undefined	Draft
4.	Functional Requeements	REQ_0004	1		Undefined	Draft
5.	Non-Functional Requeements	REQ_0005	1		Undefined	Draft
6.	References	REQ_0006	1		Undefined	Draft

后面的工作就是对其中各 Requirement 做进一步的细化,对各需求模块做更为细致的划分,即分层细化,这样也和层次化的文档吻合。这里我们以对 Functional Requeements 的设计为例进行讲解,先看看细化完成后的需求文档视图(部分):

4.	Functional Requeements	REQ_0003	1	0
5.	4.1 Use of Forum	REQ_0009	1	0
6.	4.1.1 User	REQ_0021	1	0
7.	4.1.1.1 Number of Users-online 查看在线用户数量信息,对所有用户均适用	REQ_0017	1	
8.	4.1.1.2 Look/Edit users' self info 查看/编辑已经登录用户的个人信息,只有已经注册且成功登录用户才具有此功能权限	REQ_0016	1	
9.	4.1.1.3 Look the Info of Users 查看已注册用户的个人信息,只有已经登录用户才能查看	REQ_0015	1	
10.	4.1.1.4 User Login/Logout 根据登录与否限制用户权限,登录注销功能实现两种权限的切换!	REQ_0014	1	
11.	4.1.1.5 User Register 用户注册功能,提供注册用户的访问权限	REQ_0013	1	
12.	4.1.2 Post	REQ_0022	1	0
18.	4.2 Posts Manage	REQ_0010	1	0
19.	4.2.1 Check-out post Compulsively 强制回帖:权限只对论坛版主开发,且相应论坛版主只对相应论坛的帖子具有该操作权限,版主根据自己的主观判断确定是否应该执行强制回帖,如:长期没有回复,帖子内容有违论坛原则等。对于分数分配,若除发帖人之外没有其它用户回	REQ_0027	1	

现在让我们开始吧!

方法一:

需求文档视图,选中 Functional Requirements->点击视图工具栏"Insert Sub-Object"工具(而不是"Insert a Row"工具),这样就在 Functional Requirements 中插入了一个子对象。

方法二:

于左边资源管理窗口 Requirements 目录下右键单击相应需求名称->New->Requirement 即可。如下图:



现在只要对新插入的子对象进行详细的内容编辑设计即可，同样地我们也可以对各子对象通过再次添加子对象作进一步的细化工作。

如果要提升或降低某部分的需求层次，则可以通过工具栏中的 Promote 和 Demote 来实现调整。

定义 Users 和 Groups

Users (用户) 指在一个需求模型中至少和一个已定义需求有关的人的集合。

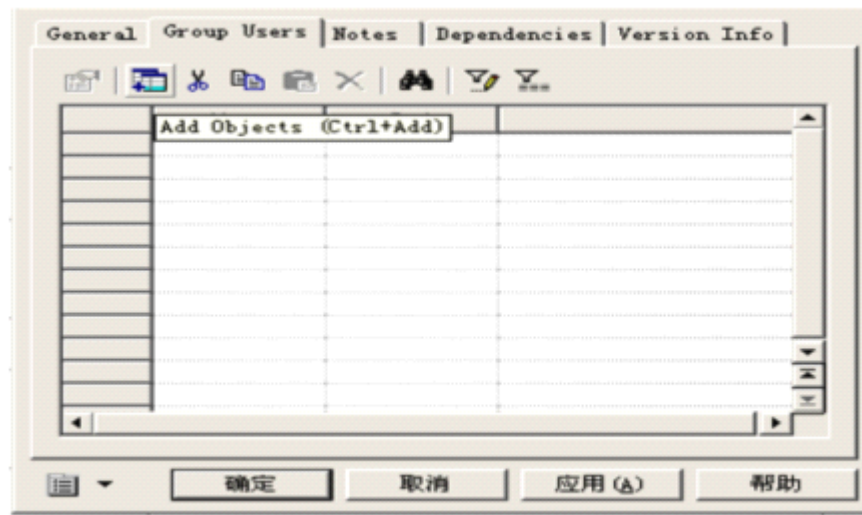
Groups (组) 指专属于开发进程中一个或多个方面的用户类别。每个用户组要与需求模型中至少一个已定义需求有关。

新建 User/Group

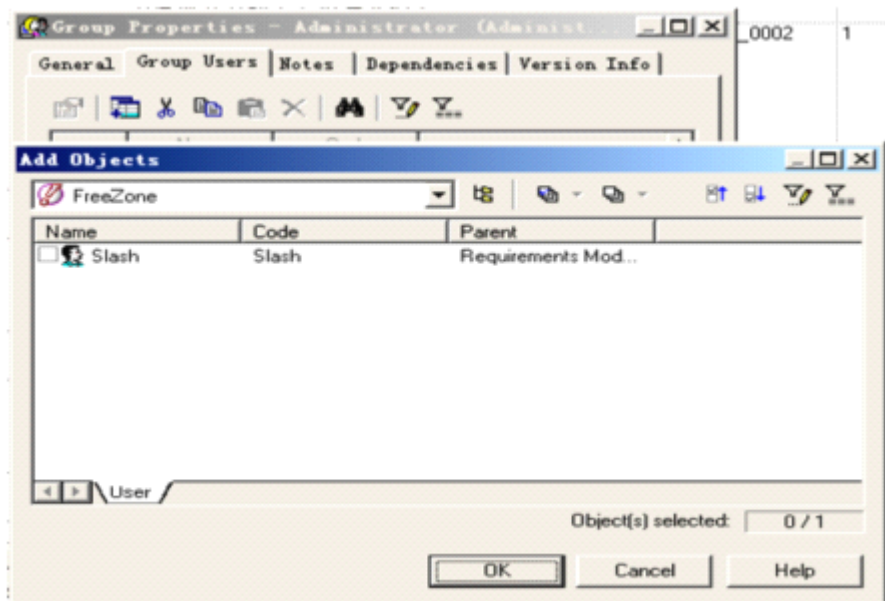
在资源浏览窗口中，右键单击模型名称（图标）-->New→User/Group,打开 User 或 Group 属性窗口，输入相应名称和代码名，确定即完成新建。

同样也可在菜单栏选择”Model”→Users/Groups 完成新建过程。

下一步是将相应的 User 与 Group 联系，添加进 Group 中，打开相应的 Group 属性，选择 Group Users 属性栏



点击属性工具栏中”Add Objects”工具，从中选择您要添加的 User 对象，当然只有在您已经建立了相应的 User 对象时才会显示 User 成员列表。



现在选择您需要添加的 User 对象，确定就可以了。

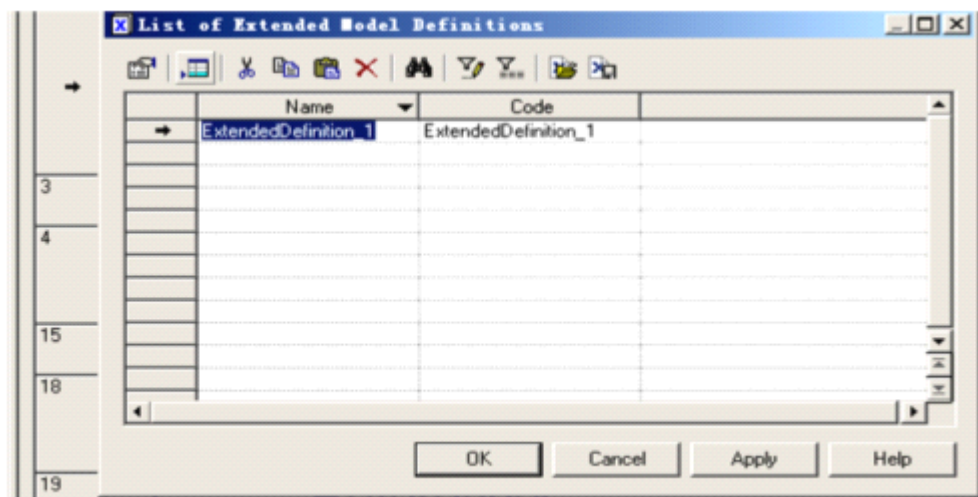
建立 Business rules(业务规则)

业务规则是对为了满足业务需求，模型应该包括的特定内容或关于如何构建模型方面的描述清单。在这里的示例模型中，我们要定义关于论坛积分制度的业务规则，具体业务规则内容见参考文档。

在 Requirement Model 状态下，PowerDesigner 默认 Businss 为不可用状态，为此我们需要通过新建 Extended model definition（扩展模型定义）来激活 Business rules。

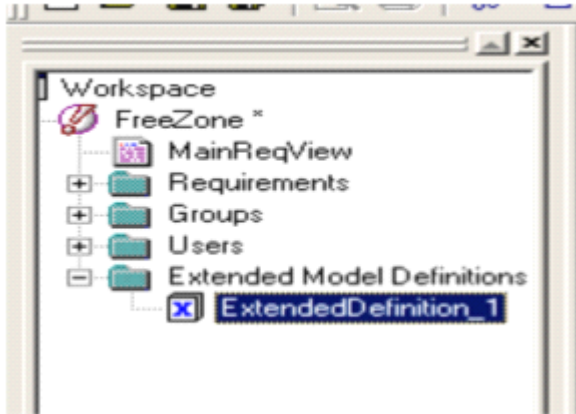
步骤如下：

选择菜单栏”Model”→ Extended Model Definitions，这时打开 List of Extended Model Definitions，通过选择其工具栏中”Add a Row”工具，如下图：

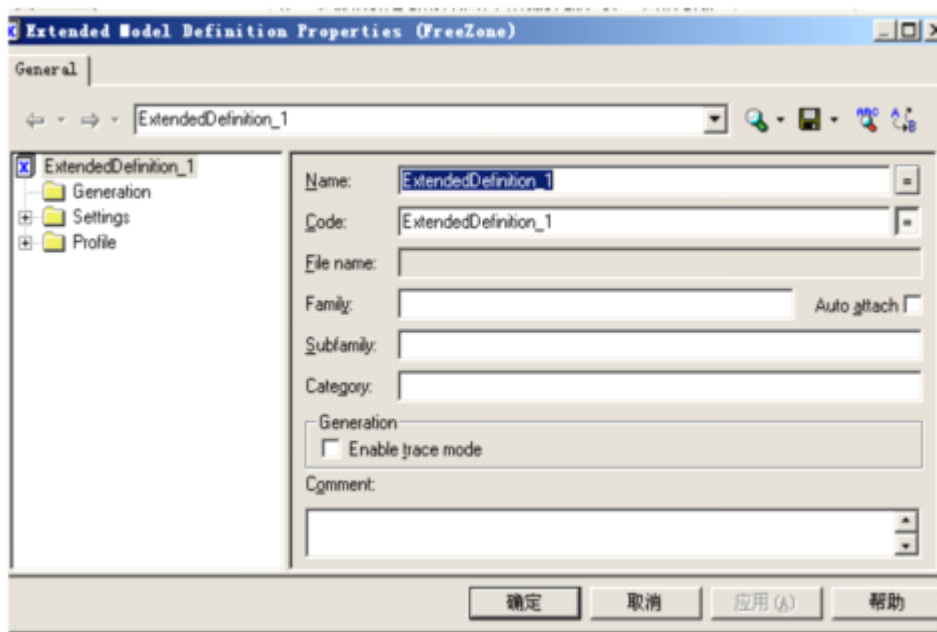


点击 Apply 即在资源浏览窗口中添加 Extended Model Definitions 目录。

在资源浏览器中打开 Extended Model Definitions 目录，双击相应扩展模型定义左边图标

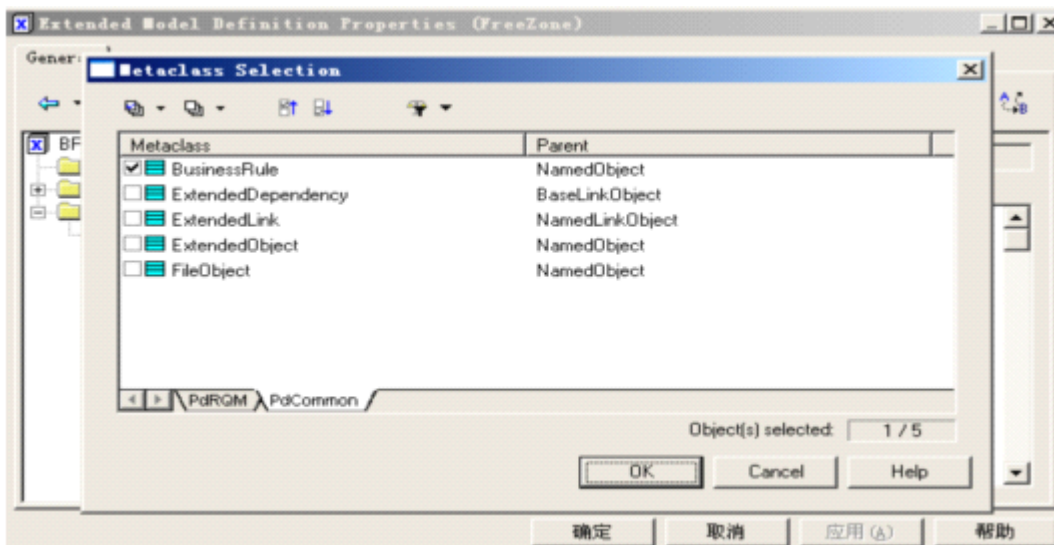


即打开 Extended Model Definition Properties



现在可以在右边输入 extended model definition 的 Name,Code 等信息。

选择左边窗口中”Profile”目录，右键单击在上下文菜单中选择”Add Metaclasses...”，这时可以看到 Metaclass Selection 对话框，选择 PdCommon 页，在 Metaclass 选择列表中选定 BusinessRule



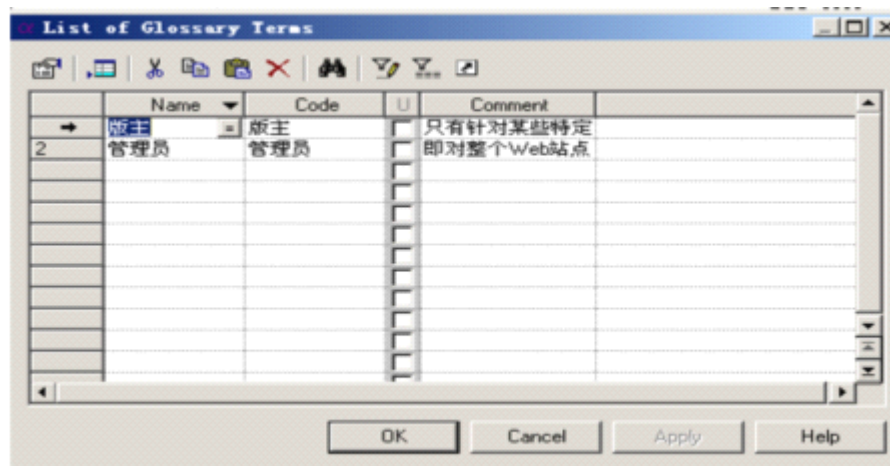
点击 OK,现在可以在 Profile 目录下看到 BusinessRule 了, 点击 OK!已经完成了 BusinessRule 的激活。

完成上述激活步骤后我们就可以执行 Business Rules 的新建了。

在资源浏览器窗口中右键单击当前需求模型->选择”New”,或通过选择菜单栏上 Model,你可以看到 Business Rule(s)选项了, 选择执行, 设定详细业务规则属性内容就 OK 了, 示例模型中我们完成了三个关于论坛积分制度方面的业务规则, 可以查看参考文档, 不再赘述!

接下来我们为示例模型添加术语表(glossary term)

选择菜单栏 Model->Glossary terms,进入 List of Glossary terms 对话框



选择工具栏上“Add a Row”工具，进行 glossary term 编辑。

或通过资源浏览器中也同样能执行添加术语操作。

若目标系统比较大，功能较多，也可以通过在系统模型中添加文件夹(package)来方便管理，也能使整个模型更清晰，具有层次性。

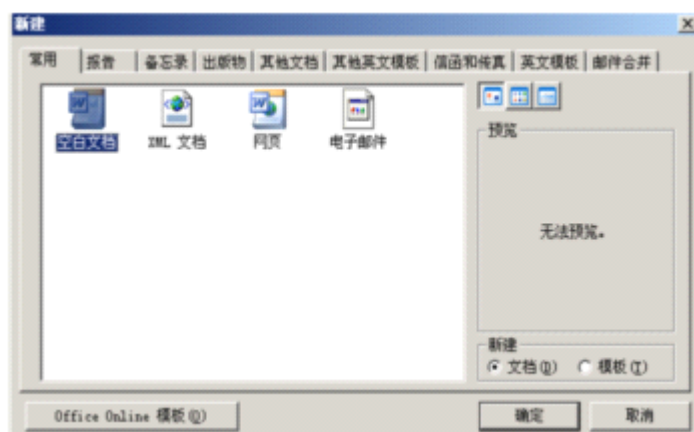
到这我们就已经基本完成了整个需求模型,接下来让我们来与 word 文档协调工作且生成内容全面的需求报告文档。

从需求模型生成 Word 文档

资源浏览窗口中，右键单击当前模型名称或图标->选择”Export as Word Document”

或在菜单栏中选择 Tools->Export as Word Document..., 这时文档生成就开始执行, 输出窗口会显示对当前模型的检验信息, 这里我们对其中的 Warning 就忽略不作考虑了。

片刻后会弹出



选择空白文档，单击确定，你可以看到文档输出了！

生成的文档如



其中红色部分文字表示与当前模型联接的信息，如果已经确定需求模型，要生成最终文档作为分析成果，可以通过在 MS Word 菜单栏上选择”Requirements”->Detach the Document from the Requirements Model，这样就实现了最终文档与需求模型的分离，同时生成的文档也没有那些红色的联接信息了。

在没有将文档与模型分离时，我们还可能在 PowerDesigner 中对需求模型进行修改，这时我们可以对文档执行更新操作，同时对符合层次化标准的 Word 文档，也可以将其转化为相应需求模型。

需求模型的个人见解就到此为止，要申明的是：以上内容只是对 PowerDesigner 提供的需求建模功能的大概说明，其中太多细节还需日后使用过程中慢慢掌握。

生成模型报告

文档生成 Report

个人觉得有必要将 Report（文档生成）提前讲解，毕竟软件工程的任何阶段都会输出相应文档，PowerDesigner 支持生成 RTF 和 HTML 两种格式文档。

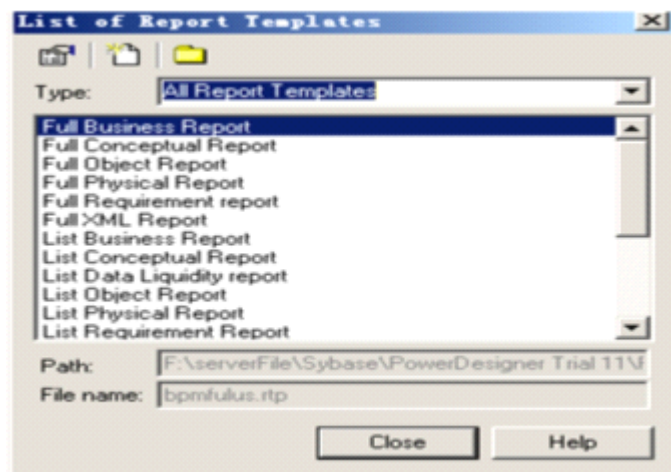
下面以刚完成的示例论坛系统的需求模型为例讲解。

PowerDesigner 提供对 Report 的操作有关于 Report Template Editor(报告模板编辑器)，Report Template(报告模板)，Report Editor(报告编辑器)，Multi-Model Report Editor（多模型报告编辑器），Report Language Editor（报告语言编辑器）

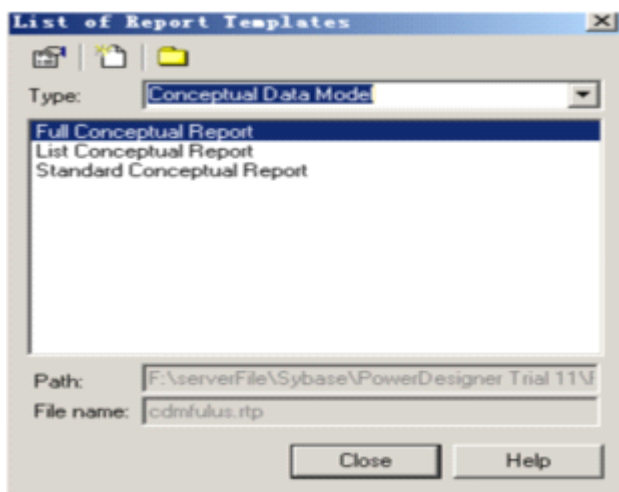
1. 使用 Report Template Editor（报告模板编辑器）

打开 Report Template Editor（报告模板编辑器）

（1）选择 Tools->Resources->Report Templates,可以打开 List of Report Templates（报告模板列表），列表显示出当前系统中存在的报告模板，如下图所示：

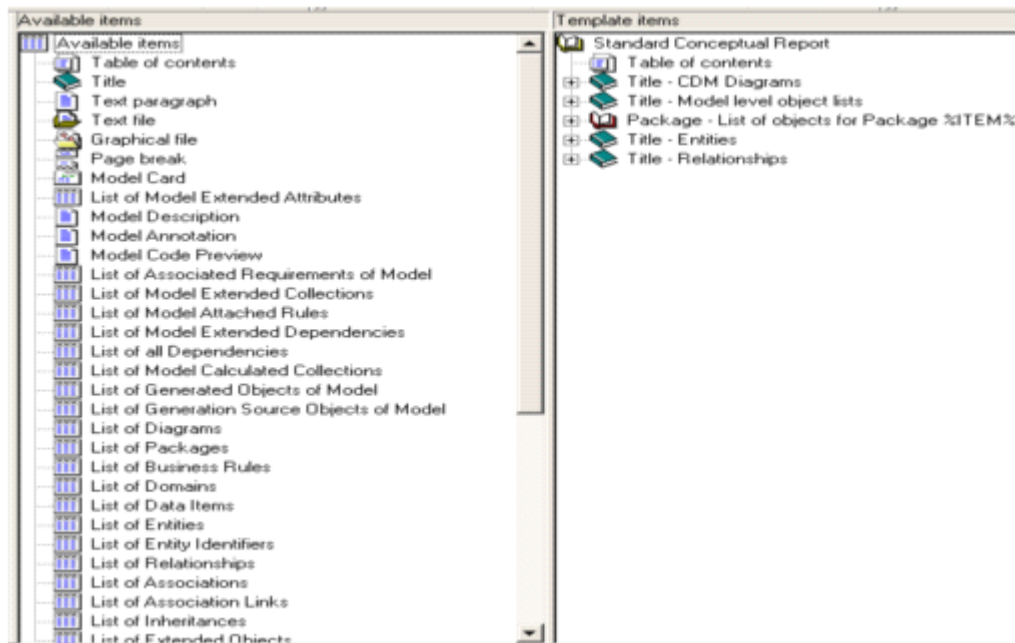


(2) 在 Type（类型）下拉列表中选择相应的模板类型，可用模板中会显示对应您选择模板类型的模板，同时您可以通过单击模板列表工具栏上的 New 工具新建您所需要的模板。



(3) 选择相应模板

(4) 通过单击工具栏上 Properties 工具或直接双击所选定模板，进入相应模板属性编辑器。其中左边 Available items 为可用项目，右边 Template items 为当前模板中项目，表示出该报告模板的结构。



现在你可以对该报告模板进行编辑修改！

也可以通过选择 Model->Reports 打开 List of Reports（报告列表），再选择报告列表工具栏上 Manage Report Templates（管理报告模板）工具打开 List of Report Templates（报告模板列表）。

2. 标准报告模板（Standard report templates）

PowerDesigner 默认自带了一系列的标准报告模板，其模板安装目录在 Sybase\PowerDesigner Trial 11\Resource Files\Report Templates 下。

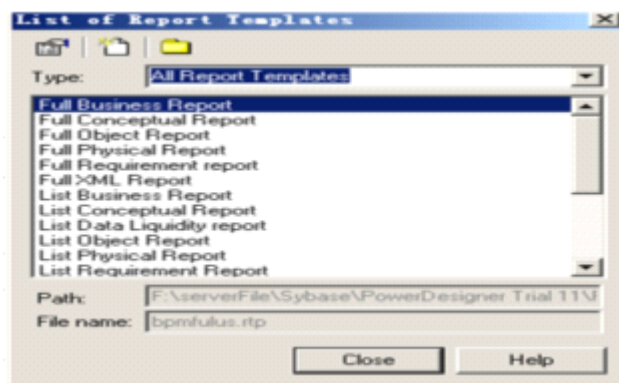
其中每种类型的模板都包含三种类型的标准模板，如下表所示：

模板类型	命名规则	所生成报告内容
Full	modelFULlanguage.RTP	目录，所有主要的模型项目
Standard	modelSTDlanguage.RTP	目录，模型图，包图和大部分 List 对象
List	modelLISlanguage.RTP	标题对象，所有的 List 对象

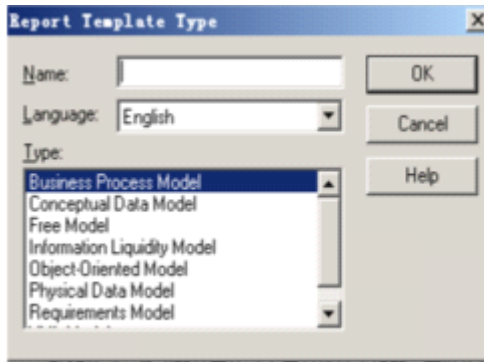
3. 创建报告模板

报告模板是一种可以用来快速生成报告文档的文件，你可以使用 PowerDesigner 自带的一些标准模板或创建你自己的模板。创建模板时需要指明你在报告中需要包含的信息，同时也可以通过选择一种你想要语言用以显示报告文档。

- (1) 选择 Tools->Resources->Report Templates,打开了 List of Report Templates 窗口



- (2) 工具栏中 New 工具，即打开 Report Template Type 窗口



- (3) 输入相应的模板名称，选择语言种类同时在模板列表中选择模板类别
- (4) 单击 OK 即进去 Report Template Editor (报告模板编辑器)，现在你可以将想要在模板中显示的项目进行添加调整了。
- (5) 完成模板编辑后，选择 File->Save,就可以将你所编辑好的报告模板保存为.rtp 文件。

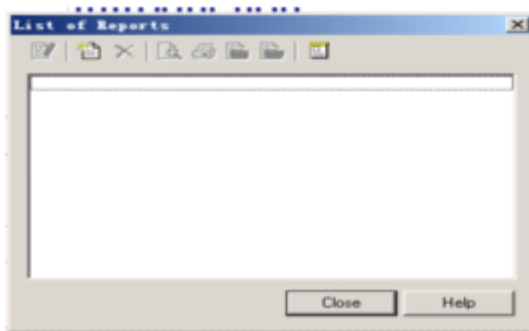
使用报告编辑器 (Using the Report Editors)

1. 创建模型报告

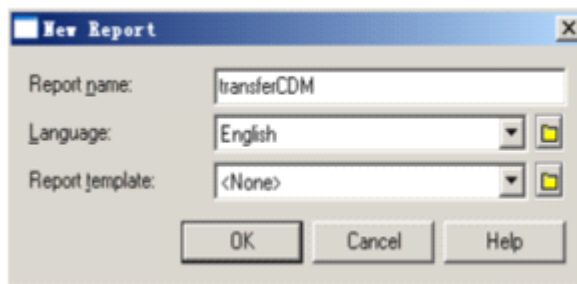
你可以通过使用报告编辑器创建模型报告和多模型编辑器创建多模型报告，但是当你要创建报告时，在当前 workspace 中必须打开至少一个模型且要有一个默认生成节点。

这里我们对前期需求模型创建报告文档作为示例。

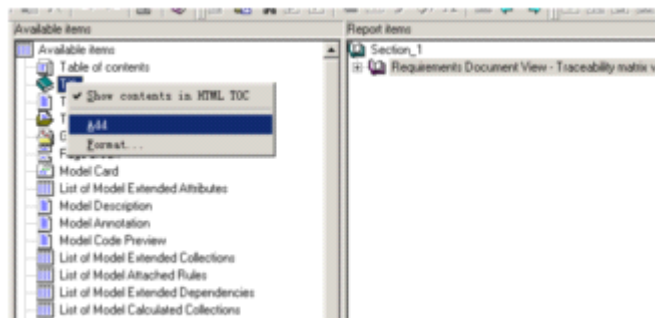
- (1) Model->Reports,打开 List of Reports 窗口



- (2) 单击工具栏上 New 工具，弹出 New Report 对话框，输入对应名称，选择语言类别和报告模板。

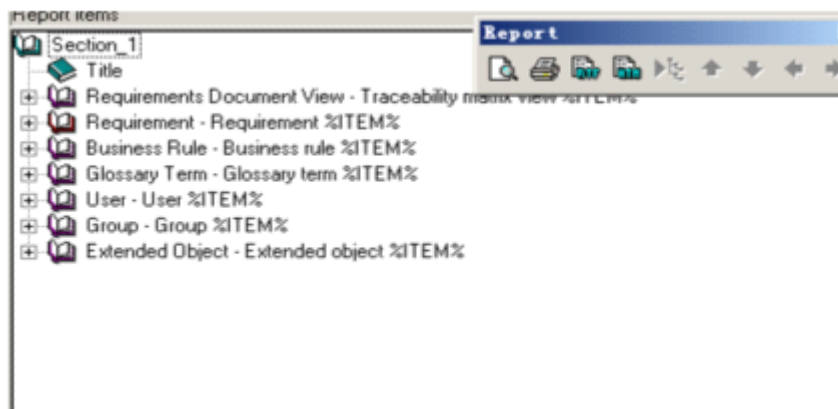


- (3) 单击 OK,即完成报告新建工作，这里我选择的报告模板为 None,接下来我们对报告内容及节点进行编辑。

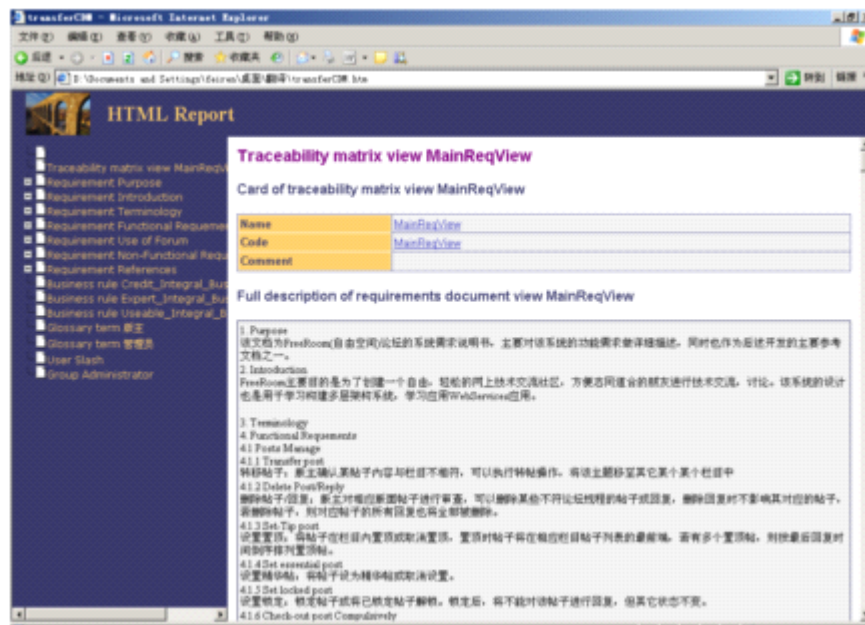


如上图对目标报告文档内容进行编辑，报告节点设计完毕后就可以生成 html 或 rtf 报告了。

(4) 选择 Report 面板中的 Generate HTML 或 Generate RTF 即可生成相应格式报告文档，若要预览文档，可以选择面板中的 Print Preview 工具。



最终生成的 html 文档如下图所示：



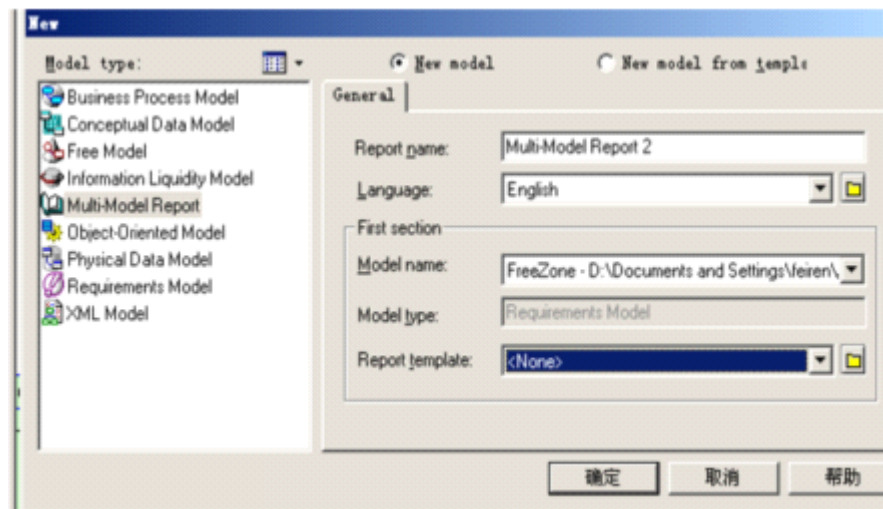
创建多模型报告（Multi-Model Report）

多模型报告能够通过使用 Section 在同一个报告中包含不同类型模型中的对象，将不同模型结合起来提供一个全局视角的综合报告。但是每个 Section 只能是一种模型类型，并且只能使用一个模板类型，被使用模型必须在当前 workspace 中处于打开状态。

(1) 在当前 workspace 中打开一些需要参与多模型报告的模型，选择菜单栏中 File->New，在弹出的新建窗口中选择 Multi-Model Report。

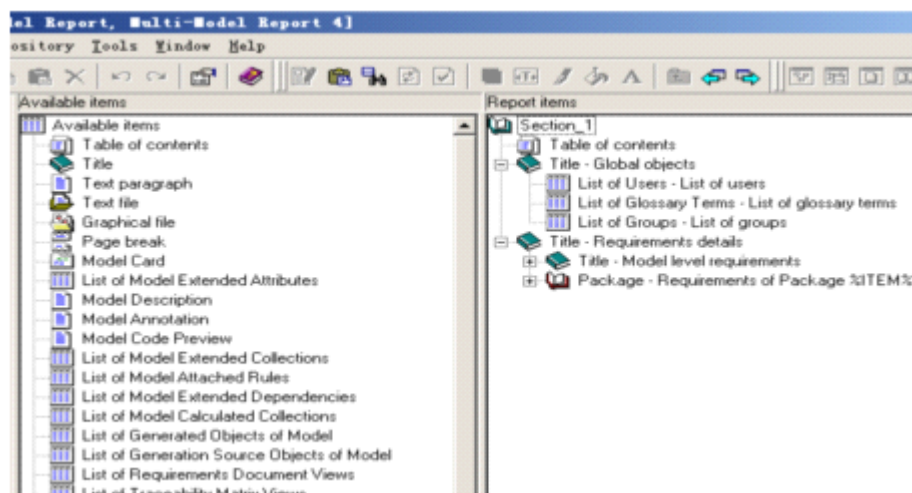
或通过右键单击当前 Workspace->New->Multi-Model Report 亦能完成多模型报告的新建。

(2) 弹出新建多模型报告窗口。



(3) 输入报告名称，选择语言种类，同时在 Model name 的下拉框中选择一个 Section 将要描述的模型。同时根据可以根据需要在 Report template 下拉列表中选择相应的报告模板。

(4) 点击 OK,确认操作！这时就已经打开多模型报告编辑窗口，如下图所示：



(5) 基本的多模型报告框架已经建好，下一步就是对其中 Section 进行设置编辑即可。根据需要加入不同模型创建适当 Section，基本操作与普通单模型报告类似！

处理 Section

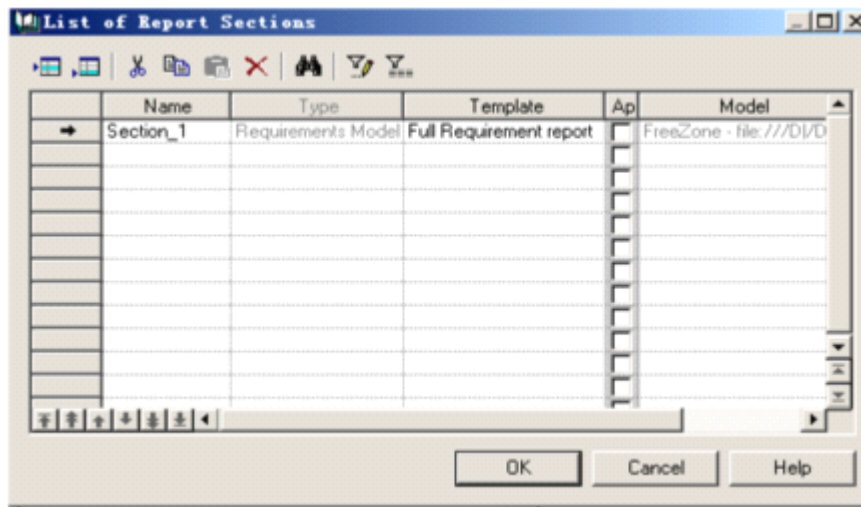
每个报告文件至少要包含一个 Section，通过使用 Section 可以使模型设计者将目标模型分为几个不同部分，便于分析模型各部分功能，因此恰当地使用 Section 可以让报告文档更加清晰，具有层次性。可以通过两种方式创建 Section（节）：

1. 创建一个空白 Section
2. 创建一个基于模板（Template）的 Section

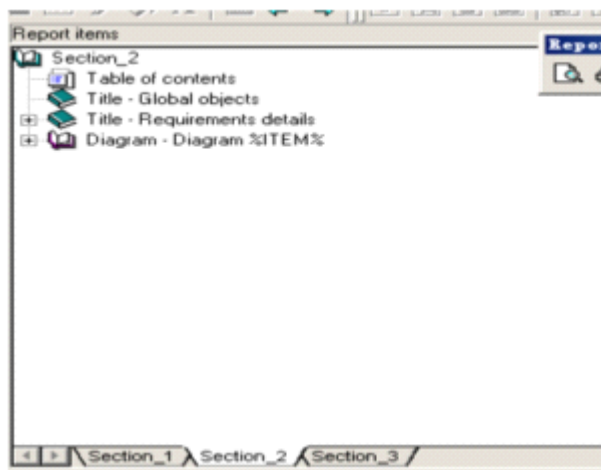
当你创建新 Section 时，模板列属性默认被设置为 None,且应用模板选项框被自动选取。

创建 Section

- (1) 在 Report Editor 编辑窗口下，选择 Report>Sections,即弹出 List of Report Sections 窗口，其中 Section 列表包含一默认生成的 Section.



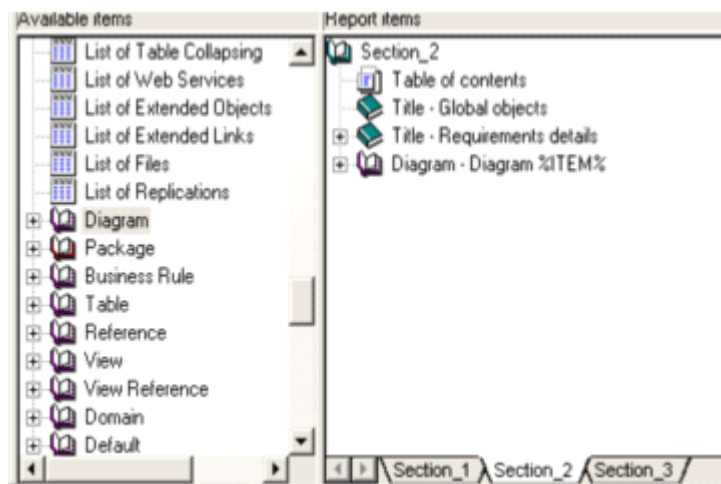
- (2) 输入 Section 名称，如果没有更改输入名称系统将会在报告项目面板（Report Items）中使用默认名称。
- (3) 如果当前报告为多模型报告，则可以在模型栏（Model column）中选择对应模型类型，多模型报告可能包括多种模型类型的 Section,如 OOM,PDM,CDM 等，但必须这些模型在当前 Workspace 中都处于打开状态。若当前报告为单模型报告，则 Model 列为不可选，系统自动设置为当前模型。
- (4) 单击模板栏（Template）,选择需要的模板类型，可选项有 None,Full Requirement report, List Requirement report,Standard Requirement report。若选择 None 则创建空白 Section.,同时 Apply Template选项框为默认选取状态。
- (5) 以上操作已经完成对当前 Section 的设置，要再次添加 Section 则通过选取工具栏上 Add a Row 或 Insert a Row 工具添加新 Section,同时再次执行（1）---（4）步骤设置其属性。
- (6) 单击 OK，现在已经完成多个 Section 的创建。其中每个 Section 在报告编辑器（Report Editor）中显示为 Report Items 面板底部的 Tab 页中，如下图所示：



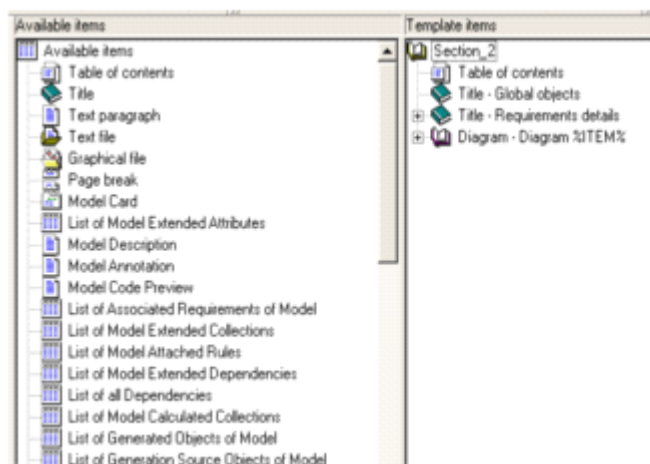
将报告中 Section 创建为模板

经常我们需要将已经设计好的 Section 供以后在其他模型中使用，为此我们可以将创建好的 Section 保存为模板。

- (1) 单击需要保存的 Section 的 Tab 页（在 Report Items 面板底部的 Tab 页）



- (2) 选择工具栏上 Report->Create Template From Section,打开报告模板编辑器（Report Template Editor）页面，则原来在报告项目（Report Items）中显示的项目（Items）这时显示在模板项目面板中。



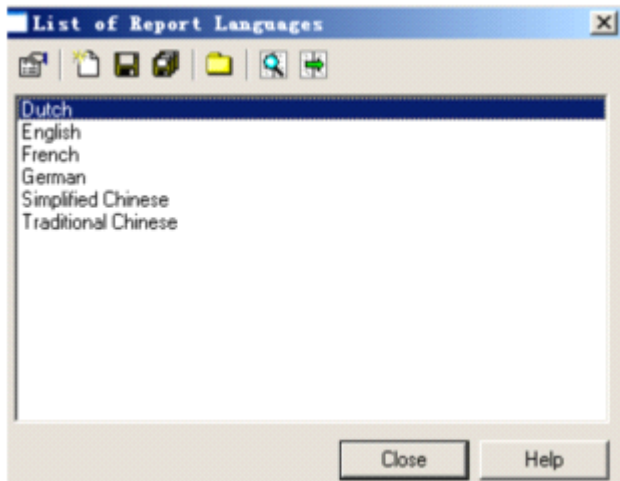
- (3) 确认模板项目后，选择菜单栏上 File->Save,即打开保存文件对话框。
- (4) 输入相应模板名称，单击保存即可。

使用报告语言编辑器（Using the Report Language Editor）

通过使用报告语言编辑器可以创建和修改报告语言的源文件（Resource files），报告语言源文件是以 XRL 为后缀的 XML 格式文件，其中包含了报告中所有可打印文本和它们的一些默认数据，报告语言源文件保存在中心区域且能够被任何模型报告共享使用，从而保证了数据一致性，节省了用户创建编辑报告文档的时间。PowerDesigner 在安装目录\Sybase\PowerDesigner Trial 11\Resource Files\Report Languages 下自带了一系列的报告语言源文件。我们也可以通过使用报告语言编辑器（Report Language Editor）创建符合自己需求的文档报告源文件。

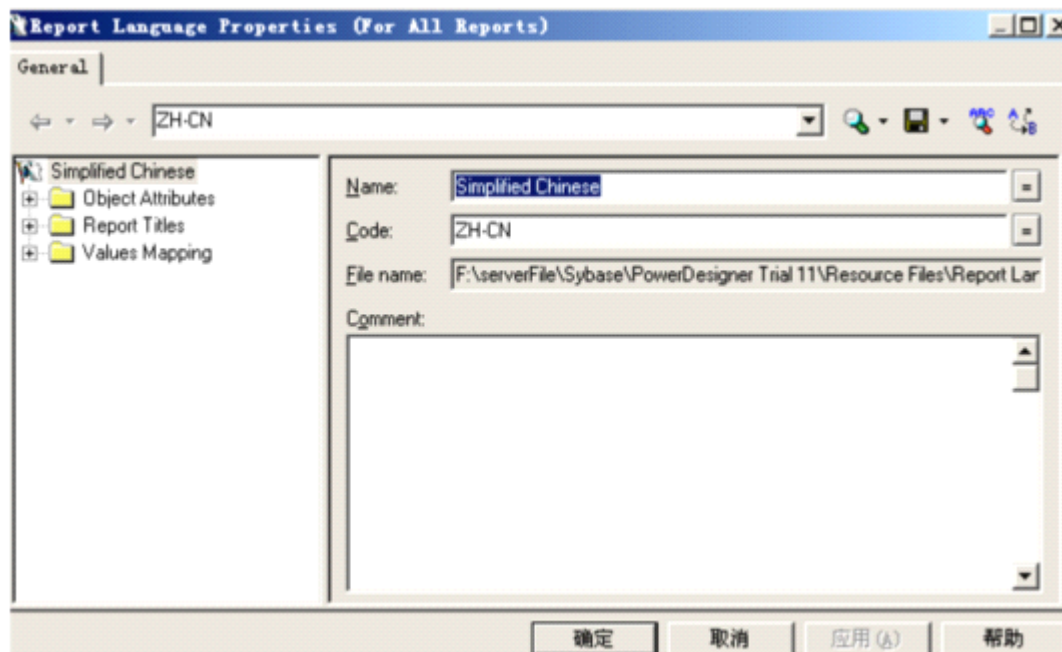
打开报告语言编辑器

- (1) 选择菜单栏上 Tools->Resources->Report Languages,即打开报告语言列表（List of Report Languages）窗口，其中显示出当前系统具有的所有报告语言列表。



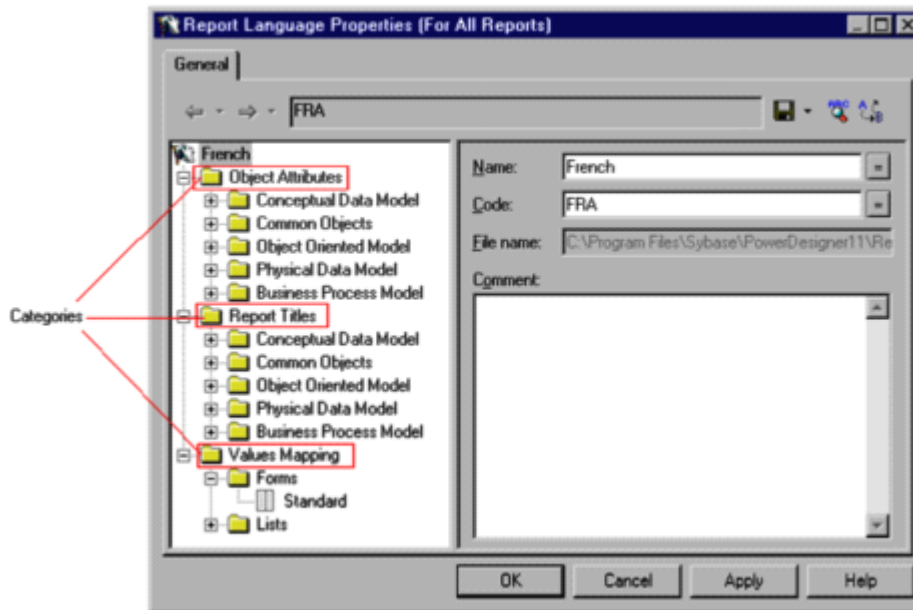
(2) 选择某种报告语言

(3) 单击工具栏上 Properties 工具，或双击该行，打开 Report Language Properties 窗口



同样你也可以通过报告编辑器打开报告语言编辑窗口：选择菜单栏上 Report->Edit Current Language，不过这时打开的语言种类是针对当前选择语言。

报告语言编辑器（Report Language Editor）由两个不同部分组成：根据语言类别和实体导航的左侧目录树与显示相关信息的右侧树型视图。



左边目录树主要包含以下三个部分

类别	描述	翻译用途
对象属性	包含每个模型中所有和对象相关联的字符串，如对象属性的名称，代码	Cards,checks,list 中对象属性的名称翻译
报告标题	包含每个模型中所有与报告项目相关联的字符串，如组织单位注释等	所有报告项目的标题翻译
值映射	包含所有与用于属性数值的关键字相关联的字符串，如未定义，或不存在等	Cards,checks,lists 中对象属性数值的关键字翻译

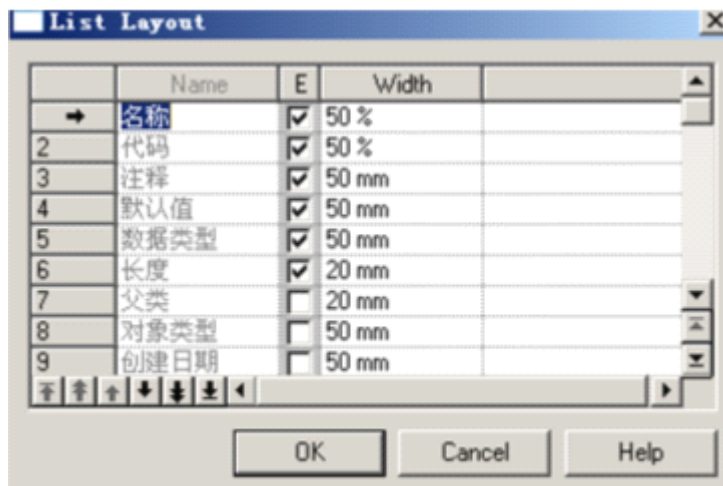
Object Attributes 和 Report Titles 分别包含 PowerDesigner 每个模型特定特征的种类。Value Mapping 类别则包含具有一个标准入口的子类别：

Forms: Cards 和 Checks 中的对象属性的关键字。

Lists: Lists 中的对象属性的关键字。

PowerDesigner 提供自带的报告语言源文件还是很符合语言习惯的，一般来说不用进行更改订制，但选择中文模板时会出现一些问题，比较常见的就是如 Primary Key,Foreign Key 等翻译存在一些差异。下面以简体中文模板对我们示例系统的 PDM 建立系统数据字典报告文件。

已经完成的示例 PDM 关系图如下：

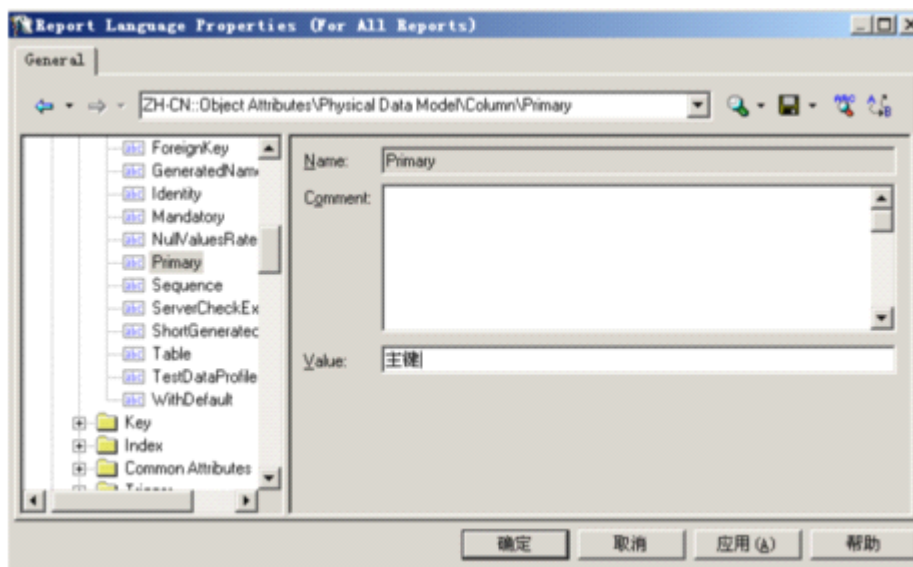


(3) 在列表中选择需要显示的对象。
这时直接生成 RTF 文档，看看文档效果。

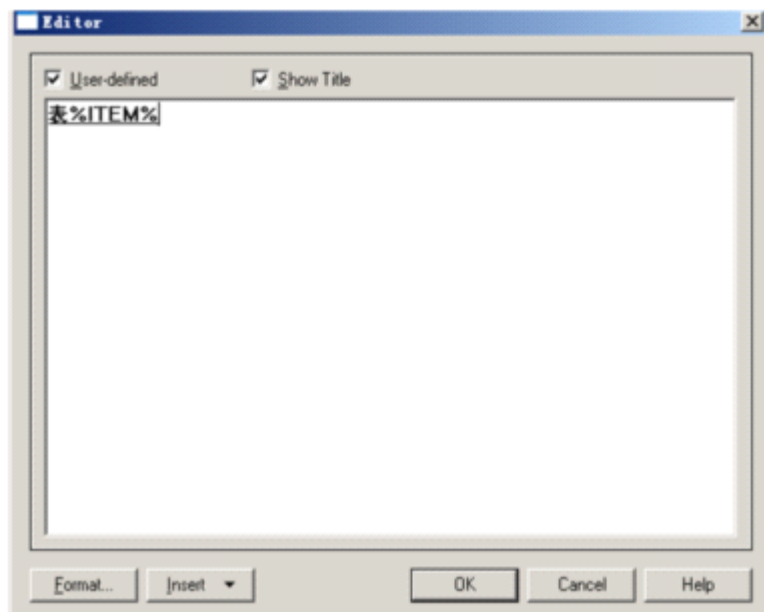
名称	代码	注释	默认值	数据类型	属性	主要的	外键
U s e r I D	U s e r I D	用户编号,唯一区别标识系统中各用户.		int	T R U E	T R U E	F A L S E
R o l e I D	R o l e I D	角色编号,唯一标识系统某一类别角色.		int	T R U E	F A L S E	T R U E
U s e r N	U s e r N	用户名称,即注册ID,可用于系统登录,不能重复.		nvarchar(40)	T R U E	F A L S E	F A L S E

看到上述文档效果估计很多朋友都会很失望的，没关系，现在让我们一步步来完善！

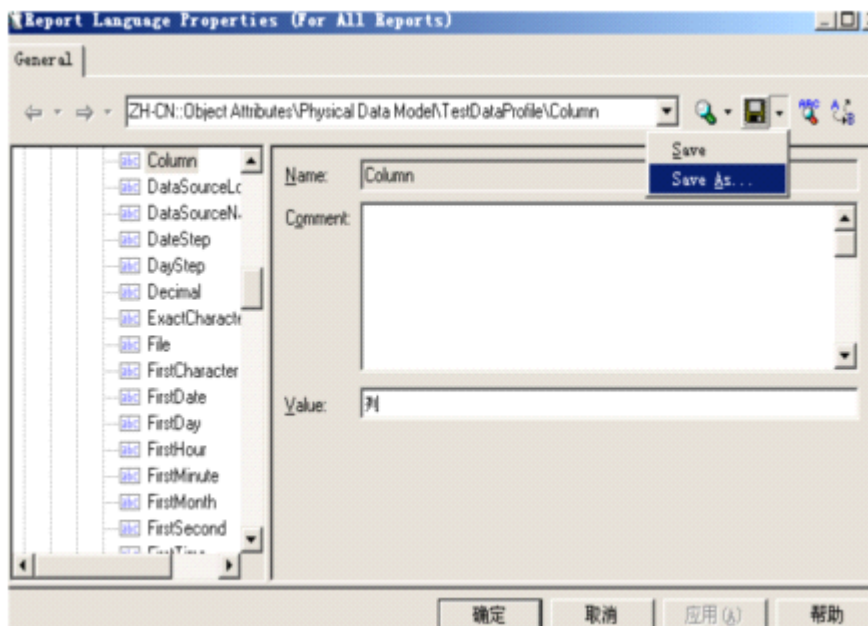
- (1) 选择菜单栏上 Tools->Resources->Report Languages... 打开 List of Report Languages（报告语言列表）窗口，这里我们选择双击 Simplified Chinese,以打开报告语言属性窗口。
- (2) 选择 Object Attributes\Physical Data Model\Column\Primary,将 Value 中“主要的”改为“主键”。



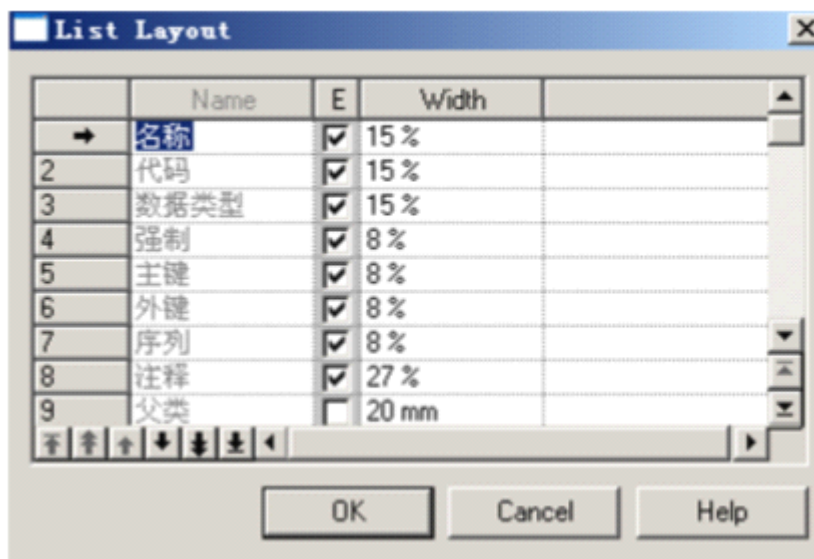
- (3) 选择 Object Attributes\Physical Data Model\Column\ForeignKey，将 Value”外来键”改为”外键”
- (4) 选择 Report Titles\Physical Data Model\Table\Columns list，将 Value”表格%PARENT%的专栏清单”改为” 表%PARENT%的列清单”。
- (5) 双击报告项目面板中的”Table- 表格%ITEM%”对象，在弹出的编辑窗口中将”表格%ITEM%”改为”表%ITEM%”，如下图：



- (6) 当然还可以通过报告源文件编辑器进行其它报告项目显示方面的更改，同时也可以使用如其它常用软件中的[查找替换](#)功能，可以在报告语言属性窗口找到相应工具。不过这时执行的是全局替换，使用前应小心。
- (7) 通常在进行报告语言属性进行更改之后，为了保证软件自带的报告语言源文件（.xrl 文件）不发生变更，可以选择”Save As...”命令。不过必须在语言报告属性窗口中执行，如下图：



(8) 调整各属性列宽度，右键单击报告项目（Report Items）面板中”List of Table Columns-表 %PARENT%的列清单”，在弹出菜单中选择”Layout”，打开 List Layout 窗口，如下图：



现在调整 Width 列的数值就行了，支持百分比和实际宽度两种属性。现在可以看看生成的文档了，如下图所示：

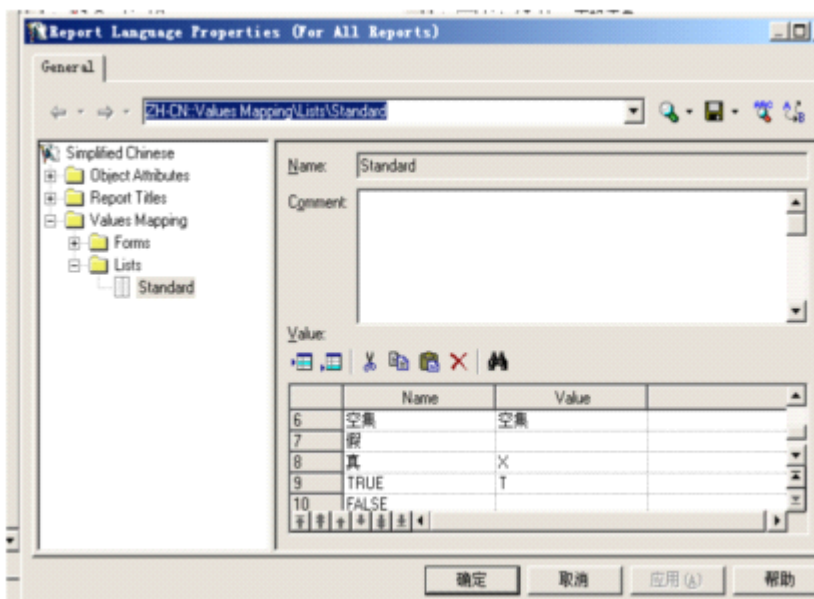
2.1 表 ForumColumn 的列清单

2.1 表 ForumColumn 的列清单

名称	代码	数据类型	标识	主键	外键	序列	注释
ColumnID	ColumnID	int	TRUE	TRUE	FALS E	<Non e>	论坛栏目编号,用于唯一标识论坛栏目。
GroupID	GroupID	int	TRUE	FALS E	TRUE	<Non e>	论坛栏目组编号,唯一标识各栏目组。
Name	Name	nvarchar(50)	TRUE	FALS E	FALS E	<Non e>	栏目名称。
Description	Description	nvarchar(500)	FALS E	FALS E	FALS E	<Non e>	论坛栏目描述,简要概述论坛栏目内容。
Update Time	Update Time	smalldatetime	TRUE	FALS E	FALS E	<Non e>	栏目更新时间,指对栏目进行系统设置时间,取系统时间。
IsAvailable	IsAvailable	bit	TRUE	FALS E	FALS E	<Non e>	是否可用,1为可用,0为不可用,默认设置为1。
SortOrder	SortOrder	int	TRUE	FALS E	FALS E	<Non e>	栏目排列序号,用于安排论坛栏目在主页的显示顺序。

为了使显示效果更简洁点,不妨将其中大部分的 FALSE 不显示,TRUE 也只用 T 代换,为此我们需要将系统的 TRUE 和 FALSE 进行转换,需要在报告语言属性中更改映射表。

- (9) 在报告语言属性窗口中选择“Values Mapping\Lists\Standard”,添加 True 和 FALSE 映射即可,如下图所示:



现在在进行文档生成,基本上满足正式文档要求。

当然关于模型报告还有很多细节问题,这里不能做到一一分析,可以在日后实际使用中慢慢发掘,毕竟运用才是关键!好了,这一小节就到此为止!

概念数据模型 CDM(Conceptual Database Model)

以下我们要完成对示例论坛系统的数据库设计工作,首先让我们建立目标系统的概念数据模型(CDM)。

在进行相关 CDM 演示之前,让我先简要介绍概念数据模型(CDM)的相关概念。我们进行数据库设计时,一般都是概念层次(Conceptual level)开始的。在概念层次上,你无须考虑数据库的实际物理执

行细节。概念模型（CDM）描述了与任何软件或数据存储系统无关的数据库整体逻辑结构，通常包含了与物理数据库无关的数据对象，提供了一种对用于运行企业或业务行为的形象化的表达方式。

CDM 功能：

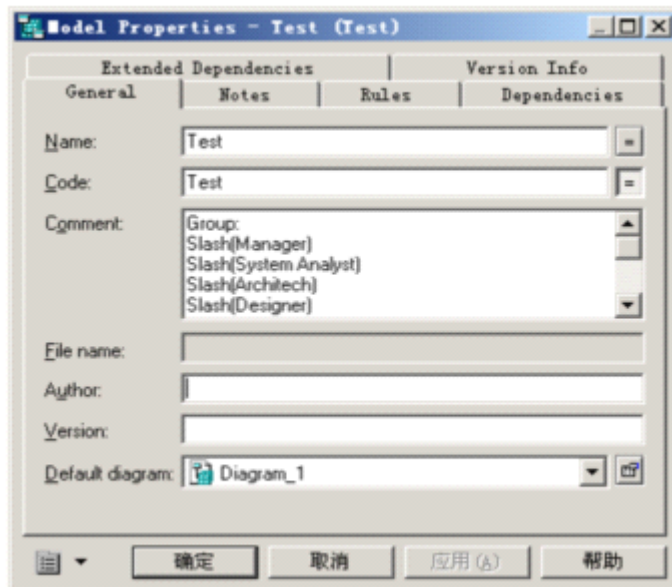
- （1）通过创建实体关系图表（E-R）来描述数据的组织结构。
- （2）能够校验数据设计的合理性。
- （3）生成指定了相应物理实现数据库的物理数据模型（PDM）
- （4）能够生成用 UML 标准描述 CDM 中对象的面向对象模型（OOM）
- （5）为在不同的设计阶段创建另一个模型版本，可以生成概念数据模型（CDM）

关于 Palette 工具面板中含义简介：

工具	对应名称	含义
	Pointer	选择图形
	Grabber	选定某个范围的图形
	Zoom In	放大
	Zoom Out	缩小
	Properties	显示相应图符的属性
	Delete	删除图符
	Package	插入一个包（package）的图符
	Entity	插入一个实体图符
	Relationship	插入一个关系（Relationship）图符
	Inheritance	插入一个继承（Inheritance）图符
	Association	插入一个关联（Association）图符
	Association Link	插入一个关联连接（Association Link）图符

新建 CDM

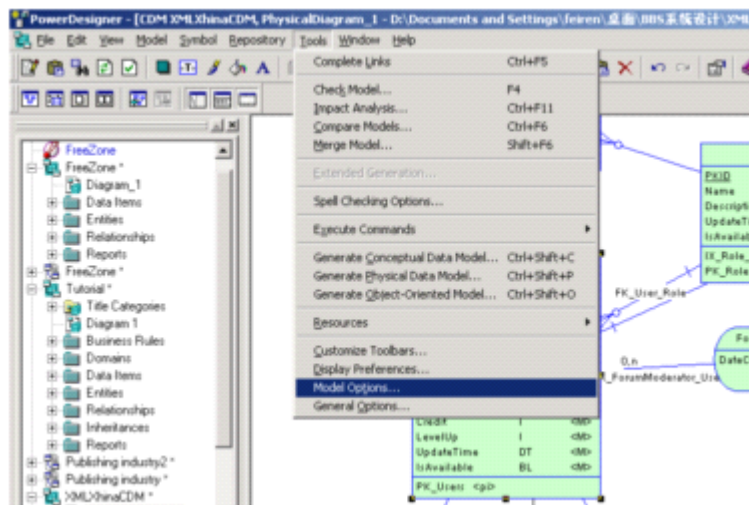
- （1）选择 File->New，打开 New 窗口，在左边模型选择列中选 Conceptual Data Model,单击 OK，即确认创建概念数据模型。
- （2）双击资源浏览窗口中新创建的 CDM 名称图标，打开 CDM 模型属性窗口，进行相关属性信息设置。如下图：



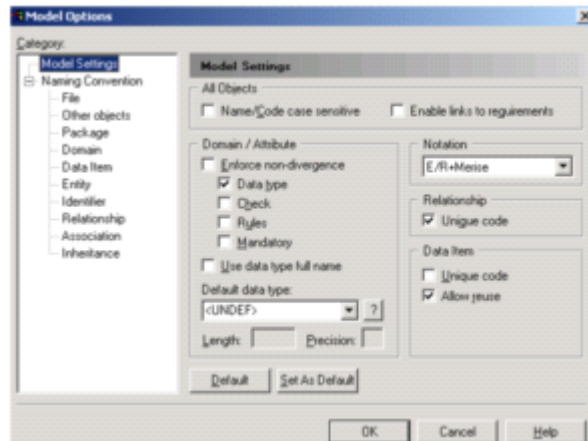
对刚创建的 CDM 进行详细之前有必要先说说有关实体属性命名问题。

PowerDesigner 默认在 CDM 中不能存在相同名称的实体属性，这也是考虑到可能产生的一些如主键外键等名称冲突问题，但当我们进行实际数据库设计时，可能会多次使用相同数据项（DataItem）便于理解各实体。为此需要对更改 PowerDesigner 相关设置。软件默认认为 DataItem 不能重复使用(重名)，需要进行以下操作：

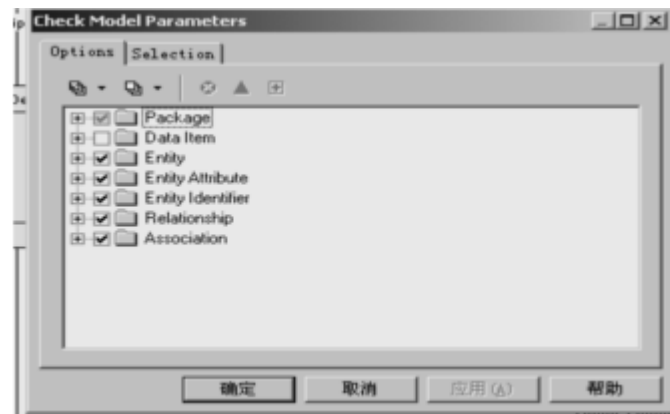
选择 Tools->Model Options,



在 Model Setting 设置目录中，将 Data Item 下的 Unique Code 取消选中即可，系统默认将 Unique Code 和 Allow Reuse 均选中。



同时该设置均是面向特定模型的，即针对当前模型有效，若希望在其它模型中也有此命名设置，则需要重新进行设置。不过在 Check Model 时，如果选择全部 Check，则依旧会报 DataItem 重名的错误信息，这时需要我们在人为检查确认数据项无误时，可以在选择不对 DataItem 不检查，如下图示：

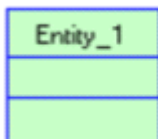


各种数据类型对应匹配(这里只给出与 SQL Server 中的常用对应类型，其它 DBMS 可以使用类似处理)

实体及各类关系

实体 (Entity)

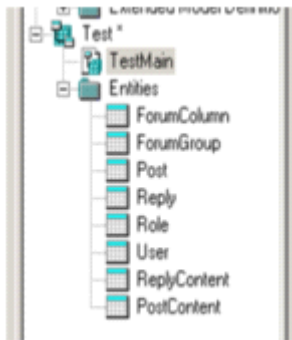
- (1) 在新创建的 CDM 中，选择 Palette 工具面板中的 Entity 工具，再在模型区域淡击鼠标左键，即添加了一个实体图符。



- (2) 单击鼠标右键或单击面板中 Pointer 工具，使鼠标处于选择图形状态。
- (3) 双击新创建的实体图符，打开实体属性窗口，输入实体名称和代码。
- (4) 单击 OK，即完成实体创建过程。

继续上述操作，创建多个实体，分别设置为不同名称，具体信息参考示例文档。

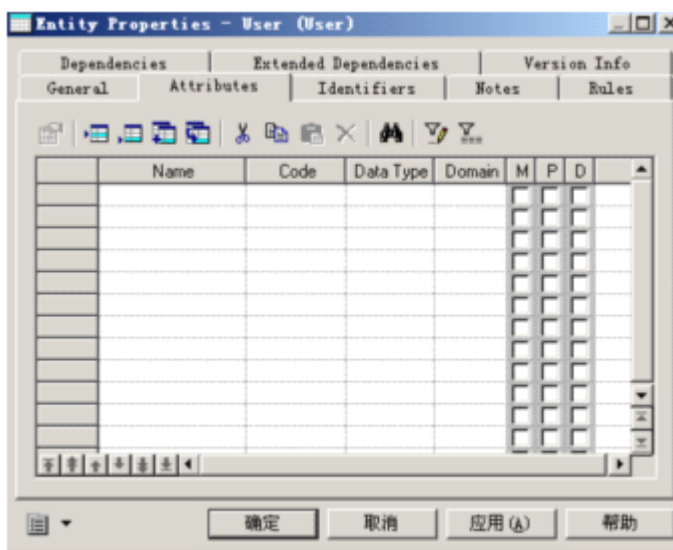
实体创建完成后资源浏览窗口中层次结构如下所示：



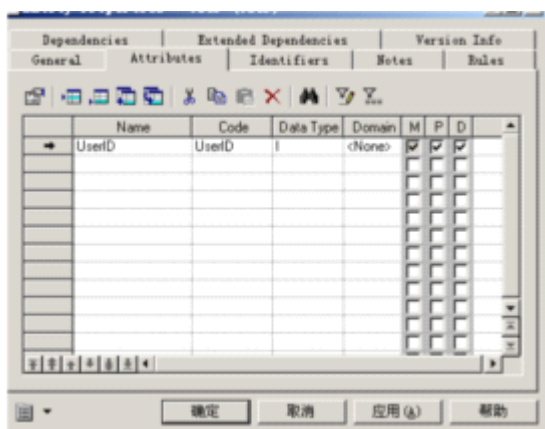
现在编辑各实体的详细内容，如属性组，实体间关系等。

实体属性（Entity Attributes）

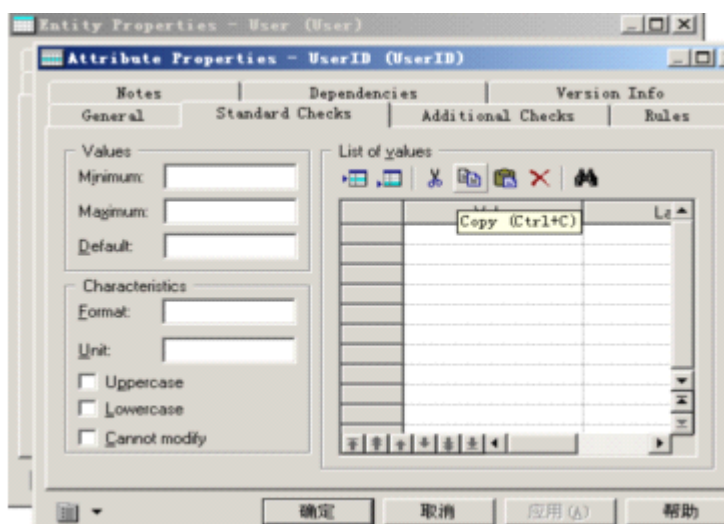
- (1) 以 User 实体为例，打开实体 User 属性窗口，进入 Attributes 属性页，如下图所示：



- (2) 单击属性窗口工具栏中 Add a Row 工具，即在属性实体属性列表中添加了一个属性，同时设置该属性相关信息，如数据类型，是否为主标识符，是否不可为空等。
- (3) 详细设置新添加的属性为 UserID,作为系统唯一标识区别的用户编号，同时选择 P,M,数据类型 (DataType) 选择 Integer。如下图：



- (4) 对属性列进行更为详细的设置，可以通过单击对应属性列左边箭头，进入 Attribute Properties 窗口，可以进行更为精确详细的设置，如数据上下限，精度等。如下图：



(5) 同时若要更改实体属性列表中显示的相关选项可以通过单击工具栏中 Customize Columns and Filter 工具以打开 Customize Columns and Filter 窗口



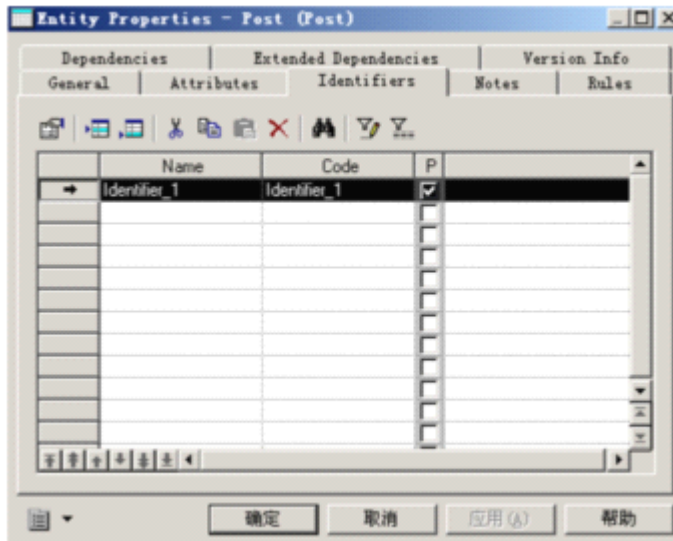
只要在列表中选择想要显示的项目即可完成设置。

标识符 (Identifiers)

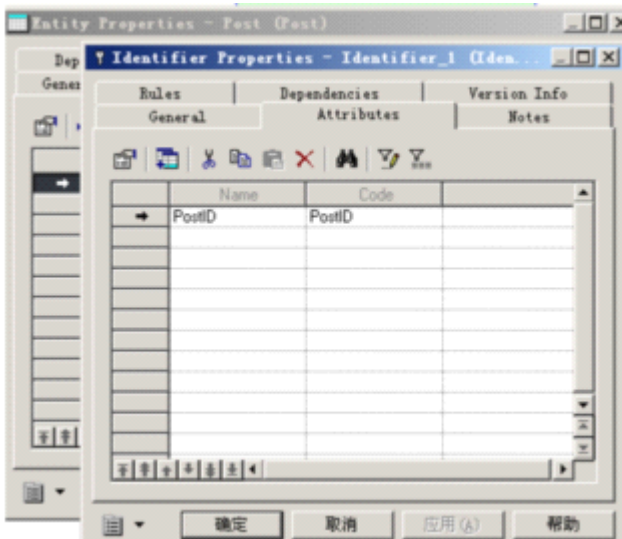
标识符是能够用于唯一标识实体的每条记录的一个实体属性或实体属性的集合, CDM 中的标识符等同于 PDM 中的主键 (Primary Key) 或候选键 (Alternate Key)。每个实体至少要有一个标识符, 若一个实体中只存在一个标识符, 它就自动被默认指派为该实体的主标识符 (Primary Identifier)。

指定相应标识符

- (1) 在双击图表中对应实体以显示实体属性窗口。
- (2) 在当前实体属性窗口中选择 Identifiers 属性栏, 如下所示:



- (3) 可以通过单击工具栏上 Property 工具或双击所要选择的标识符栏，进入标识符属性编辑窗口。
- (4) 选择 Attributes 属性，可以看到当前标识符所关联的属性列表，如下图：



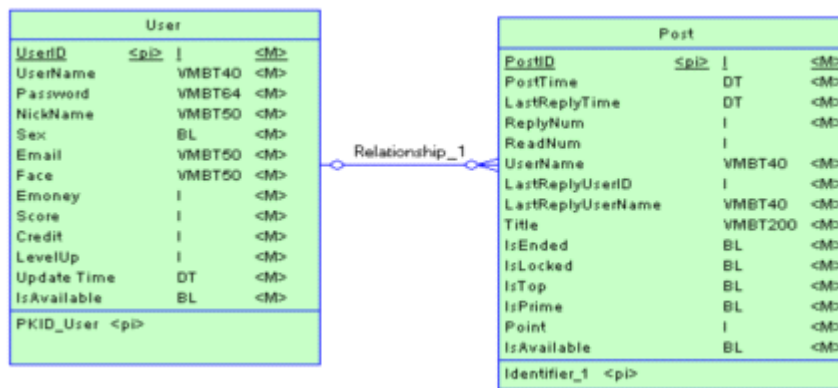
- (5) 单击工具栏中 Add Attributes 工具，即可以进行为当前标识符添加属性。

关系（Relationship）

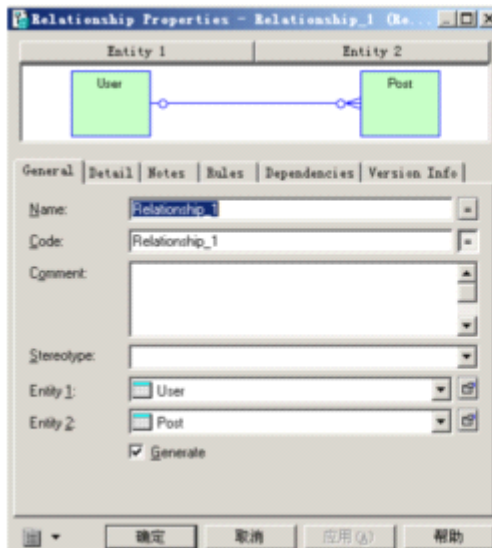
关系（Relationship）表示实体间的连接。如在一个人力资源管理系统中的 CDM 中，员工是团队中的成员，关系“Member”连接了员工（Employee）和团队（Team），这种关系表述了每个雇员在团队中工作且每个团队都由员工组成。

建立关系（Relationship）这里以用户实体（User）和帖子实体（Post）为例

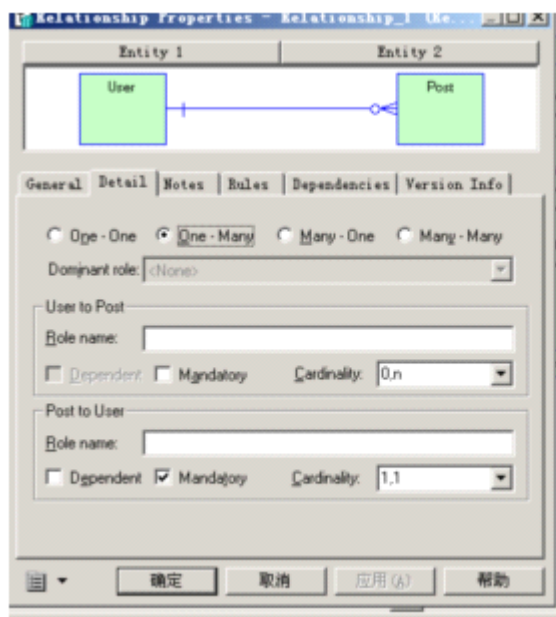
- (1) 在 Palette 面板中左键单击 Relationship 工具
- (2) 在实体 User 上单击鼠标左键，按住不放，拖拽鼠标至实体 Post 上后才松开，这样即建立了 User 和 Post 之间的 Relationship.



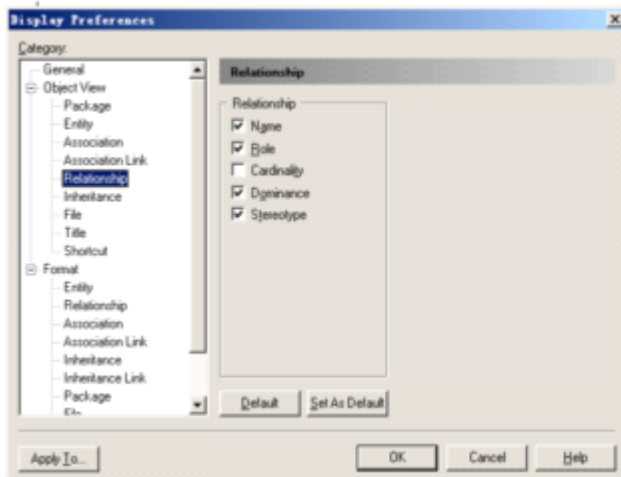
- (3) 单击鼠标右键或左键单击 Palette 面板上的 Pointer 工具，使鼠标返回至选择状态。
- (4) 双击图表中的刚建立的两实体之间关系（Relationship）以打开关系属性窗口，便于对关系进行详细定义。



- (5) 输入相应的 Name 和 Code，选择 Detail 选项，进入如下属性编辑页：



- (6) 选择 One-Many 选项，因为 User 和 Post 为”一对多”关系，且每一条 Post 均对应有 User，因此 User to Post 角色的基数（Cardinality）下拉列表中选择”0,n”,在 Post to User 角色的基数列表中选择”1,1”。同时 Role name 中输入相应的角色名称。
- (7) 确定修改后，单击 OK,即可在模型图表中显示新建的 Relationship。
- (8) 若要自定义关系显示信息，可通过选择菜单栏中 Tools->Display Preferences，打开 Display Preferences 窗口，在左边树型菜单中选择 Object->Relationship，这时即可在右侧选择你所要显示的项目了。



当然你也可以选择其它节点，实现对图符的显示属性设置。

各种类型关系（Relationship）

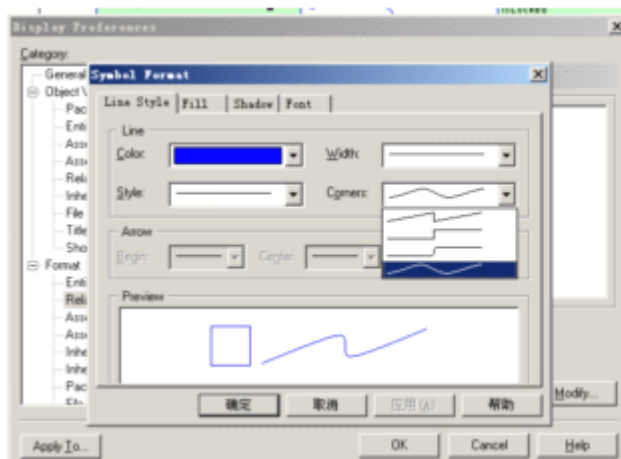
这部分是比较令人头疼的，不太好懂，需要投入较多时间研究。

自反关系（Reflexive relationship）

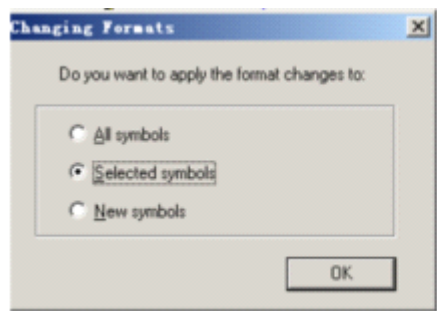
是一种实体和它自身的关系。这里用员工的管理概念来表述管理人员管理员工，同时管理人员也属于员工范畴。

- (1) 左键单击 Palette 面板中 Relationship 工具
- (2) 在实体内单击鼠标左键且按住不放，将鼠标拖放至实体旁的空白位置后松开鼠标。
- (3) 再次单击实体即成功创建自反关系。

不过这时自反关系的图符不太雅观，可以通过先选定需要更改的图符，然后选择 Display Preferences->Format，单击 Modify 以打开 Symbol Format 窗口，然后更改 Line Style 属性中的 Corners 下拉框中选项



确认修改后，最后在单击 Display Preferences 窗口的 OK 按钮后会弹出 Change Formats 选择对话框，若只要将修改应该至当前的自反图符，只需选择所选定图符（Selected symbols）即可。



依赖关系（Dependent relationship）
支配关系（Dominant relationship）
强制关系（Mandatory relationship）

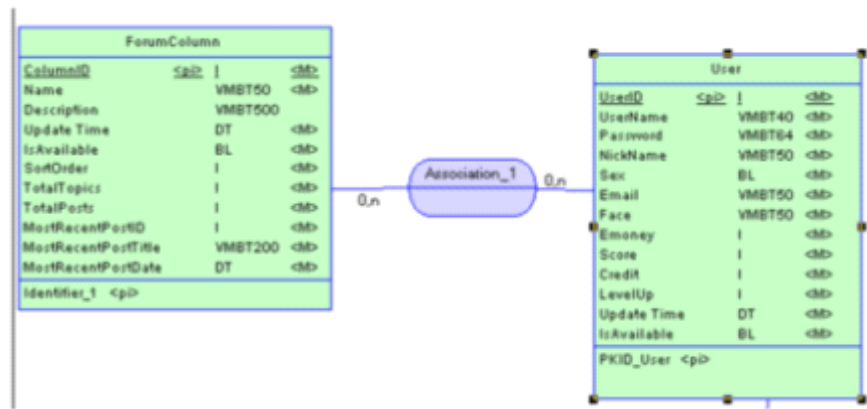
以上其它关系不再赘述，需要在实际使用过程加以运用才能加深进一步的理解，同时以上知识点和关系型数据库的理论知识密切相关，PowerDesigner 的这些功能只是对应于这些理论的一种运用映射。

关系（Association）

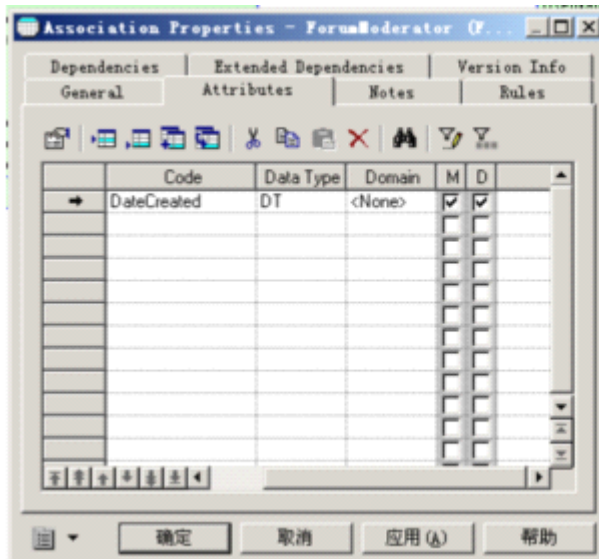
Association 也是一种实体间的连接，在 Merise 模型方法学理论中，Association 是一种用于连接分别代表明确定义的对象的不同实体，这种连接仅仅通过另一个实体不能很明确地表达，而通过”事件（Event）”连接来表示。下面通过示例论坛系统的用户实体（User）和论坛栏目（ForumColumn）实体的 Association 来讲解。示例论坛系统中通过一个 Association 来表示目标系统中论坛栏目对应的版主关系，包括了属性创建时间（DateCreated）用于记录版主添加的时间。

创建 Association

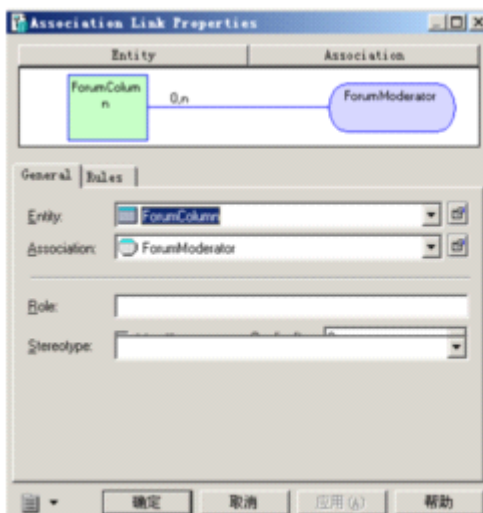
- （1）在 Palette 面板中单击 Association Link 工具
- （2）在实体 User 内单击鼠标左键且按住不放，拖放鼠标至另一实体 ForumColumn 上，松开鼠标左键，即在两实体间创建了 Association。如下图：



- （3）双击模型图表中刚创建的 Association 图符以打开 Association Properties 窗口。
- （4）输入 Association 的 Name 和 Code，选择 Attributes 属性页，添加实体属性 DateCreated，并设置相关属性，如下图：



- (5) 同时可以通过在模型图表中双击相应的 Association Link 来打开 Association Link Properties 来编辑连接属性：



按类似方法可以创建论坛栏目实体（ForumColumn）和角色实体（Role）之间的 Association。

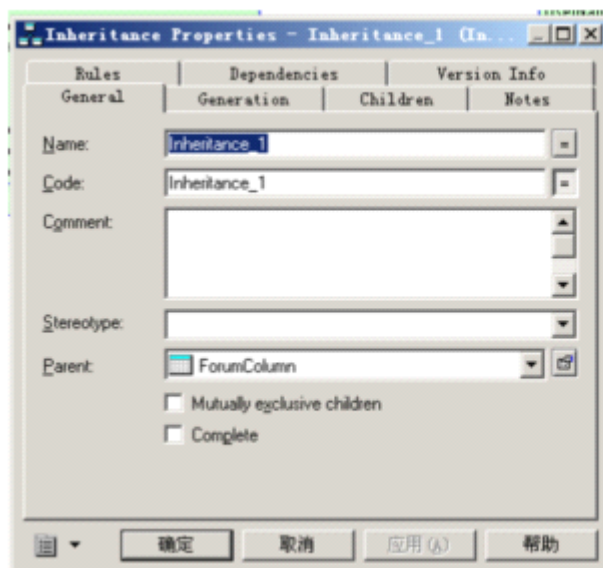
继承（Inheritance）

Inheritance 允许你定义一个实体为另一个更一般（常规）的特例。涉及到继承的实体之间有着共同相似的特征，但却是不同的。超类（或父类）指那些包含共同特征的更一般的类，而特例则被成为子类型，包含了一些更为具体和特殊的特例。

关于继承方面的例子不少，稍具有面向对象观念的都应该能够理解，不再赘述。

而 PowerDesigner 中关于继承方面的操作过程在这只作简要介绍：

- (1) 在 Palette 面板中单击 Inheritance 工具
- (2) 左键单击子类型，按住鼠标不放，拖放至鼠标至父类型实体图符中，松开鼠标，即完成了一个 Inheritance Link 的创建
- (3) 要再次添加另一子实体时，可以单击 Inheritance 工具，从半圆形图处拖动鼠标至另一子类型实体，然后松开鼠标即可。
- (4) 双击新创建的继承图符或实体之间的连接线即可打开弹出 Inheritance Properties 编辑窗口。

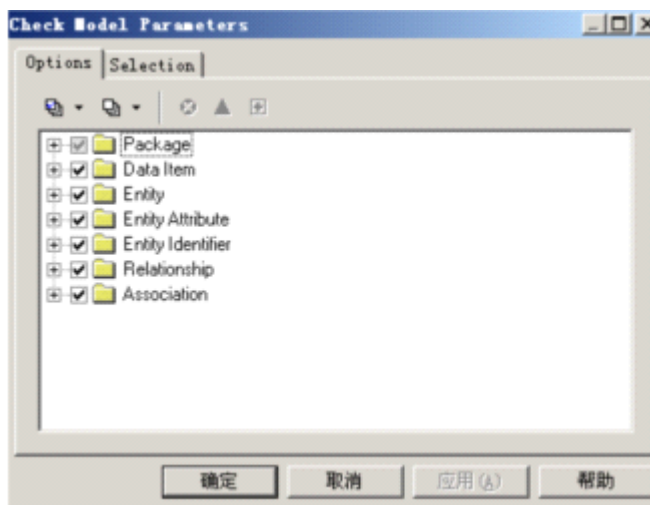


(5) 输入相应 Name 和 Code，完成基本设置，单击 OK，即完成创建过程。

现在已经基本上完成了目标系统的概念建模过程，为此下一步我们需要校验已经设计好的模型，以便于能够正确地转换为物理数据模型（PDM）。

检验模型（Check）

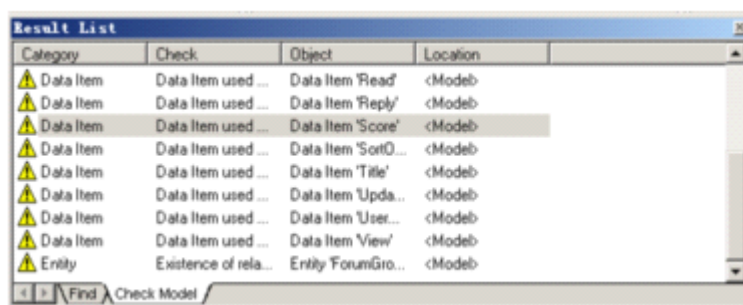
(1) 选择 Tools->Check Models，打开 Check Model Parameters 窗口，如下图：



在这你可以对需要 Check 的项目进行自定义选择。

(2) 确认选择后，单击 OK,则 PowerDesigner 开始对模型进行检验。

(3) 完成检验后，PowerDesigner 会将检验结果在输出列表中显示出来



我们可以根据所列出的错误信息对模型进行修改,错误信息分别有 Error,Warning, Automatic correction 三种,同时只要经过检验后没有 Error 一类的错误信息,我们就可以将该 CDM 转化为对应 PDM。

生成 PDM

当你从一个 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:

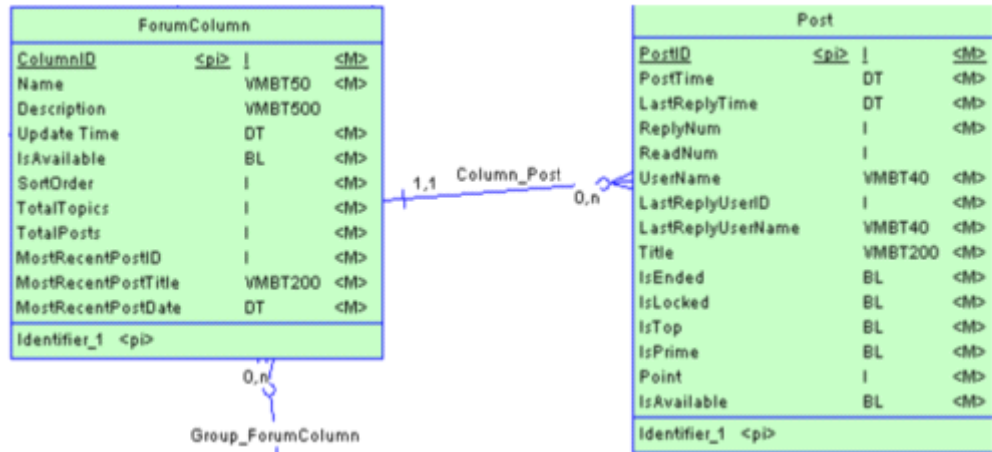
CDM 对象	在 PDM 中生成的对象	备注
实体 (Entity)	表 (Table)	
实体属性 (Entity Attribute)	列 Table Column)	
主标识符 (Primary Identifier)	根据是否为依赖关系确定是主键或外键	
标识符 (Identifier)	候选键 (Alternate key)	
关系 (Relationship)	引用 (Reference)	

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。

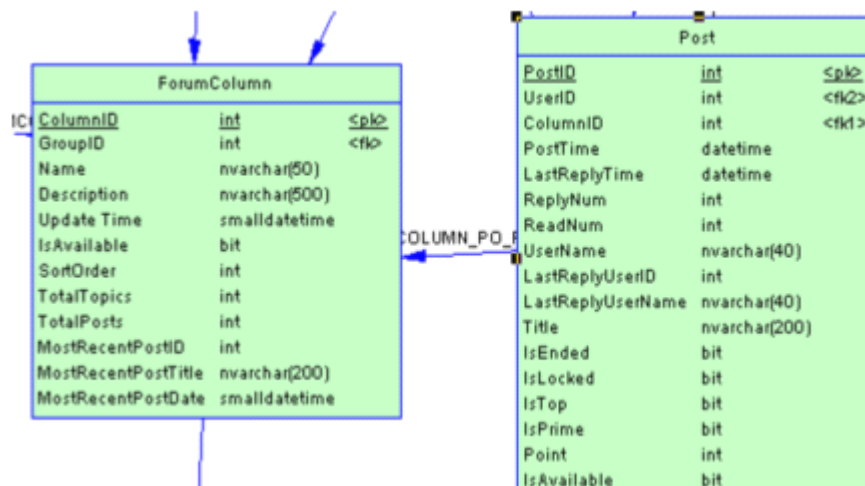
- 在 PDM 中生成的键类型取决于 CDM 中用于定义一个 Relationship 的基数和依赖类型。
1. 非依赖性一对多关系 (Independent one-to-many relationships)

- 在非依赖性关系中,“一”端的实体主标识符将转化为:
- (1) 由关系中“一 (one)”端的实体生成的表的主键 (Primary key)
 - (2) 由关系中“多 (many)”端的实体生成的表的外键 (Foreign key)。

如下图所示:



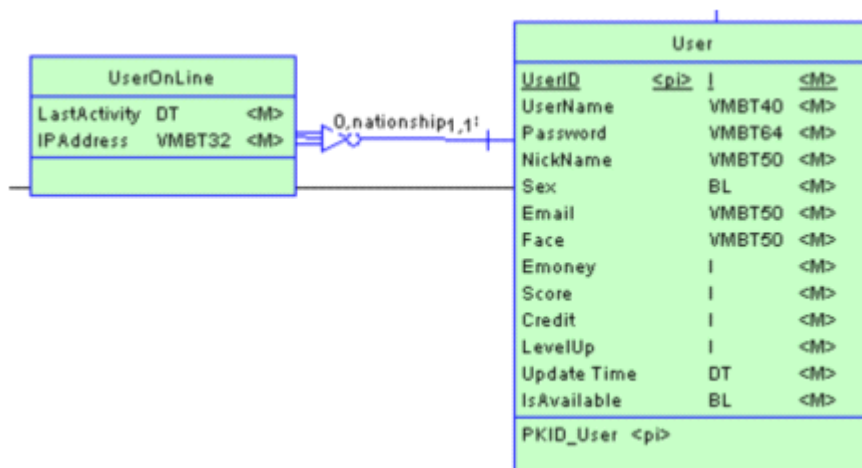
CDM 中 Independent one-to-many relationship



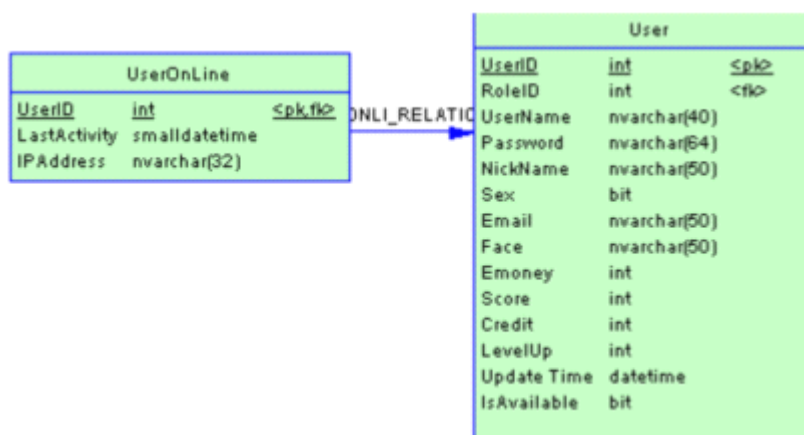
生成的 PDM 中的 Independent one-to-many relationship

2. 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。



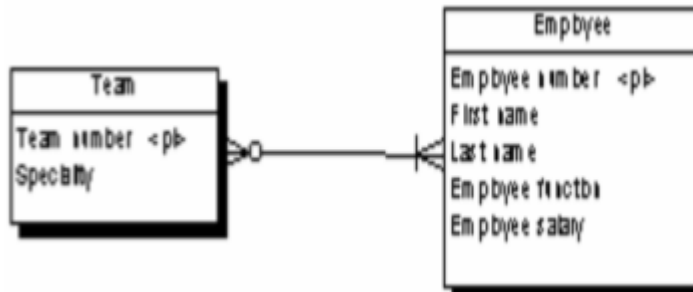
CDM 中 Dependent one-to-many relationship



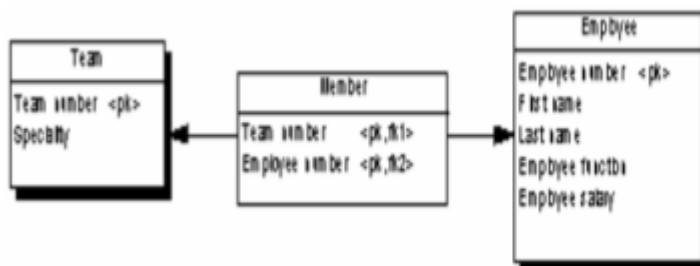
生成的 PDM 中的 Dependent one-to-many relationship

3. 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



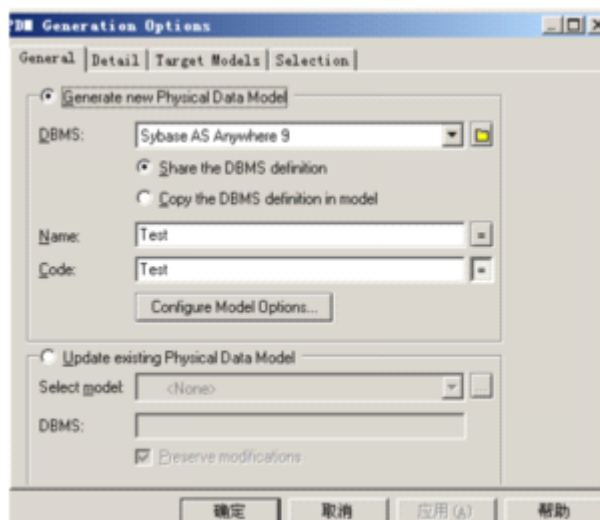
生成的 PDM 中的 Dependent one-to-many relationship

4. 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。

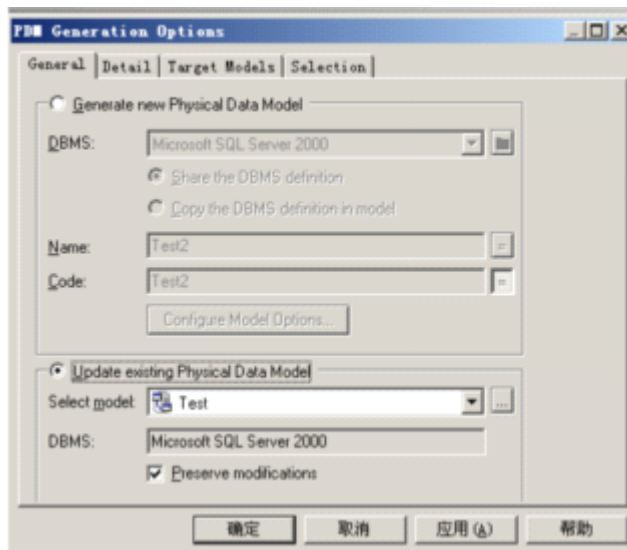
个人觉得在生成 PDM 过程中有关生成主键，外键等问题比较棘手，我自己在生成该示例论坛系统的 PDM 时就遇到这方面问题，后来在多次对一些设计得比较优秀的开源系统进行反向工程，然后慢慢研究借鉴，发现自己在设计过程的一些问题，因此觉得这方面只有多多研究才能逐渐得心应手。准备差不多了，开始生成我们需要的 PDM。

（1） 选择菜单栏上 Tools->Generate Physical Data Model 弹出 PDM Generation Options 窗口，如下图：



- (2) 选择 Generate Physical Data Model, 在 DBMS 下拉列表中选择相应的 DBMS, 输入新物理模型的 Name 和 Code.
- (3) 若单击 Configure Model Options 则进入 Model Options 窗口, 可以设置新物理模型的详细属性。
- (4) 选择 PDM Generation Options 中的 Detail 页, 设置目标 PDM 的属性细节。
- (5) 单击 Selection 页, 选择需要进行转化的对象。
- (6) 确认各项设置后, 单击确定。即生成相应的 PDM 模型。

生成 PDM 后, 我们可能还会对前面的 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。这方面我们可以参考前面生成报告 (Report) 方面的内容。

物理数据模型

PDM 基础

PDM 是用于定义详细定义物理结构和数据查询的数据库设计工具。你可以在 PDM 中使用不同类型的图表, 这取决于你所要设计的目标数据库的类型。当今关于数据库方面比较热门的话题莫过于数据仓库, 数据集市, OLAP, 数据挖掘等内容了。而 PowerDesigner 对这几方面的设计都有很好的支持, 分别支持了操作型数据库, 数据仓库或数据集市, OLAP 等类型数据库系统。相信大家都应该有所了解, 关于这几个概念就不再赘述, 本小节内容主要是涉及操作型数据库的专题。

PDM 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。你可以修改装载在 PowerDesigner 中 DBMS，对于每个你将要修改的初始 DBMS，你都可以创建一个相应的新 DBMS。

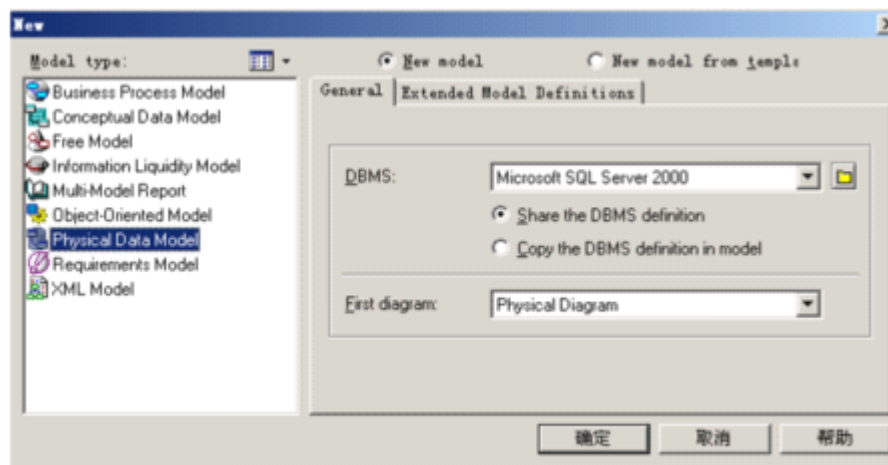
新建 PDM

你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 V6 版本的数据仓库分析模型迁移等。

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。

业务规则概念

业务规则是业务进程需要遵从的一些规则，它们可能是政府法令，客户需求或者内部的一些方针规范。业务规则通常来自于简单的观测，如“客户可以通过拨打免费热线下订单”，而在设计过程中，我们就需要将该过程分解成更加详细的描述。如当下订单时客户需要提供什么样的信息或根据客户的信用度来判定客户能够订购多少产品。

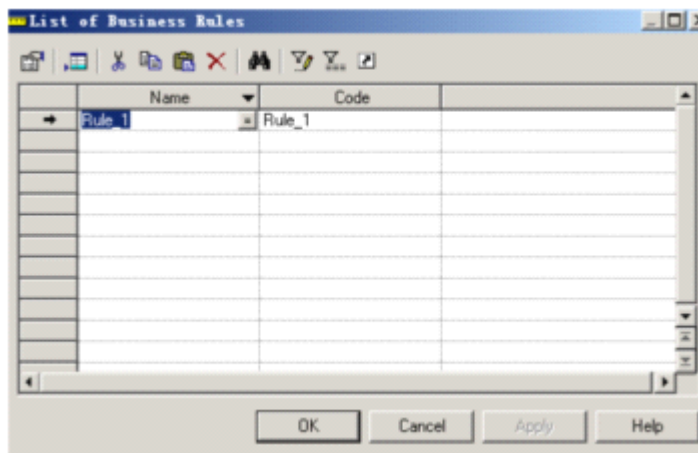
业务规则能够规划并将模型文档化。如规则“一个雇员仅属于一个部门”可以帮你图形化地在一个雇员和一个部门之间建立联系。

业务规则用一种不易用图形化表达的信息补充模型图形，如有些规则以公式或验证规则的形式来表达一些特殊的物理概念，而这些技术表达方式通常不能通过图形化形式显示出来。也可以将业务规则和 PDM 中具体对象联系起来，如果建立了验证规则与列或域之间的联系，你就可以通过业务验证规则来检查参数。

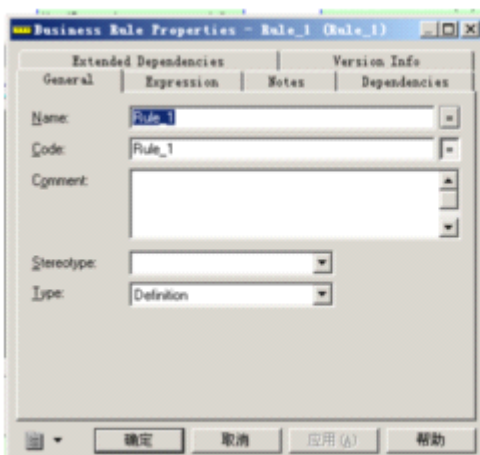
创建业务规则（Business rule）

- (1) 选择 Model->Business Rules，打开 List of Business Rules 窗口，列表显示当前模型中存在的业

务规则，如下图：



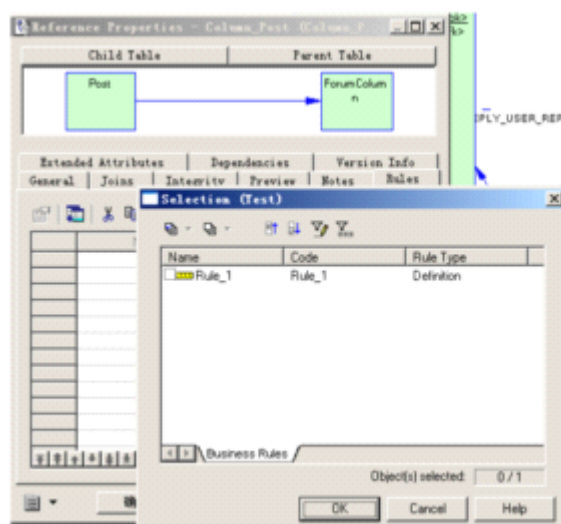
- (2) 单击工具栏中 **Add a Row** 工具或单击列表中一个空白行，即添加一个新业务规则。
- (3) 输入相应的 **Name** 和 **Code**，单击 **Apply**，提交业务规则的新建。
- (4) 双击所选择的业务规则或单击工具栏上 **Properties** 工具，打开业务规则属性（**Business Rule Properties**）窗口，如下图所示：



- (5) **Type** 下拉列表中选择相应的业务规则方式，待选类别有定义（**Definition**），事实（**Fact**），公式（**Formula**），需求（**Requirement**），验证（**Validation**），约束（**Constraint**）。但只有验证（**Validation**）和约束（**Constraint**）类型的业务规则才能生成到数据库中。
- (6) 选择 **Expression** 属性窗口，有两种类型的业务规则表达式，分别为 **Client** 和 **Server**。其中 **Server** 部分为可以生成到数据库中，而 **Client** 部分则仅用于模型文档的生成。
- (7) 设置完毕，单击“确认”，完成业务规则创建过程。

在 PDM 中应用业务规则

- (1) 在当前模型图表中双击将要应用业务规则的对象，以打开该对象属性窗口。
- (2) 选择 **Rules** 属性，列表中显示应用至该对象上的业务规则列表。
- (3) 单击工具栏中 **Add Objects** 工具以显示业务规则列表。如下图：



- (4) 选择你想要添加的应用于该对象的业务规则，单击 OK。
- (5) 在对象属性框中单击 OK，即完成业务规则应用，若添加的是约束规则或验证规则，你可以通过 Preview 选项看到业务规则生成的数据库代码。

建立物理图表（Physical Diagram）

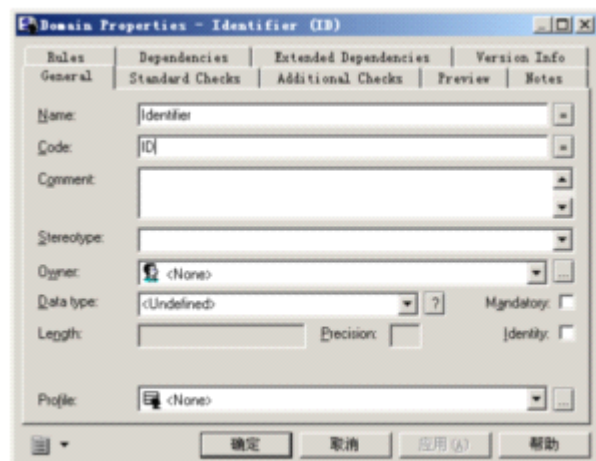
由于 PowerDesigner 中 CDM 和 PDM 的很多操作类似，因此在后面的讲解中尽量简化一些操作细节。下面以 PowerDesigner 自带的示例模型 Project Management 为例讲解，该文件位于安装目录 Sybase\PowerDesigner Trial 11\Examples 下的 project.pdm 文件。

域 (Domain)

域（Domain）可以帮助你确定模型中的信息类型。域定义了一组对列可用的数值，对列应用域可以简化对不同表中列的数据类型标准化工作。

创建域

- (1) 选择 Model->Domains 以打开域列表（List of Domains）窗口。
- (2) 单击工具栏中 Add a Row 工具，新建域。
- (3) 输入相应的 Name 和 Code，这里我们输入 Identifier 和 ID。
- (4) 单击 Apply 提交 Domain 的创建，单击工具栏中 Properties 工具以打开 Domain Properties 窗口，如下图：



(5) 选择数据类型 (Data type)，设置 Length 等属性，同时可以选择 Standard Checks 属性页以编辑详细约束。

(6) 单击 OK，确认修改。

这样就已经完成域 Identifier 的创建过程。

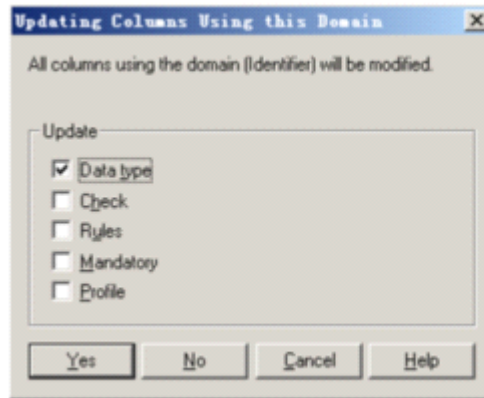
修改 Domain 属性

(1) 选择 Model->Domains 以打开域列表 (List of Domains)。

(2) 单击你想要修改的域对象，使目标域对象处于选择状态。

(3) 双击该域对象或单击工具栏中 Properties 工具以打开域属性 (Domain Properties) 窗口，现在可以对域属性进行更改了。

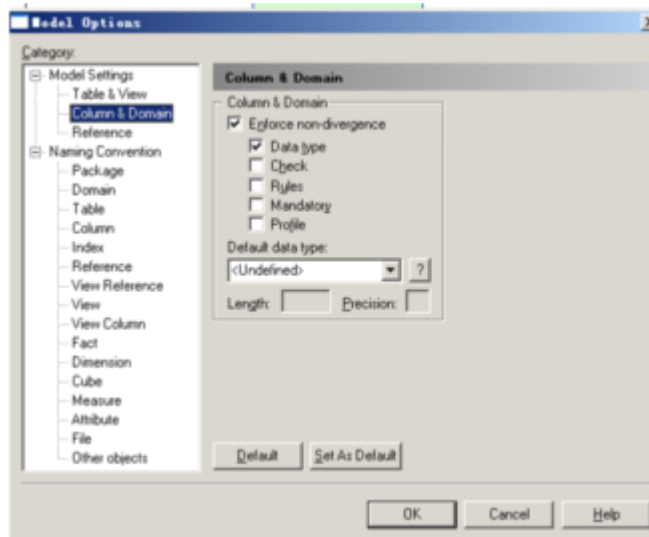
(4) 确认更改后，点击“应用”，若该域已经被某些列使用，则会弹出下列窗口：



该窗口提示询问是否想要修改应用了该域的列的域属性。你可以选择你想要更新哪些使用该 Domain 的模型元素。

(5) 单击“确认”，完成修改。

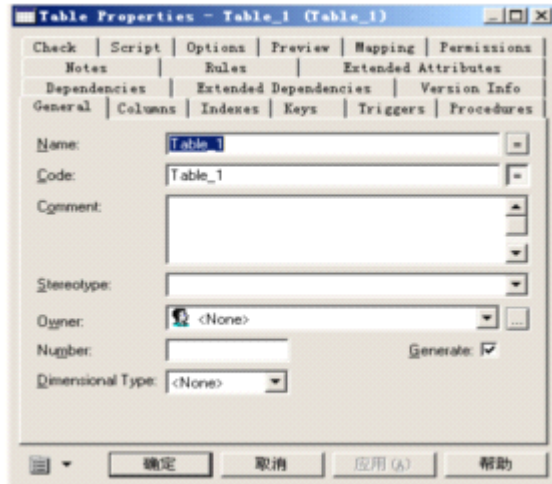
同时我们也可以设置强制执行 Domain 的修改，即在域 (Domain) 定义发生更改时，数据类型就会强制自动改变。步骤如下：选择 Tools->Model Options，打开 Model Options 窗口，选择左边树形子菜单中 Column&Domain 选项，如下图：



选择 Enforce non-divergence，再选择相应模型元素即可，这样每次 Domain 发生更改后，对应使用了该 Domain 的模型元素就自动发生更改。

创建表 (Table)

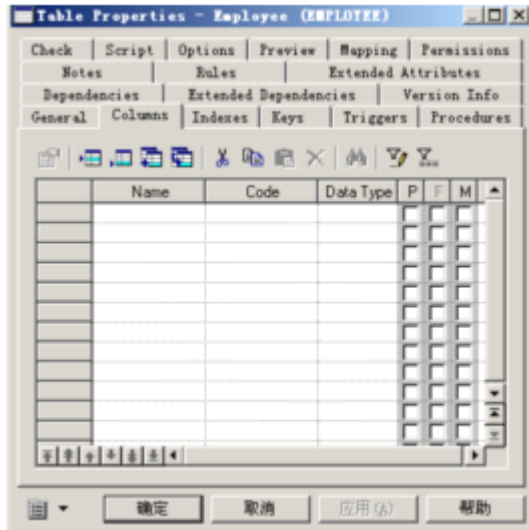
- (1) 左键单击 Palette 面板中 Table 工具
- (2) 左键单击模型图表空白区域以在模型图表中新建 Table 图符。
- (3) 单击鼠标右键或单击 Palette 面板中 Pointer 工具，使鼠标处于选择状态。
- (4) 左键双击模型图表中刚创建的 Table 图符以打开 Table 属性窗口，如下图：



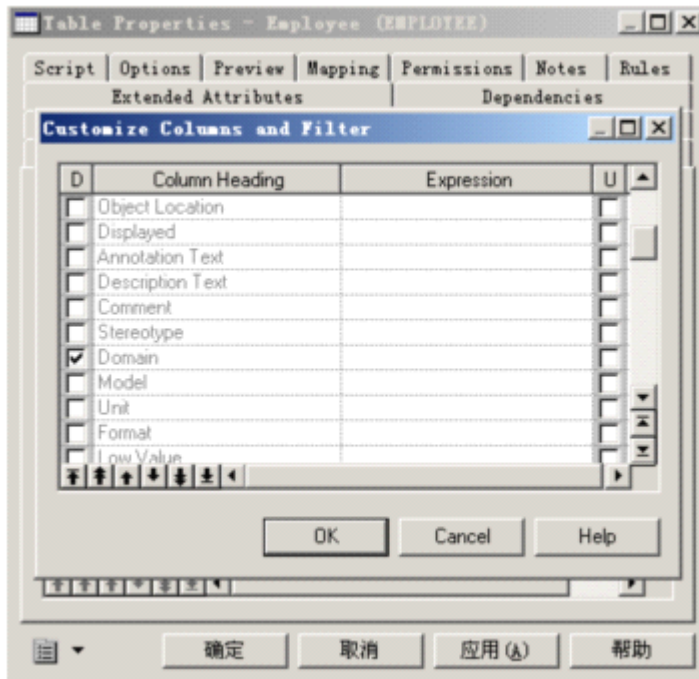
- (5) 输入相应表的名称和代码。
- (6) 其中 Number 选项为物理数据库中表的记录的大概估计，用于后述的估计数据库的大小规模；Generate 选项表示是否在物理数据库中生成该表。这里我们将 Number 设置为1000，勾选上 Generate。
- (7) 单击“确定”，即完成表 Employee 的创建。

添加编辑列

- (1) 打开表 Employee 的属性窗口，选择 Columns 属性页，如下所示：

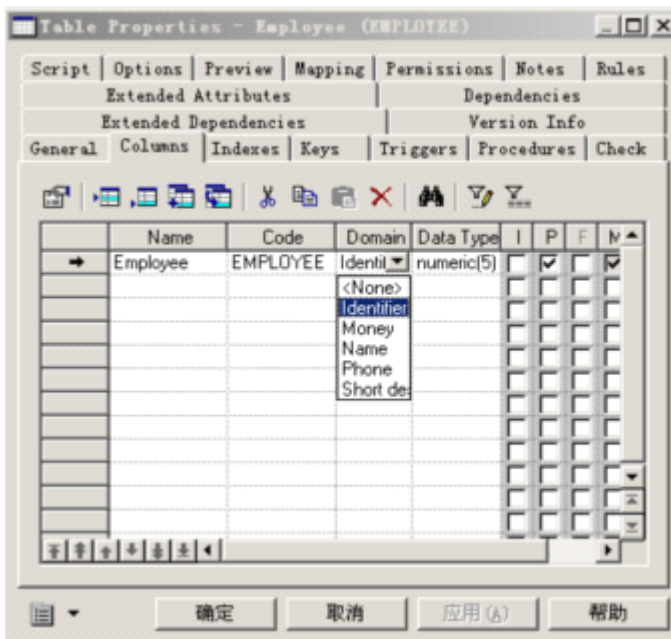


- (2) 这里我们需要使用域 Identifier，前面我们已经完成了域的创建工作，这里可以直接应用于列中。先设置 Columns 属性列表中显示 Domain，单击工具栏上 Customize Columns and Filter 工具，弹出 Customize Columns and Filter 窗口，在列表中选择 Domain，如下图：



(3) 单击 OK,则这时 Columns 属性页中显示 Domain 属性。

(4) 编辑列 Employee 属性，在 Domain 下拉列表中选择 Identifier，如下图：



(5) 单击确认，即将域（Domain）Identifier 应用至该列。

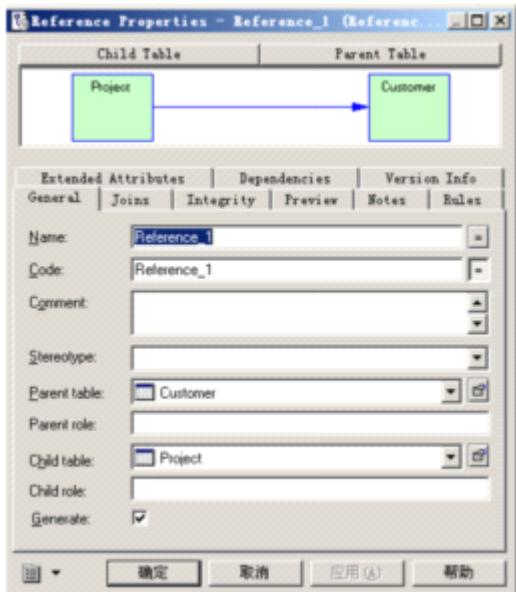
定义引用（Reference）

引用（Reference）是一个父表（parent table）和子表（child table）之间的连接，它定义了数据表各列用于主键，候选键，外键或用户指定列之间的完整性约束。当两个表中的数据列通过了引用（Reference）连接时，子表中的该列的每个值都对应了父表中对应列的一个相同的值。

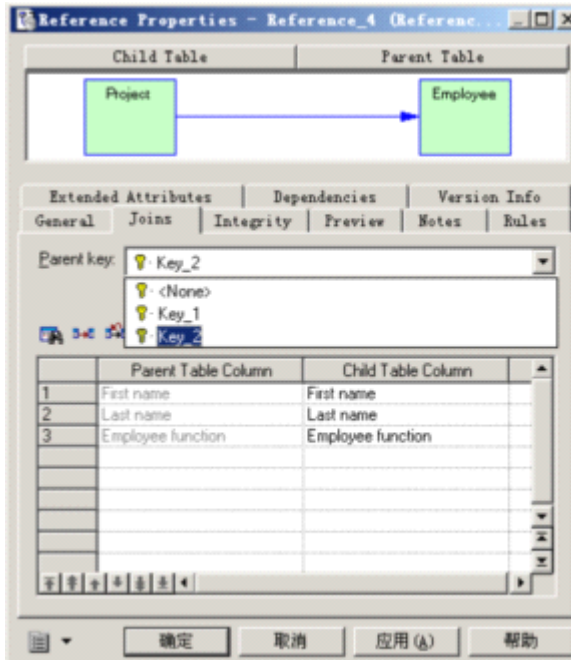
在一对引用关系中，数据列之间通过连接（Join）连接，根据在主键/候选键中列的数目，指定列的数目，一个引用关系中可能包含一个或多个连接（Join）。

建立引用 (Reference)

- (1) 选择 Palette 面板中 Reference 工具
- (2) 在模型图表区域，左键单击子表图符并按住鼠标不放，拖动鼠标至父表图符，松开鼠标，即在两表之间建立了引用关系。
- (3) 单击 Palette 面板中 Pointer 工具或单击鼠标右键使鼠标处于选择状态。
- (4) 双击模型图表中的引用 (Reference) 连接图符以打开引用属性 (Reference Properties) 窗口，如下图：



- (5) 输入相应的引用 (Reference) Name 和 Code。
- (6) 定义引用连接 (Join)，选择 Joins 属性页，如下图所示：

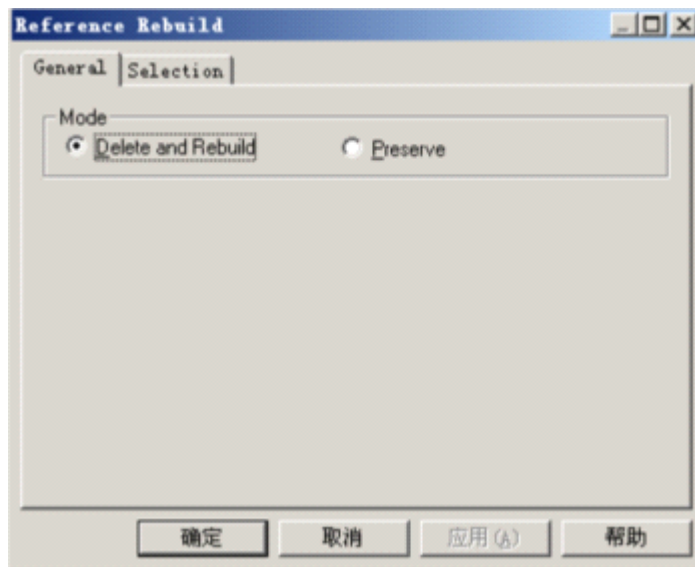


- (7) 在 Parent key 选项列表中选择相应的父表键，此时列表中显示出当前连接 (Join) 所连接的父表列和对应的子表列。
- (8) 现在可以对对应于每个父表列 (Parent Table Column) 的子表列 (Child Table Column) 进行选择更改。

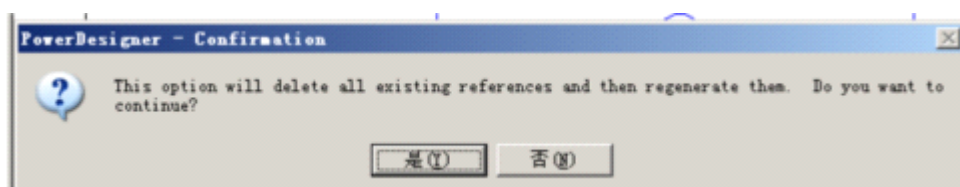
重建引用（Rebuilding references）

有时我们进行反向工程时可能不会将所有的对象都添加进去，这时可能会遇到引用冲突问题，即在已经添加进反向工程的项目中可能包含一些具有引用关系的表，而这些引用关系与没有添加进反向工程中。这时我们可以借助 PowerDesigner 提供的重建引用（Rebuilding references）功能，对引用进行选择重建。

- （1） 选择 Tools->Rebuild Objects->Rebuild References，弹出重建引用窗口。



- （2） 在 General 属性页中选择重建引用方式（其中 Delete and rebuild 方式为删除所有现有引用，再根据匹配键列创建新的引用；Perserve 方式为保留所有现有引用，且根据匹配键列建立新的引用）。
- （3） 选择 Selection 属性页，根据需要选择你要重建引用的表。
- （4） 确认选择后，单击”确认”按钮，若你选择的重建方式为 Delete and rebuild，则会弹出如下确认对话框：



- （5） 确认重建，单击”是(Y)”即可完成重建引用。

引用完整性（Referential Integrity）

引用完整性是管理数据主键，候选键和外键之间数据一致性的一系列规则，它定义了当你更新或删除父表中的一个被引用的列，或从父表中删除一条包含被引用的列的数据记录时要发生的动作。有以下两种方式实现引用完整性：

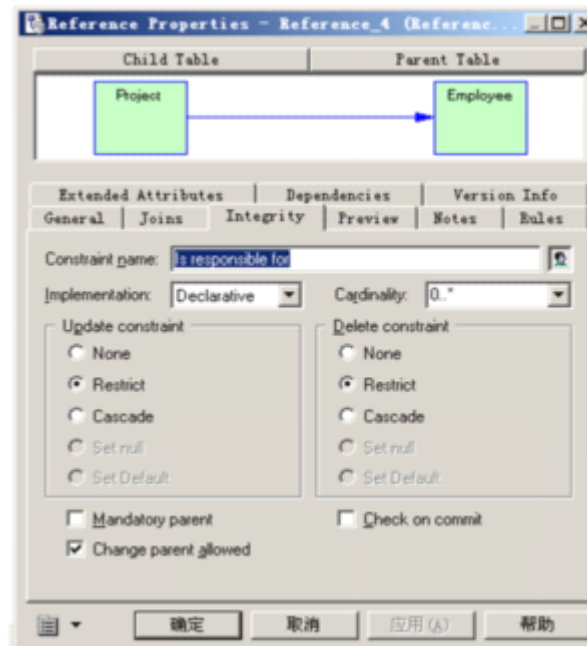
Declarative 引用完整性通过详细引用来定义，当引用生成目标 DBMS，它评估引用的正确性并生成相应的错误消息。

Using triggers 通过在引用属性窗口中定义的基于完整性约束的触发器（Trigger）来实现引用完整性约束。触发器用于衡量引用的正确性并生成适当的用户自定义错误信息。

注：对于目标数据库你可以作为生成目标数据库的选项而定义引用完整性，但并不是所有的类型的 DBMS 都支持使用引用完整性作为生成数据库的选项，此时当你生成这些类型 DBMS 的 SQL 脚本时，其中不会包含引用完整性的定义。

定义引用完整性

- (1) 双击模型图表中的引用图符以打开引用属性窗口。
- (2) 选择 **Integrity** 属性页，显示出其中用于引用完整性约束的一系列可设置选项，如下图：

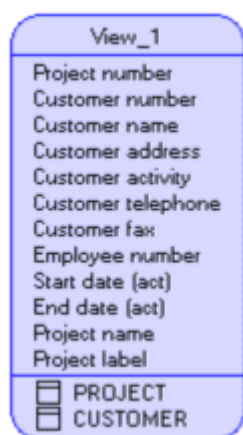


- (3) 上图中各设置项目这里就不再多说。

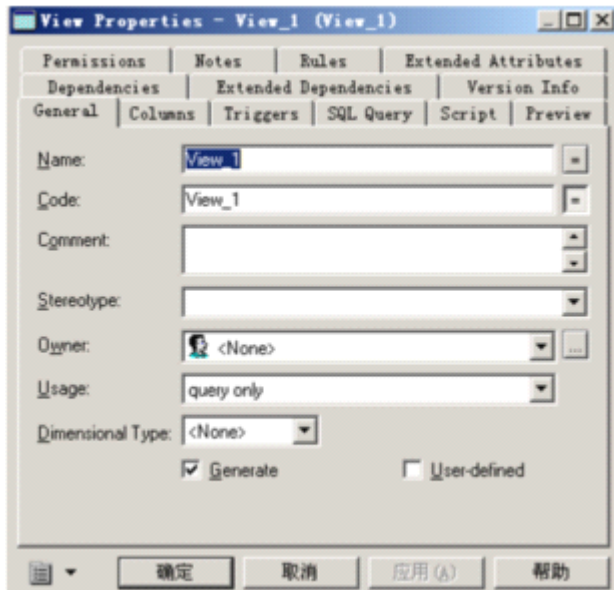
视图 (View)

为图表中已选定对象创建视图 (View)

- (1) 在模型图表中选择一个以上的表 (Table) 或视图 (View)
- (2) 选择 **Tools->Create View**，这时可以看到已经模型图表中会出现一个视图图符，其中显示出了所选定的表和视图中所有字段，如下图：



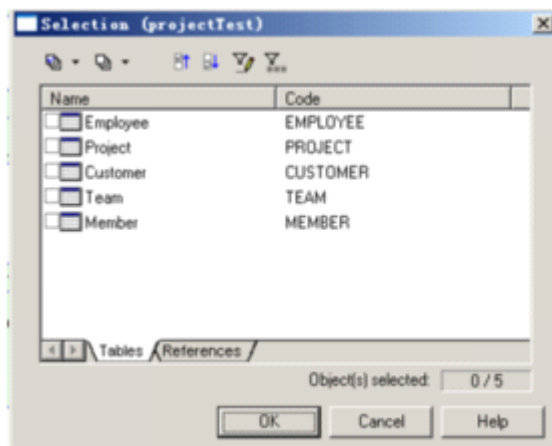
- (3) 双击刚创建的视图图符以打开视图属性 (View Properties) 窗口。



- (4) 输入相应的 Name 和 Code,同时可以通过不同属性页对视图属性进行进一步详细设计。
- (5) 单击“确认”，完成视图创建过程。

先建立空白视图，再选择所需表和视图

- (1) 选择 Tools->Create View，打开如下选择窗口：

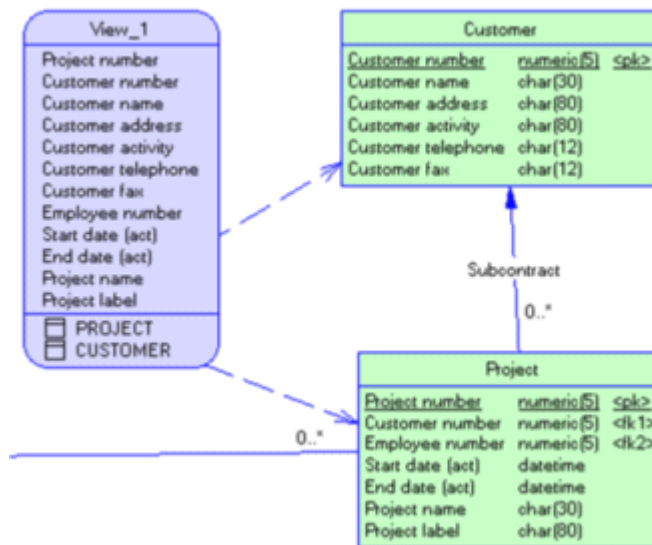


- (2) 在列表中选择你所要添加的表和视图，单击 OK，确认添加即可完成创建，这时在模型视图中出现视图图符。其它操作略。

为视图使用扩展依赖（Extended Dependencies）

扩展依赖是物理图表对象之间的连接，这些连接有助于让模型对象之间的关系更加清晰，而不会被 PowerDesigner 解释或检查，主要是用于文档化而建立的。

- (1) 选择 Palette 面板中 Link/Extented Dependency 工具
- (2) 在模型图表中先单击目标视图，拖放鼠标至其有关联的表和视图上，即将视图和表之间建立了联系，如下图所示：



注：使用扩展依赖仅仅是为了文档化模型，使对象关系更为清晰而已。

定义视图生成次序

你可以通过使用扩展依赖来定义视图的生成次序，扩展依赖是 PDM 对象之间的自由连接，这种连接能够使模型对象之间的关系更加清晰。正如前面内容所说，通常这些连接都仅仅用于文档生成而不会被 PowerDesigner 解释和检查。然而，如果指定视图之间的扩展依赖的 Stereotype 为《DBCreateAfter》时，则这些连接在生成数据库脚本时也会被解析。

如果创建一个自反，循环的扩展依赖，且其 Stereotype 方式为《DBCreateAfter》，则在检验模型时会提示错误信息。如果忽略该错误，则视图会依照字母顺序创建，如果不考虑其生成次序，数据库在生成视图时又可能会导致错误。

例如，我们为表 STORE 创建视图 DEPARTMENT STORE，其 SQL Script 为：select STORE.STOR_ID,STORE.CITY,SOTRE.STATE from STORE。同时为了仅显示公司库存数据的一部分我们需要创建在视图 DEPARTMENT STORE 基础上创建另一视图 COMPUTER COUNTER 取出其中一部分信息。

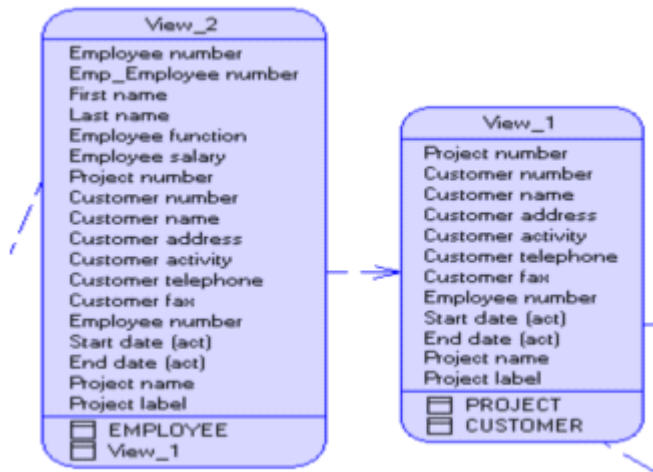
在默认情况下会按字母顺序生成视图，则 COMPUTER COUNTER 会生成失败，由于它所依赖的视图 DEPARTMENT STORE 还没有被生成。为解决这个问题，你可以在这两个视图之间创建 Steretype 方式为《DBCreateAfter》的扩展依赖，如下图：



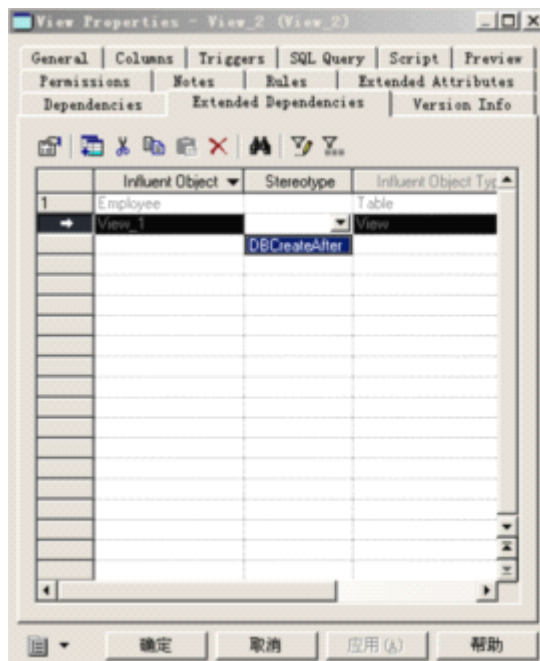
注：《DBCreateAfter》选项只有在两个视图对象之间才有效，扩展依赖所连接的两对象皆为视图（View）时才可选《DBCreateAfter》，否则为不可选状态。

步骤：

- （1）选择 Palette 面板中 Link/Extended Dependency 工具
- （2）左键单击依赖视图（Dependent View）拖动鼠标至另一视图上，松开鼠标，即在两视图之间建立了扩展依赖连接，如下图：



- (3) 双击其中的扩展依赖连接线，弹出视图属性窗口的扩展依赖属性页，列表中显示相关扩展依赖。
- (4) 在对应的扩展依赖行单击 Stereotype 列的选择箭头，选择《DBCreateAfter》即可。



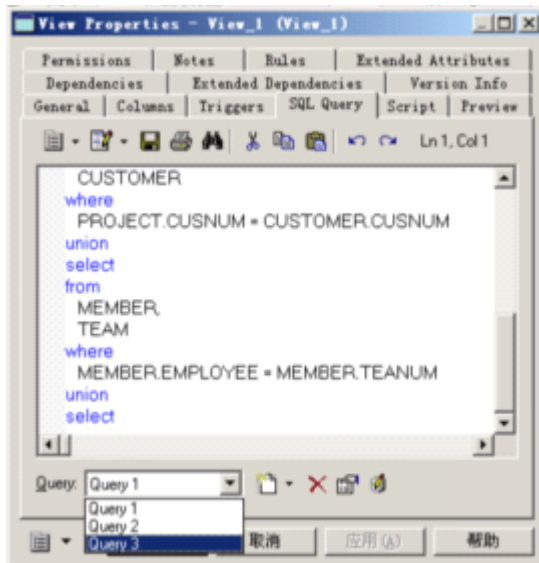
从上图中可以看到，在列表第一行的 Stereotype 下拉列表中没有《DBCreateAfter》选项，只有视图之间的扩展依赖连接才支持《DBCreateAfter》选项。

查询 (Query)

我们通过对表和视图进行查询，再通过对查询所得表进行关系运算来构建视图，PowerDesigner 中通过视图属性窗口中 SQL Query 属性页详细定义每个查询，每个 SELECT 查询都显示在 Query 的下拉列表中。其中每个视图可以有一个或多个查询组成，各查询之间通过关系运算（如 Union，Union All，Intersect 等）构成整个视图。

编辑定义查询 (Query)

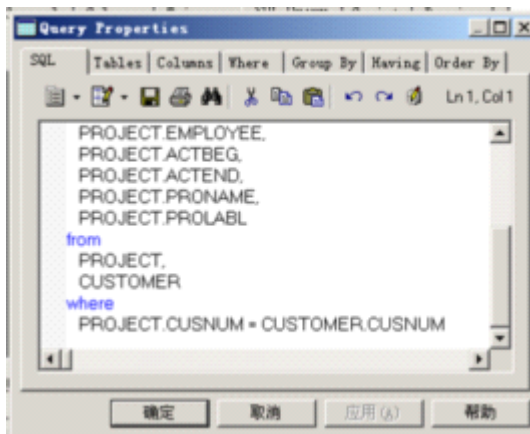
- (1) 在模型图表中双击视图图符以打开视图属性窗口。
- (2) 选择 SQL Query 属性页，如下图：



- (3) 若要新建查询，则单击 Query 下拉列表右边的 Add a Query 工具，同时也可以通过单击 Add a Query 右边的箭头选择各个查询之间的连接关系，如下图所示：

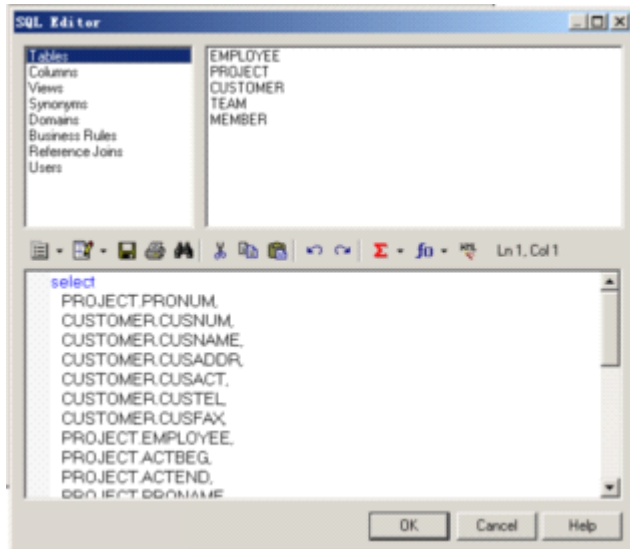


- (4) 在单击 Add a Query 工具后弹出查询属性 (Query Properties) 窗口，如下图：



这时通过选择不同的属性页可以对查询进行详细编辑。

- (5) 接 (3) 步，同时可以单击视图属性 (View Properties) 窗口中 Query 下拉列表右边的 Edit with SQL Editor 工具以打开 SQL Editor，如下图：



现在可以很轻松地对该视图的查询进行编写订制了。

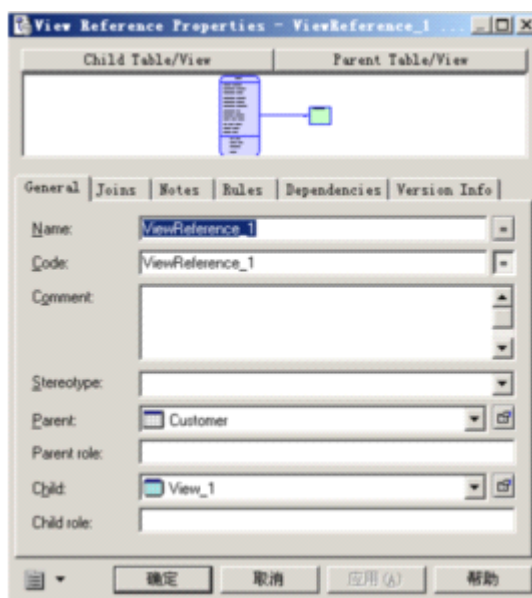
这里只是简要介绍 PowerDesigner 中关于 Query 方面的内容，实际上涉及内容非常多，不能一一列举，相信熟悉数据库操作的朋友应该能很快掌握。

在物理模型中定义视图引用（View References）

视图引用是父表 / 视图和子表 / 视图之间的连接，也包括父字段和子字段之间的连接（Join），视图引用（View Reference）可以在两个视图或视图和表之间创建，而不能用于连接两个表（Table），视图引用不会生成在目标数据库中，

创建视图引用（View Reference）

- （1） 在 Palette 面板中单击 Reference 工具。
- （2） 在模型图表中，单击子表或子视图并按住鼠标，拖动鼠标至父表或父视图区域中，松开鼠标。
- （3） 单击 Palette 面板中 Pointer 工具或单击鼠标右键，使鼠标处于选择状态。
- （4） 在模型图表中双击刚创建的引用图符以打开视图引用属性（View Reference Properties）窗口，如下图所示：



- (5) 输入 Name 和 Code 后, 选择 Joins 属性页, 现在可以根据工具栏中各工具对其中连接 (Join) 设置。
- (6) 确认设置, 单击“确定”。

触发器 (Triggers) 和存储过程 (Procedures)

注: 后述内容不再包含过多的数据库设计理论知识。

触发器 (Triggers)

触发器模板 (Trigger templates)

触发器模板是对创建触发器的一种预先定义形式, PowerDesigner 集成了针对各种所支持的 DBMS 的一系列模板,

使用触发器 (Using triggers)

触发器是一种能灵活地用于数据表管理和修改的工具, 其中还可以包含变量, 针对不同类型的 DBMS 能定义相同类型的一些触发器。

自动创建触发器

我们可以为模型中一个或多个数据表自动创建触发器 (triggers), 这些触发器的创建是针对那些定义了引用完整性的引用 (References), 以及那些值依赖于一个 Sequence 的列。

那些使用了触发器模板但是已经被修改过, 以及用户定义的触发器在重建触发器的时候都不会被创建, 所有通过 Rebuilding Triggers 重建的触发器都是根据 DBMS 和用户自定义触发器模板而创建的, 其中所有已经被选择的数据表都会被创建 INSERT 触发器, 而 UPDATE 和 DELETE 触发器则会根据相应触发器的引用完整性来确定创建。当你更改目标 DBMS 类别, 如从 Sybase 转变为 Oracle 或 IBM DB2, 这时我们定义的触发器就需要重新创建。

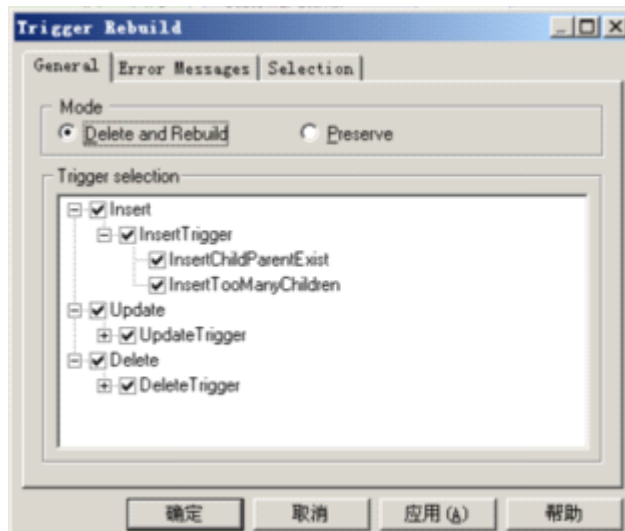
你可以选择以下两种模式重新创建触发器:

Delete 和 Rebuild 删除已经存在的触发器 (不包括用户自定义触发器) 然后根据引用完整性和数据列的 Sequence 重新创建触发器。

Preserve 保留现有用户已经定义的触发器然后根据引用完整性和数据列的 Sequence 重新创建其它触发器。

创建触发器操作步骤:

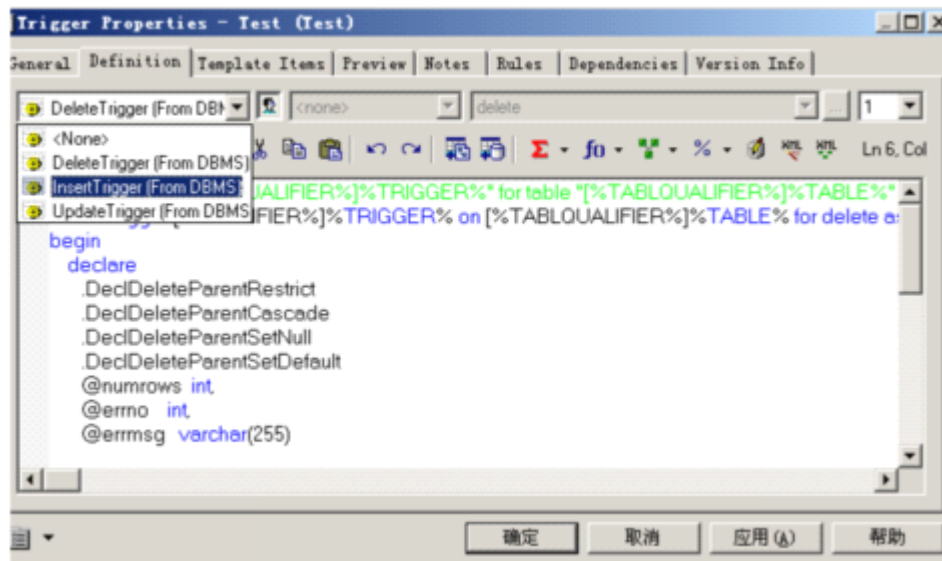
- (1) 选择 Tools->Rebuild Objects->Rebuild Triggers。即打开 Rebuild Triggers 窗口, 在 General 属性页中显示出现有触发器模板的名称。



- (2) 如果你想要删除现有的触发器再重建，请选择 Delete and Rebuild 单选框；或选择 Preserve 单选框确保重建触发器时能够保存现有触发器。
- (3) 同时在 General 属性页中选择你想要创建触发器。
- (4) 展开所选择的触发器节点。
- (5) 展开相应的触发器模板节点，其中显示出包含在该触发器模板定义中的模板项。
- (6) 选择你想要包含在目标触发器中的模板项（template item）。
- (7) 同样展开其它节点进行相同操作以选择需要包含的模板项。
- (8) 选择 Selection 属性页，会列出当前模型中的数据表（Table）。
- (9) 在列表中选择你希望创建触发器的表，单击“确认”，完成创建过程。

手动创建触发器

- (1) 在模型图形中双击你想要创建触发器的表的图符，弹出对应表属性窗口。
- (2) 选择 Triggers 属性页，显示出已经定义的触发器列表。
- (3) 单击工具栏中 Add a Row 工具，即在列表中添加了一个 Trigger。
- (4) 输入相应的 Name 和 Code，单击“应用”，提交触发器的创建。
- (5) 双击刚创建完的 Trigger 行，弹出触发器属性窗口（Trigger Properties）。
- (6) 打开 Definition 属性页，可以在模板下拉列表中选择相应的触发器模板，如下图：



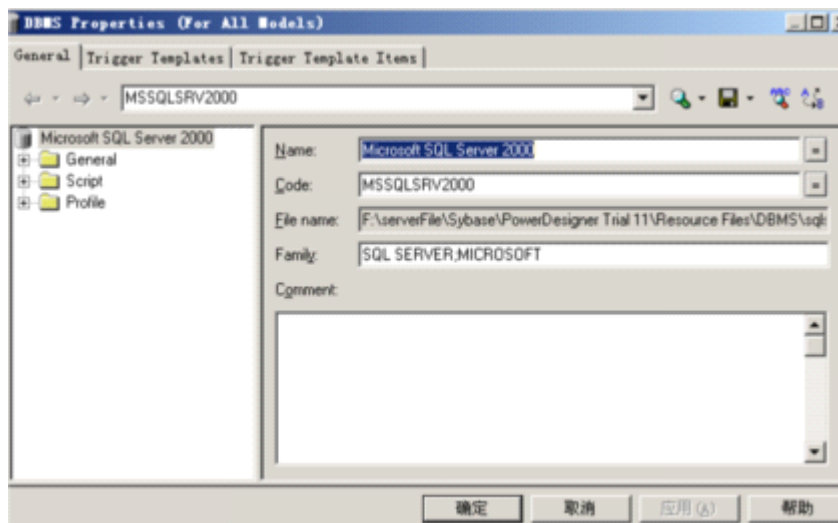
你也可以通过选择下拉列表中”None”选项创建触发器，这时不使用任何触发器模板，你需要在文本框输入详细的触发器定义。

- (7) 进行详细的触发器定义内容修改。
- (8) 在 Order 下拉列表中选择你希望该触发器执行的顺序。
- (9) 单击”确认”完成创建定义过程。

定义存储过程（Stored Procedures）与函数（Functions）

为存储过程和函数定义模板

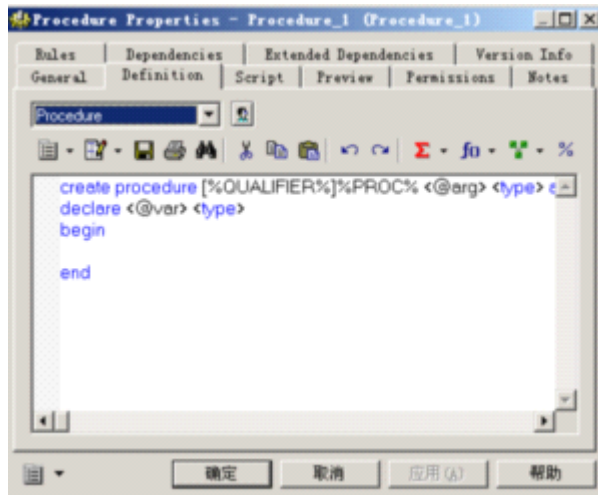
- (1) 选择 Database->Edit Current DBMS 以打开 DBMS 属性窗口，如下图示：



- (2) 展开 DBMS 树形视图中的 Script 节点，继续展开 Objects 节点。
- (3) 展开 Procedure 节点。
- (4) 单击 CustomProc 条目以编辑存储过程模板，或单击 CustomFunc 条目以编辑函数模板。
- (5) 输入你想要进行的更改，单击”确认”即完成模板定义过程。

创建存储过程和函数

- (1) 单击 Palette 面板中 Procedure 工具。
- (2) 在模型图表空白区域单击鼠标左键，即在图表中添加了一个 Procedure 图符。
- (3) 单击 Palette 面板中的 Pointer 工具或单击鼠标右键使鼠标处于选择状态。
- (4) 双击刚添加的图符，弹出存储过程或函数属性窗口，如下图：



- (5) 输入相应的 Name 和 Code 后,选择 Definition 属性页,在下拉列表中选择需要创建的类别:Procedure 或 Function。
- (6) 在文本框中输入详细的 Procedure 或 Function 定义信息,你也可以通过使用工具栏中的一些脚本项来编辑定义。
- (7) 单击“确认”,完成存储过程或函数的创建过程。

当然,您也可以同时使用菜单栏上 Model->Procedure 来完成创建过程,这里不再赘述。

将存储过程与表关联

如果当前 DBMS 支持存储过程的话,你可以使存储过程与数据表关联,该特性允许更新表或从表中读取数据。当我们将 PDM 向 OOM 转换时,与数据表关联的存储过程就会转化为所生成的类中的 Stereotype 为《procedure》的操作。通过将存储过程与数据表相关联,你可以定义所生成的类中的操作。

当我们将 OOM 转化为 PDM 时, Stereotype 为《procedure》的类操作将转化为与最终生成的表相关联的存储过程。

步骤:

- (1) 打开目标数据表的属性窗口。
- (2) 选择 Procedure 属性页。
- (3) 单击工具栏中 Add Objects 工具以打开对象选择 (Selection) 窗口。
- (4) 在现有存储过程列表中选择你需要关联该表的存储过程,单击“OK”,则相应存储过程已经添加进存储过程列表中。
- (5) 单击“确认”完成关联过程。

生成触发器和存储过程脚本

在 PowerDesigner 中,我们可以通过直接使用 ODBC 或间接使用脚本创建或修改数据库触发器和存储过程,使用脚本时将生成脚本文件,若使用 ODBC 创建或修改数据库触发器和存储过程则需要直接与 DBMS 连接。

定义存储过程的生成顺序

这方面的内容同前面讲解的关于视图创建顺序问题相似,我们可以通过使用扩展依赖 (Extended Dependency) 定义生成存储过程的次序,扩展依赖是 PDM 对象之间的自由连接,这些连接使模型对象之间的关系更加清楚。通常这些连接不会被 PowerDesigner 解释或检查,而仅仅是用于生成文档。然而,如

果你指派一个存储过程之间的扩展依赖的 Stereotype 为《DBCreateAfter》，那么在生成数据库过程中该扩展依赖也会被分析。

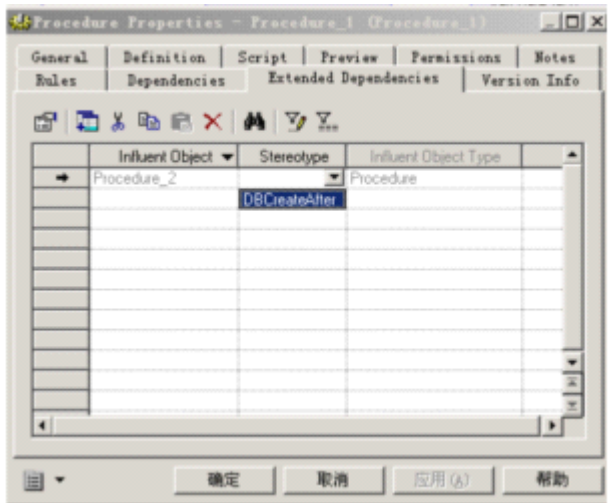
扩展依赖所起始的存储过程是依赖过程而连接终点的过程为影响（存储）过程,这时依赖过程可能使用到影响过程，所以影响过程需要在依赖过程之前生成，否则 PowerDesigner 会按照字母顺序生成脚本，则可能会导致错误。这时我们就需要在依赖过程和影响过程之间设定相应的生成顺序。

步骤：

- （1） 先需要建立存储过程之间的扩展依赖关系，单击 Palette 面板中 Link/Extended Dependency 工具。
- （2） 在模型图表中单击依赖（存储）过程图符，拖动鼠标至影响（存储）过程，松开鼠标，则建立两存储过程之间的扩展依赖关系，如下图：



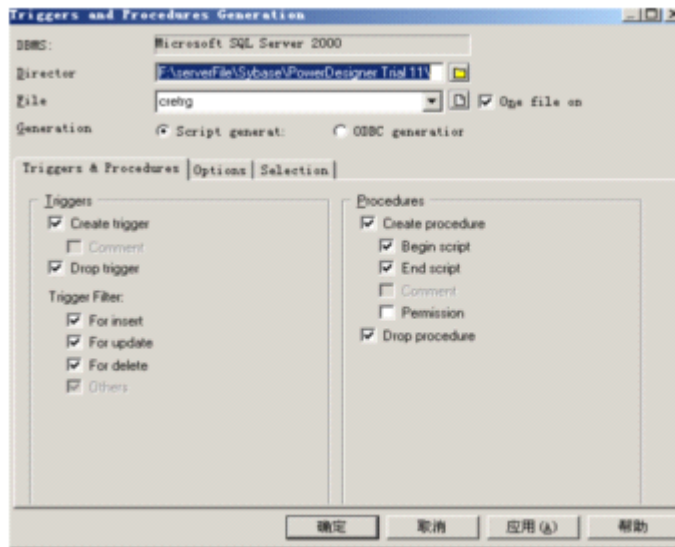
- （3） 单击 Palette 面板中的 Pointer 工具或单击鼠标右键使鼠标处于选择状态。
- （4） 在图形模型中双击依赖（存储）过程或双击扩展依赖图符以打开存储过程属性窗口。
- （5） 选择 Extended Dependencies 属性页，在 Stereotype 下拉列表中选择《DBCreateAfter》选项，如下图：



- （6） 单击”确定”，完成创建过程。

生成触发器和存储过程脚本

- （1） 选择 Database->Generate Triggers & Procedures，弹出 Triggers and Procedures Generation 属性窗口，如下图：



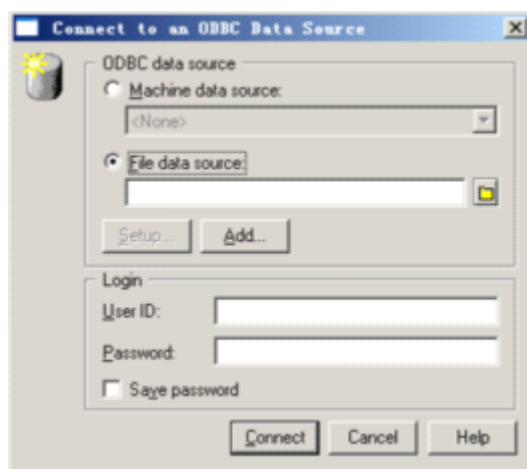
- (2) 选择相应的文件保存路径，输入对应生成的脚本名称，选择 Script Generation 选项。
- (3) 设置相应的参数。
- (4) 选择 Options 属性页，选择相应的选项。
- (5) 选择 Selection 属性页，选择需要对生成触发器和存储过程的模型；如过你需要为属于某一特殊用户的表生成触发器脚本，可以在用户下拉列表中选择相应的用户；选择需要生成触发器和存储过程的对应数据表。
- (6) 单击“确认”，PowerDesigner 随即开始生成脚本，Output 窗口会显示生成过程信息，脚本最后生成结果会在 Result 窗口中显示，其中列出脚本文件的路径。

直接在数据库中生成触发器

PowerDesigner 能够通过 ODBC 连接数据源以直接在数据库中生成触发器。

步骤：

- (1) 选择 Database->Generate Triggers & Procedures。
- (2) 在 Triggers and Procedures Generation 窗口中的 Generation 选项中选择 ODBC Generation 单选框。
- (3) 经过其它类似生成脚本的设置后，单击“确认”。
- (4) 随即弹出 Connect to an ODBC Source 对话框，如下图：



- (5) 在下拉列表中选择相应的机器数据源或选择一个数据源文件。
- (6) 输入用户 ID 和 password。

- (7) 单击”Connect”，消息窗口会显示对应的生成过程信息。
- (8) 单击”确认”，完成创建过程。

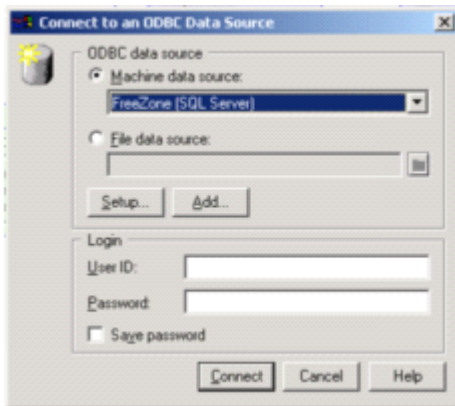
数据库的创建与修改

前面也已经接触过相关与数据库操作例子,PowerDesigner 中我们可以直接生成数据库脚本(Script),也可以直接通过 ODBC 与相关 DBMS 中数据源相连接,从而可以利用 PowerDesigner 提供的强大功能实现数据库的创建和修改操作.

通过 ODBC 操作数据库

步骤:

- (1) 选择 Database->Connect, 弹出 Connect to an ODBC Source 窗口, 如下图:



- (2) 在上面窗口中你可以对所想要连接的数据源进行设置, 如果想重新设置数据源, 可以通过单击窗口中 Setup 按钮; 若是选择 File data source 选项框, 则打开数据文件(.sql)选择窗口; 同时若需要设置数据源连接属性, 则可通过单击 Add 按钮对可选数据源进行设置, 弹出如下窗口:



- (3) 设置完毕, 单击”确定”, 返回 Connect to an ODBC source 窗口, 输入相应的登录信息, 单击”Connect”按钮, 完成 ODBC 数据源连接设置.

定制脚本

有时我们需要对所设计的数据库脚本添加一些诸如版权, 创建日期, 注释等信息, 对于一些大型系

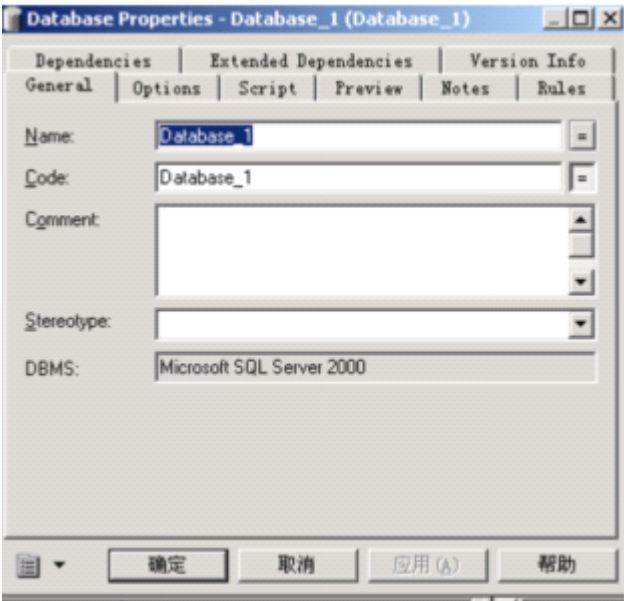
统还可能需要对相应数据表视图等对象也添加相应的信息，这也在某种程度上方便了开发成员之间的交流。
下面我们开始定制过程：

为数据库创建添加 Begin and End Scripts

（1） 选择 Model->Model Properties，打开 Model Properties 窗口，如下图：



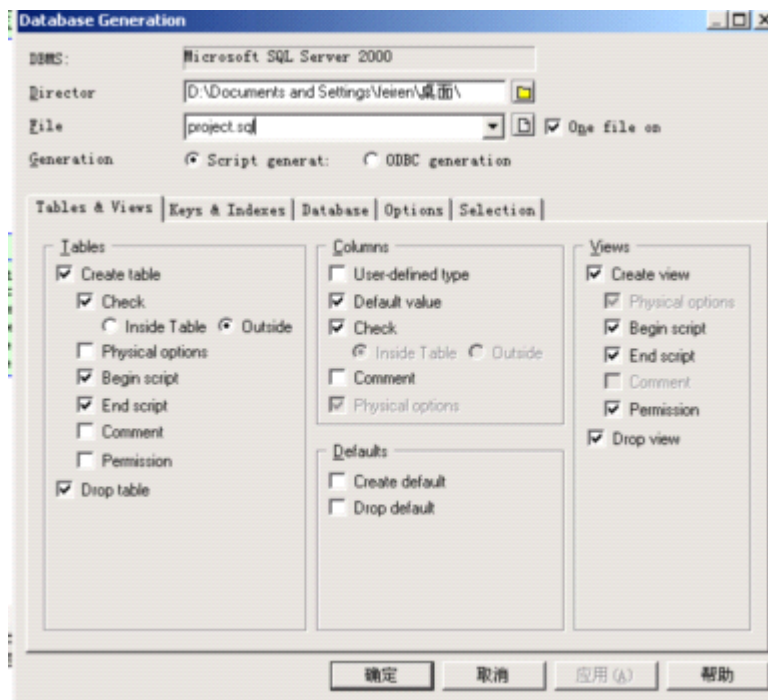
（2） 在 General 属性页中单击 Database 文本框旁的 Create 工具，弹出 Database Properties 窗口，如下图：



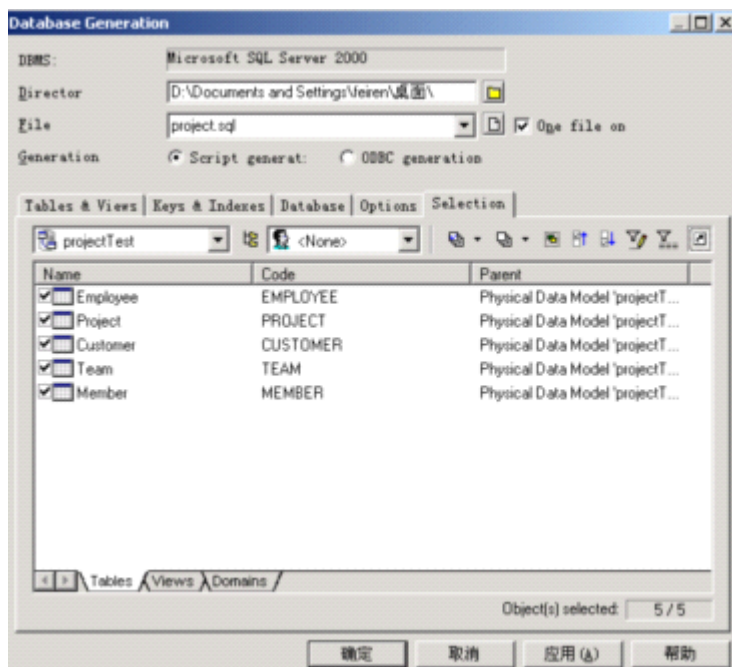
（3） 输入相应的 Name 和 Code，选择 Script 属性页，现在可以通过切换 Begin 和 End 页定制对应信息。
同理我们可以通过打开数据表的属性窗口—>选择”Script”属性页为数据表插入 Begin and End Scripts，具体操作不再赘述。

生成 SQL 脚本

（1） 选择 Database->Generate Database，弹出 Database Generation 窗口，其中包含生成数据库的各种参数选项，如下图：



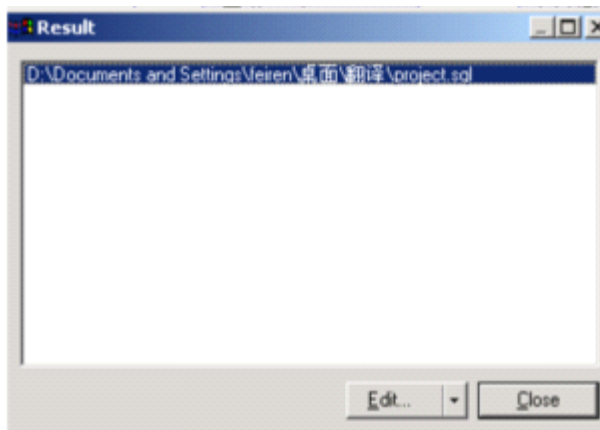
- (2) 选择相应的脚本文件存放目录，并输入相应的脚本文件名称。
- (3) 在 Generation 选项栏中选择 Script general 单选框，确认生成数据库方式为直接生成脚本文件。
- (4) 勾选上 One file on，表示所生成脚本将包含于一个文件中，否则 PowerDesigner 会为生成的每个不同表格都单独生成一个脚本文件。
- (5) 调整设置当前 Tables & Views 属性页中各选项参数。
- (6) 同样通过选择不同属性页分别设置 Keys & Indexes, Database, Options 等生成脚本参数。
- (7) 选择 Selection 属性页，如下图：



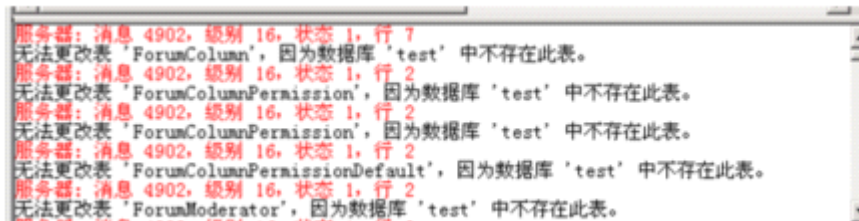
- (8) 在对象列表中选择需要生成脚本的对象。
- (9) 单击“确定”，完成生成脚本配置过程。

PowerDesigner 开始执行脚本生成过程，这时输出窗口会显示相应的生成过程信息，最后弹出 Result 窗口，

如下图:



这时让我们利用已经生成的脚本文件来创建数据库，这里我们使用的 DBMS 是 MS SQLServer2000。现在我们新建一数据库 FreeZone，在查询分析器中打开当前新建的 FreeZone 数据库下，执行刚生成的脚本文件，发现查询分析器输出窗口报告以下错误信息：



而在左边树型资源窗口中刷新，发现已经成功创建了相应的数据库对象。让我们回头看看所生成的脚本文件内容：

```
/* DBMS name:      Microsoft SQL Server 2000 */
/* Created on:     2006-8-31 23:19:14 */
/*=====*/

alter table ForumColumn
    drop constraint FK_FORUMCOL_GROUP_FOR_FORUMGRO
go

alter table ForumColumnPermission
    drop constraint FK_FORUMCOL_FORUMCOLU_ROLE
go

alter table ForumColumnPermission
    drop constraint FK_FORUMCOL_FORUMCOLU_FORUMCOL
go

alter table ForumColumnPermissionDefault
    drop constraint FK_FORUMCOL_RELATIONS_ROLE
go

alter table ForumModerator
    drop constraint FK_FORUMMOD_FORUMMODE_USER
go

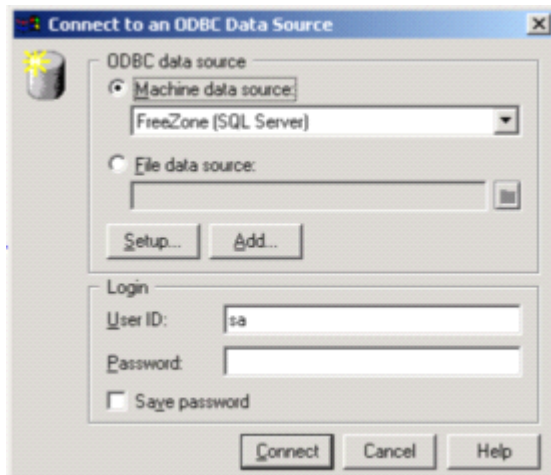
alter table ForumModerator
    drop constraint FK_FORUMMOD_FORUMMODE_FORUMCOL
go
```

这时应该差不多清楚了，所生成脚本中先删除所有的外键，同时检查原数据库中是否存在相同对象，若存在则将其删除，所以当我们第一次执行脚本时数据库中并不存在相应数据库对象，才会报告“不存在.....”等错误信息，同时每个执行过程后都加“go”，确保某段代码没能成功执行时不会影响到后续代码

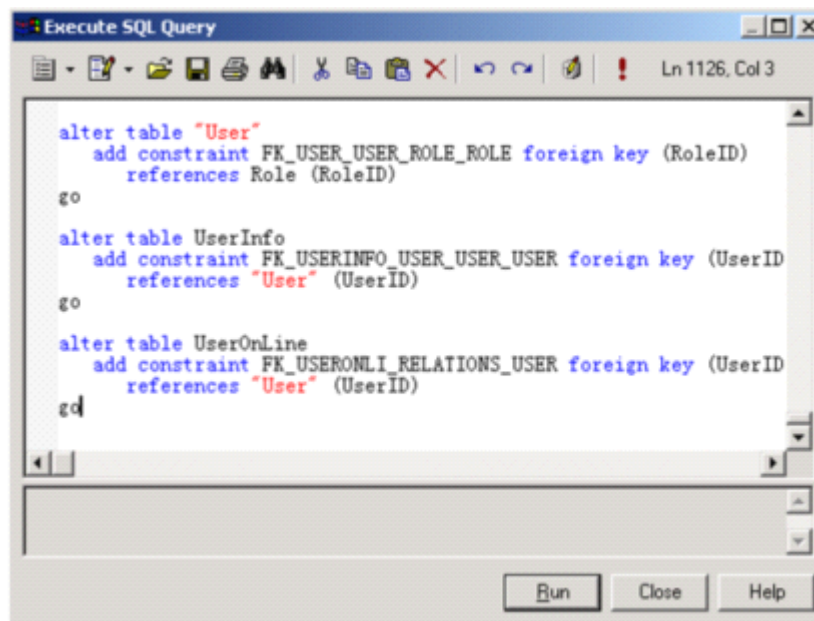
的执行。

直接创建数据库（使用 ODBC）

- （1） 选择 Database->Generate Database.
- （2） 选择相应的脚本存储目录，输入相应脚本名称.
- （3） 在 Generation 选项栏中选择 ODBC generation 单选框.
- （4） 后续其它参数设置同前面生成脚本文件类似，不再赘述.
- （5） 单击“确定”按钮，会弹出 Connect to an ODBC Data Source 窗口，如下图：



- （6） 选中 Machine data source 单选框后在下拉列表中选择对应的数据源；或选中 File data source 并选取对应数据库文件.
- （7） 输入对应用户登录信息，单击“Connect”，确认连接，PowerDesigner 随即开始脚本生成进程，完毕后弹出 Execute SQL Query 窗口，如下图：



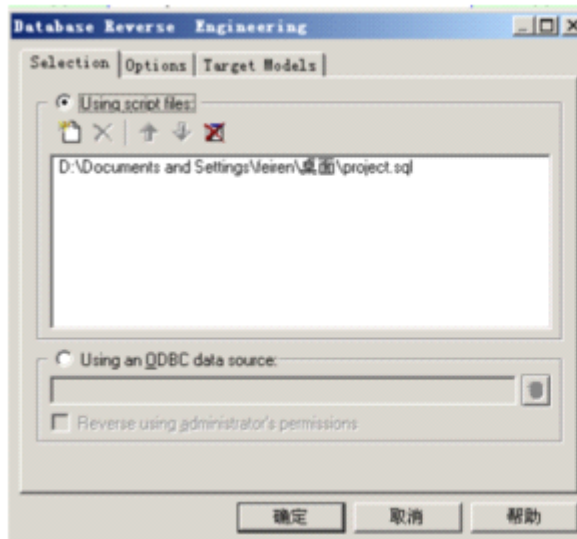
- （8） 单击“Run”，即开始数据库创建过程.

在 PDM 中进行反向工程 (Reverse Engineering)

从脚本文件反向工程数据对象

步骤:

- (1) 新建 PDM, 这里示例演示选择的 DBMS 为 MS SQLServer2000。
- (2) 选择 Database->Reverse Engineer Database, 弹出 Database Reverse Engineering 窗口, 如下图所示:

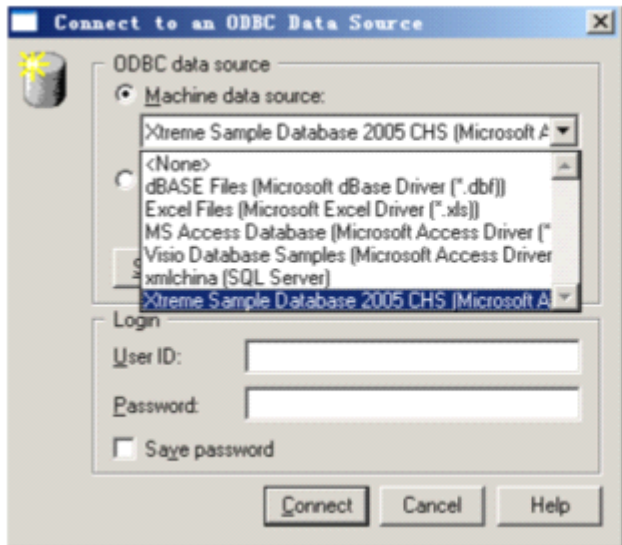


- (3) 选择 Using script files 单选框, 即为从脚本文件执行反向工程。
- (4) 单击工具栏上 Add Files 工具, 选择要进行反向工程的脚本文件; 同时你也可以添加多个脚本文件, 但需要注意的是: 应该将触发器或存储过程脚本文件放在数据表脚本文件后面, 因为反向工程时需要先执行数据表脚本。因此需要正确调整各种脚本文件顺序, 否则可能导致触发器不能正确地进行反向工程。
- (5) 在 Options 属性页中设置反向工程选项。
- (6) 单击“确定”, 完成创建过程。

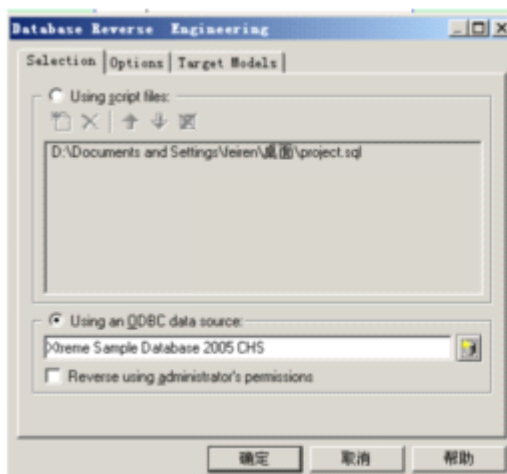
这时会弹出进度窗口显示当前反向工程进度, 完成后会在当前 PDM 中产生相应的模型对象。

从 ODBC 数据源反向工程

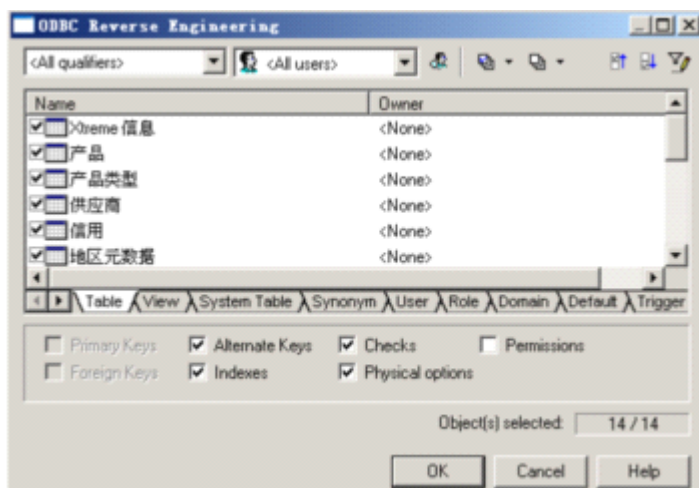
- (1) 新建 PDM, 这里也将 DBMS 设置为 MS SQLServer2000。
- (2) 选择 Database->Reverse Engineer Database, 弹出 Database Reverse Engineering 窗口。
- (3) 这时我们不再选择 Using script files, 而是选择 Using an ODBC data source 单选框, 即设置是从 ODBC 数据源进行反向工程。
- (4) 单击 Connect to an ODBC Source 工具以显示 Connect to an ODBC Source 对话框, 如下图:



- (5) 单击 Machine data source 单选框，从数据源下拉列表中选择相应要进行反向工程的数据源；或者选择 File data source 单选框，通过 Select a File DSN 工具以在相应目录选择相应 DSN 文件。
- (6) 输入对应用户 ID 和密码。
- (7) 单击 Connect，这样所选定的 ODBC 数据源就会在 Database Reverse Engineering 窗口中，如下图：



- (8) 可以选择 Reverse using administrator's permissions 选项框以使用数据库管理员身份进行反向工程。
- (9) 接下来通过选择 Options 和 Target Models 属性设置反向工程选项。
- (1 0) 单击“确定”，即开始执行反向工程过程，弹出 ODBC Reverse Engineering 窗口，其中列出相应的对象，其中只有数据表和触发器是被默认选中的，如下图：



(11) 同时还可以通过工具上一些快捷工具对其中要进行反向工程的数据库对象进行订制。

(12) 选择你想要添加进反向工程的数据库对象，可切换不同的 Tab 页进行选择，确认选择后，单击“OK”，完成创建设置。

可以看到 PowerDesigner 已经开始反向工程过程，同时输出窗口会显示出当前进程信息。

同时若当前 PDM 中已经存在模型对象，这时进行反向工程，则可以根据需要进行模型合并，覆盖等操作。

总结

同时需要提几点：PowerDesigner 提供的反向工程功能（Visio 也提供了类似功能）很不错，如当我们研究一较大系统的成型代码时，首先需要了解其中设计方案，这时可以先对其数据库进行反向工程生成对应的 PDM；反向工程也使异构数据库之间的转化更为简单。同时为了更快捷的学习 PowerDesigner，也可进行各种模型的转化，从而更快掌握 PowerDesigner 的强大功能。

以上只是简要介绍了 PowerDesigner 的部分功能，其它如水平分割，垂直分割，合并等方面数据库优化内容就不再罗嗦了，PowerDesigner 只是为提供的一种使分析设计人员更加便捷的工具，实际需要设计出良好的数据库系统好需要用户掌握扎实的相关理论知识，关系代数，元组演算，域演算等；同时需要具备相关规范化理论方面的基础知识（范式，函数依赖，属性闭包等），同时针对不同的目标系统灵活选用不同的设计方案！