数据库实验指导书

实验四：一个小型系统的设计与实现

#### 2024 秋

目录

目录

[实验四：一个小型系统的设计与实现 1](#_Toc182386664)

**[1](#_Toc182386665)** [实验目的 3](#_Toc182386665)

**[2](#_Toc182386666)** [实验环境 3](#_Toc182386666)

**[3](#_Toc182386667)** [实验内容 3](#_Toc182386667)

[3.1 实验任务 3](#_Toc182386668)

[3.2 实验要求 3](#_Toc182386669)

[3.3 选题说明 3](#_Toc182386670)

**[4](#_Toc182386671)** [数据库设计 5](#_Toc182386671)

[4.1 绘制 CDM 5](#_Toc182386672)

[4.2 CDM 转成 LDM 17](#_Toc182386673)

[4.3 LDM 转成 PDM 20](#_Toc182386674)

[4.4 在 PDM 上创建视图、索引、触发器、存储过程、存储函数 22](#_Toc182386675)

[4.4.1 创建视图 22](#_Toc182386676)

[4.4.2 定义索引 24](#_Toc182386677)

[4.4.3 定义触发器 28](#_Toc182386678)

[4.4.4 定义存储过程或存储函数 30](#_Toc182386679)

[4.5 PDM 转成 SQL 脚本 33](#_Toc182386680)

**[5](#_Toc182386681)** [常见问题 42](#_Toc182386681)

**[6](#_Toc182386682)** [参考资料 42](#_Toc182386682)

# 实验目的

1、掌握数据库设计方法；

2、了解概念模型、逻辑数据模型和物理数据模型之间的关系和不同；

3、掌握使用高级语言访问、操作数据库，加深对前后台数据交互的理解。

# 实验环境

Windows 操作系统、MySQL8.0 关系数据库管理系统、数据库设计工具。本次实验使用 C\C++，JAVA，Python，PHP 或其他语言均可。

# 实验内容

## 实验任务

1、 确定选题，并进行需求分析

2、 完成一个小型系统的数据库设计，绘制 E-R 图；

3、 将 E-R 图转成逻辑数据模型和物理数据模型，导出 sql 脚本，创建数据库；

4、用高级语言实现一个小型数据库应用系统

## 实验要求

1、 E-R 图至少包括 8 个实体和 7 个联系；

2、 需要考虑关系完整性约束：主键约束、外键约束、空值约束；

3、 设计至少 1 个视图、1 个索引（非主键、非外键索引）、1 个触发器、1个 存储过程或存储函数（视图、索引、触发器、存储过程或存储函数可以直接在DBMS 中创建，不一定要在 pdm 图的时候创建）；

4、 数据库使用 MySQL；

5、 编程语言不限；

6、 考察重点是数据库设计、前后台数据交互。图形界面是必须的，以简单明了为上。

## 选题说明

从以下 4 个题目中选 1 个：

注意：做数据库设计时要考虑所有的基本功能，在用高级语言实现一个小型数据库应用时 至少要完成红色字体的功能。

|  |  |
| --- | --- |
| 题目 | 基本功能 |
| 特色时令产品集市平台 | 1．用户注册（管理员、普通用户）；  2．普通用户可在特色时令产品平台发布和下架特色时令产品（特产品类别、特产品名、特产品描述、产地、销售价格、销售期……）；  3．普通用户可查询浏览特产品，下订单交易；  4．管理员浏览特产品，下架违规特产品；  5．普通用户可对某特产品进行收藏、添加购物车；  6．普通用户可对某单交易进行投诉（选择投诉类别，填入投诉原因）；  7．管理员可处理投诉（输入处理意见）；  8.对于多次违规的用户管理员可冻结其账号。 |
| 校园失物招寻平台 | 1. 用户注册（管理员、普通用户）；用户基本信息的增删改查（证件号、姓名、联系方式……）； 2. 普通用户可在平台发布丢失或拾取物品（失/拾物品编号、物品类别、物品名、物品描述、失/拾地点、失/拾时间、失/拾主证件号……）； 3. 普通用户可查询浏览丢失或拾取物品，发起物品领回单。 4. 管理员浏览领回单，将冒领者加入冒领失信名单； 5. 普通用户可对某个领回单进行投诉（选择投诉类别，填入投诉原因）； 6. 管理员可处理投诉（输入处理意见）； 7. 对于多违规和被投诉的用户，管理员可将其加入失信名单，冻结其账号。 |
| 校园食堂代买餐平台 | 1. 用户注册（管理员、普通用户）；用户基本信息的增删改查（校园卡号、姓名、学院、年级、联系方式……）； 2. 普通用户可在平台发布代买餐需求或服务（代买餐订单编号、类别（早/中/晚餐）、餐厅描述、餐品描述、买/代主姓名、买/代主校园卡号、买/代主联系方式、买/代主位置、买/代小费……）； 3. 普通用户可查询浏览代买餐需求或服务，下/接订单和修改订单状态（选择某个食堂、某个商家的某些菜品， 设置购买数量和总金额，设置小费， 设置外送地址， 设置联系方式……）。 4. 普通用户可对某个代买餐订单进行投诉（选择投诉类别，填入投诉原因）； 5. 管理员可处理投诉（输入处理意见）； 6. 对于多违规和被投诉的用户，管理员可将其加入失信名单，冻结其账号。 |
| 校园活动征招平台 | 1、用户注册（普通用户、管理员）；  2、管理员新增、修改、删除活动信息（时间、地点、人数、要求……）；  3、普通用户浏览活动信息和申请参加活动；  4、管理员可查询活动的申请人清单，并对申请人进行审核（通过或拒绝）；  5、普通用户能看到自己提交的申请是待审核、通过还是拒绝状态；  6、过期的活动系统自动设置为关闭状态。 |

# 数据库设计

PowerDesigner 是一款由 SAP 提供的企业级数据建模工具，广泛应用于数据库设计、数据仓库、企业架构等领域。它支持多个数据库平台，包括 MySQL、Oracle、SQL Server 等，是数据库设计师、架构师和开发人员常用的工具之一。

**PowerDesigner** 有 **CDM、LDM** 和 **PDM** 三个层次的数据模型，帮助从业务需求到实际数据库实现的设计过程。

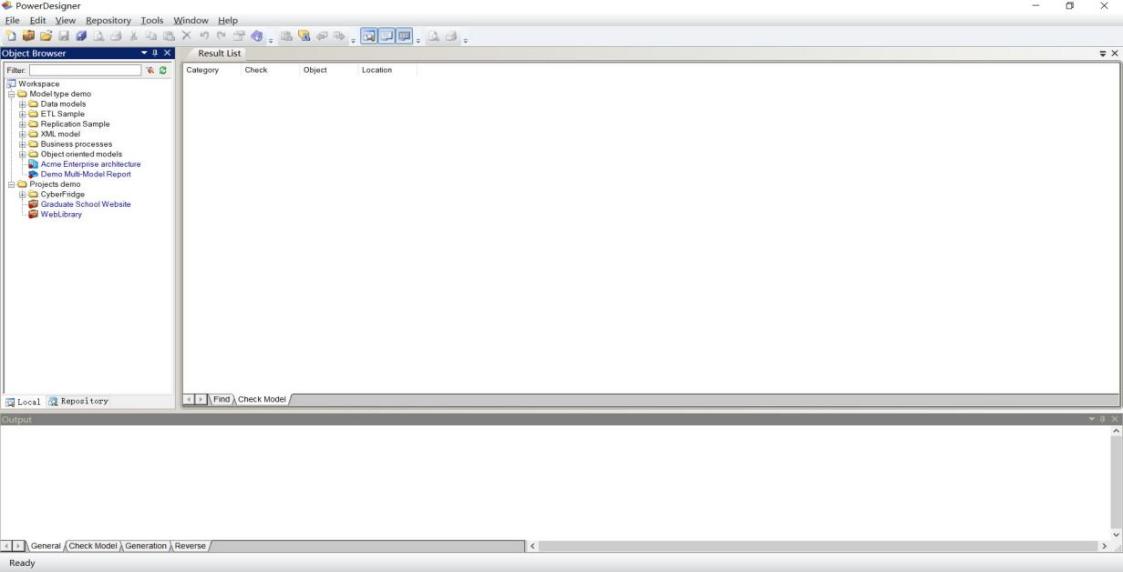
1. **CDM (conceptual data model)**：描述业务需求和数据实体的关系，不涉及具体数据库细节。它是最抽象的层次，关注数据的概念和业务规则。
2. **LDM (Logical Data Model)**：在 CDM 基础上，进一步具体化数据结构，定义实体、属性和关系，关注数据完整性和规范化，但不涉及数据库平台细节。
3. **PDM (Physical Data Model)：**最具体的模型，描述如何在特定数据库平台上实现数据模型，包括表结构、索引、存储结构等，优化数据库性能和存储。

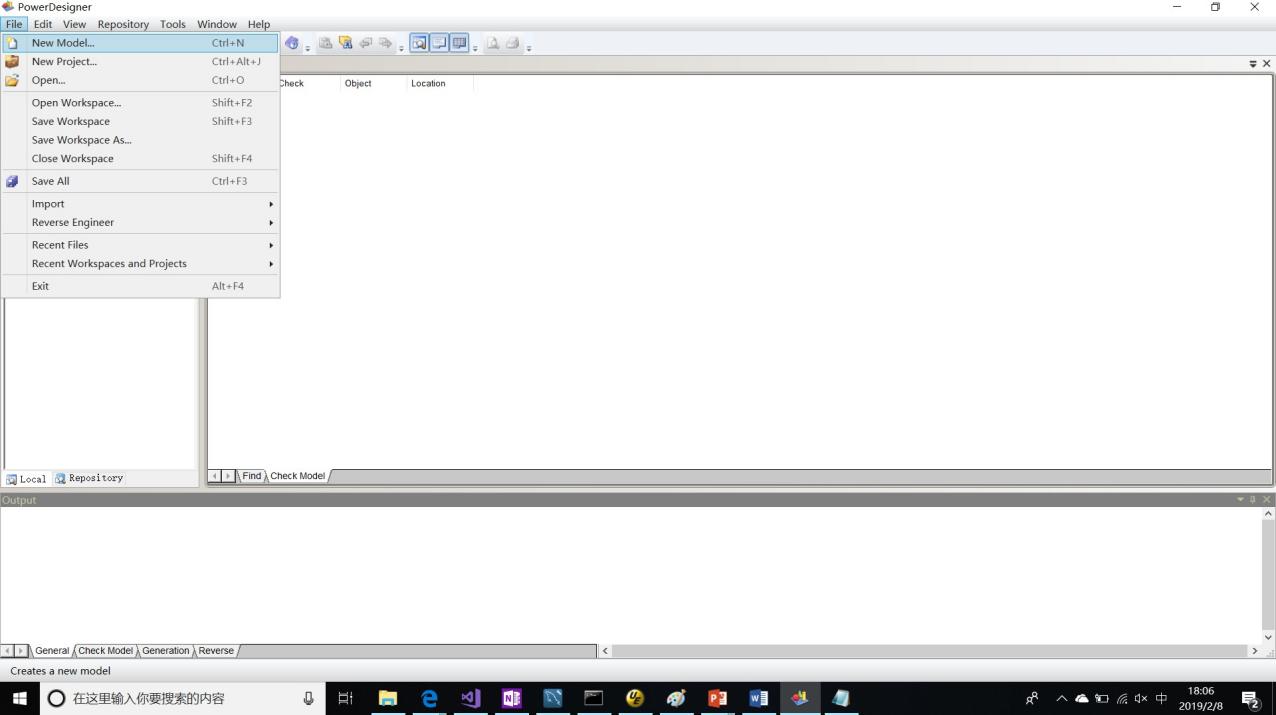
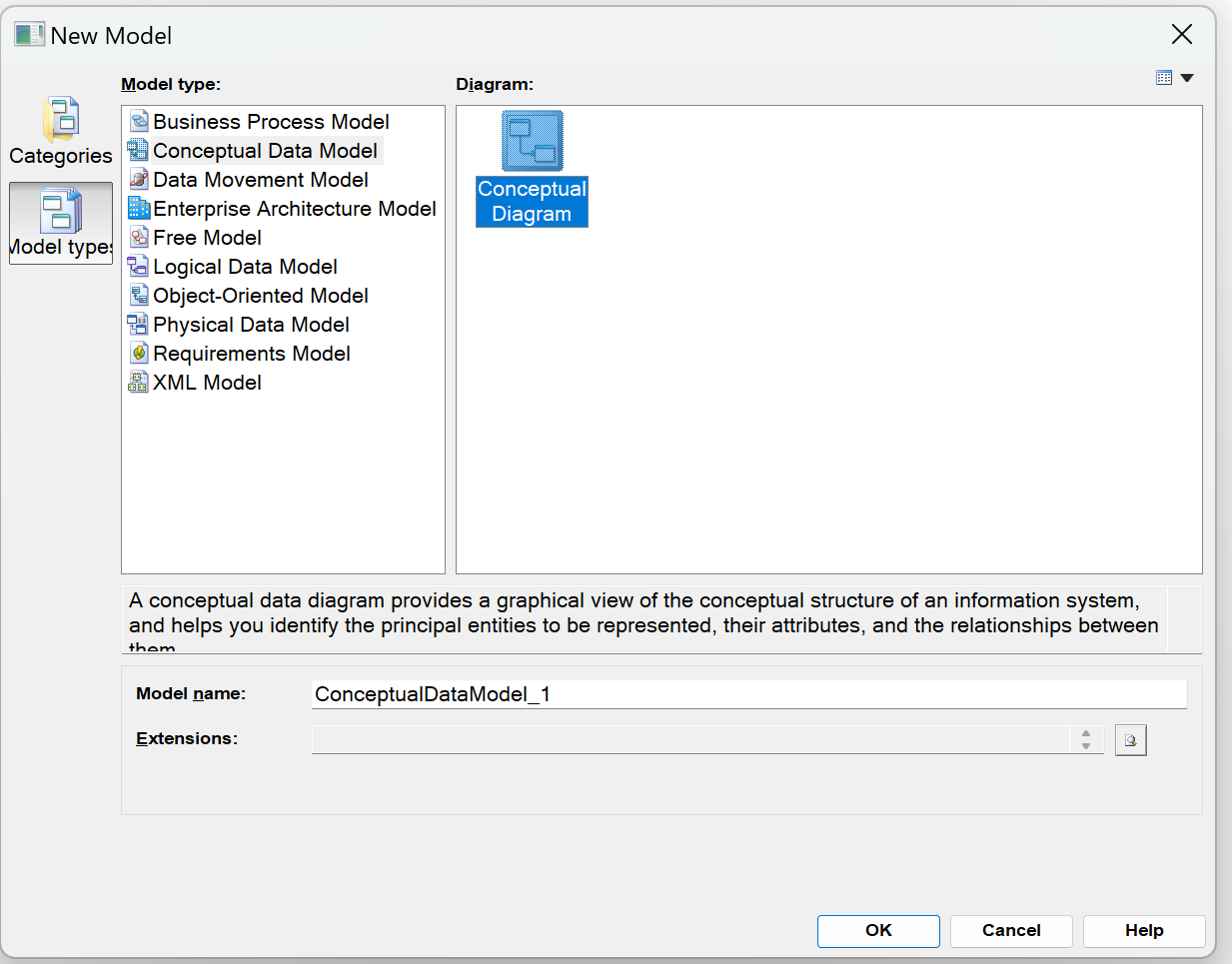
**CDM** 是业务层面的抽象，**LDM** 是结构上的设计，**PDM** 是数据库实现的细节。Powerdesigner中的 Data Architect 可利用实体- 关系图为一个信息系统创建“概念数据模型”-CDM。并且可根据 CDM 产生“逻辑数据模型”-LDM和基于某一特定数据库管理系统的“物理数据模型”。本实验使用PowerDesigner作为数据库设计工具，下载地址**<https://www.powerdesigner.biz/>**

## 绘制 CDM

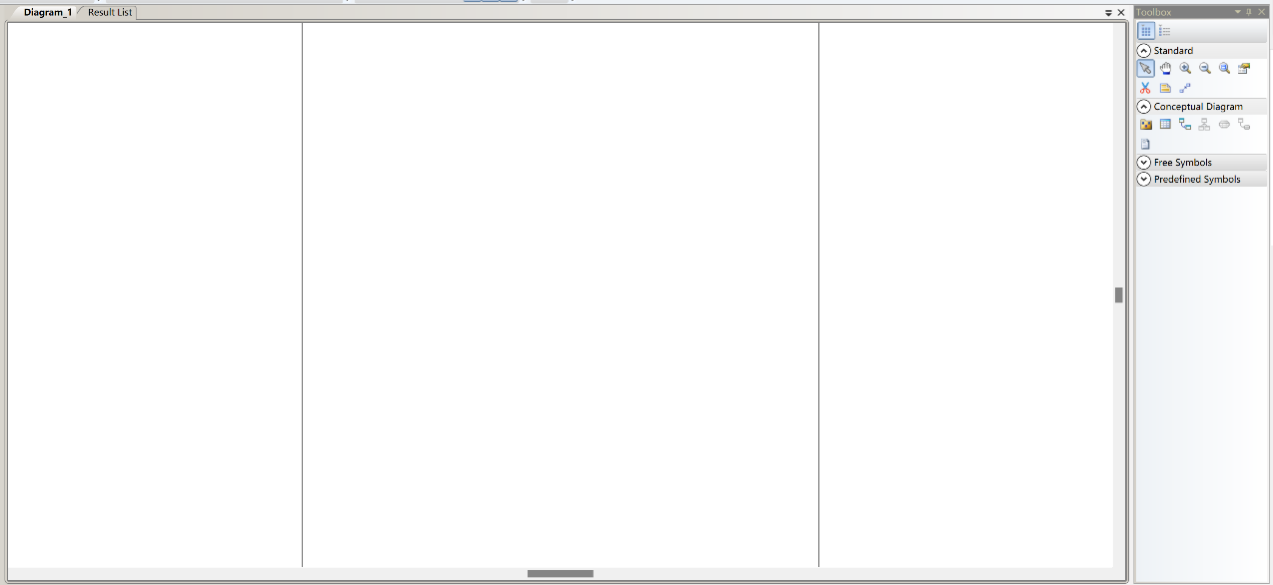
概念数据模型(CDM)，主要用途是确定业务领域的主要数据对象和它们之间的联系。为后续的设计（如 LDM 和 PDM）提供基础，确保开发团队充分理解业务需求。CDM 静态地描述系统中的各个实体以及相关实体之间的关系，系统分析员通过 E-R 图来表达对系统静态特征的理解。

我们以一个简单的“图书管理系统”为例，来学习绘制 ER 图。在绘制 E-R图之前，应该充分理解项目需要的实体、属性和实体之间的关系，明白整个系统的大体业务需求。

步骤 1．启动 PowerDesigner，进入 PowerDesigner 的操作主界面，如下图所示。

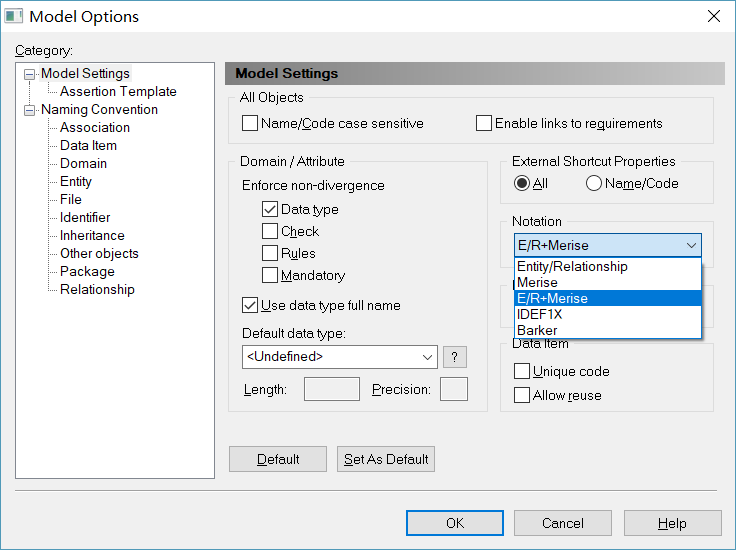
步骤 2：在 File 菜单中单击 New Model，如下图所示。在 Model types(模型类型)框中有Business Process Model(BPM，业务处理模型) 、Conceptual Data Model(CDM ，概念数据模型) 、Object- Oriented Model(OOM，面向对象模型)和 Physical Data Model(PDM，物理数据模型)等选项。我们选择 Conceptual Data Model 并单击“OK”按钮。

步骤 3：进入 PowerDesigner 的 CDM 操作窗口，如下图所示。



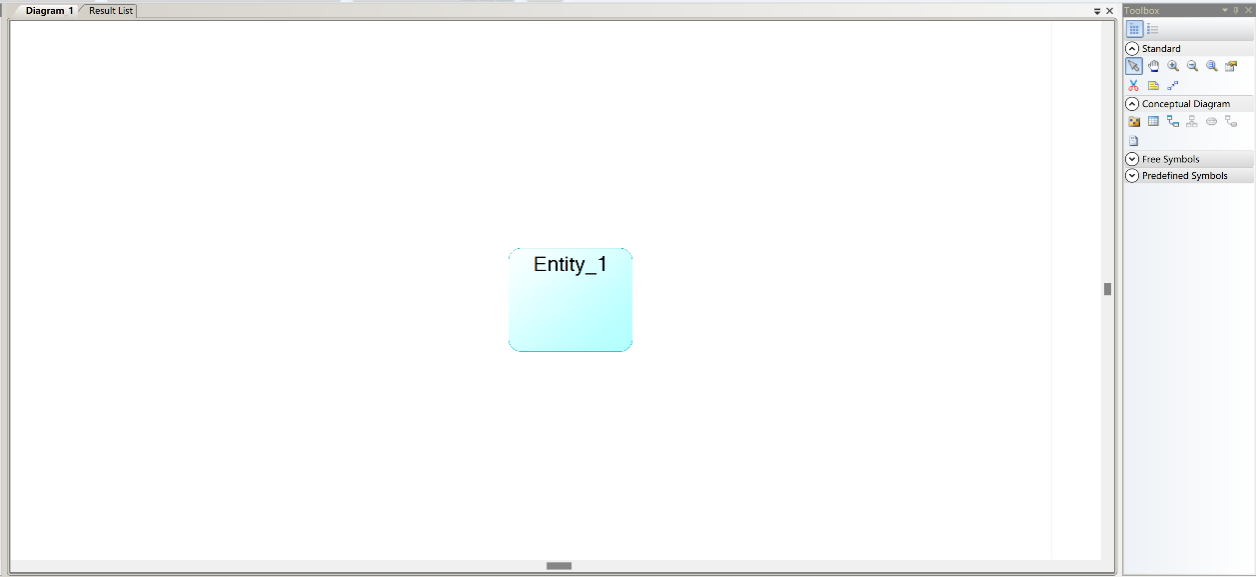
在 Toolbox 栏中，包括各种工具。用鼠标单击某个图标，再到界面中央单击(例如，画实体框)或拖动(例如，画实体关系联线)即可。

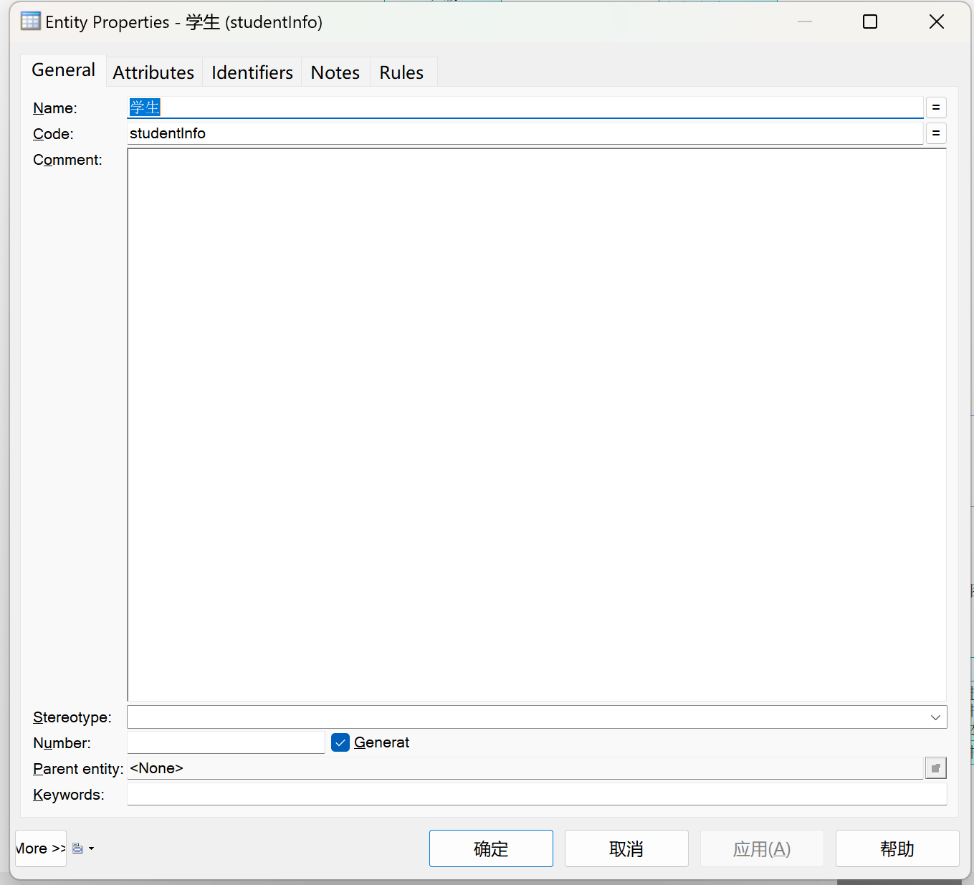
通过 Tools→Model Options→Model Settings→Notation 对其进行修改对象显示样式，如下图。



在 PowerDesigner 中，Notation（符号化方法）决定了模型图的表现形式和符号的使用方式，主要有两种类型：Entity/Relationship (E/R) 和 E/R+Merise。E/R是一种经典的数据建模方式，主要用于表示实体（Entity）、属性（Attribute）和实体之间的关系（Relationship）。E/R+Merise在 ER 模型的基础上还提供了 Merise 建模理论，允许在概念模型中使用 Association 和 Association Link。为了更好的表现实体之间的业务关系，这里建议使用E/R+Merise模型。

步骤 4：新增实体。Toolbox 中单击Concepted Diagram栏下的实体图标(Entity)，然后在操作界面中单击，得到一个实体框。单击指针图标(pointer)退出创建实体模式。



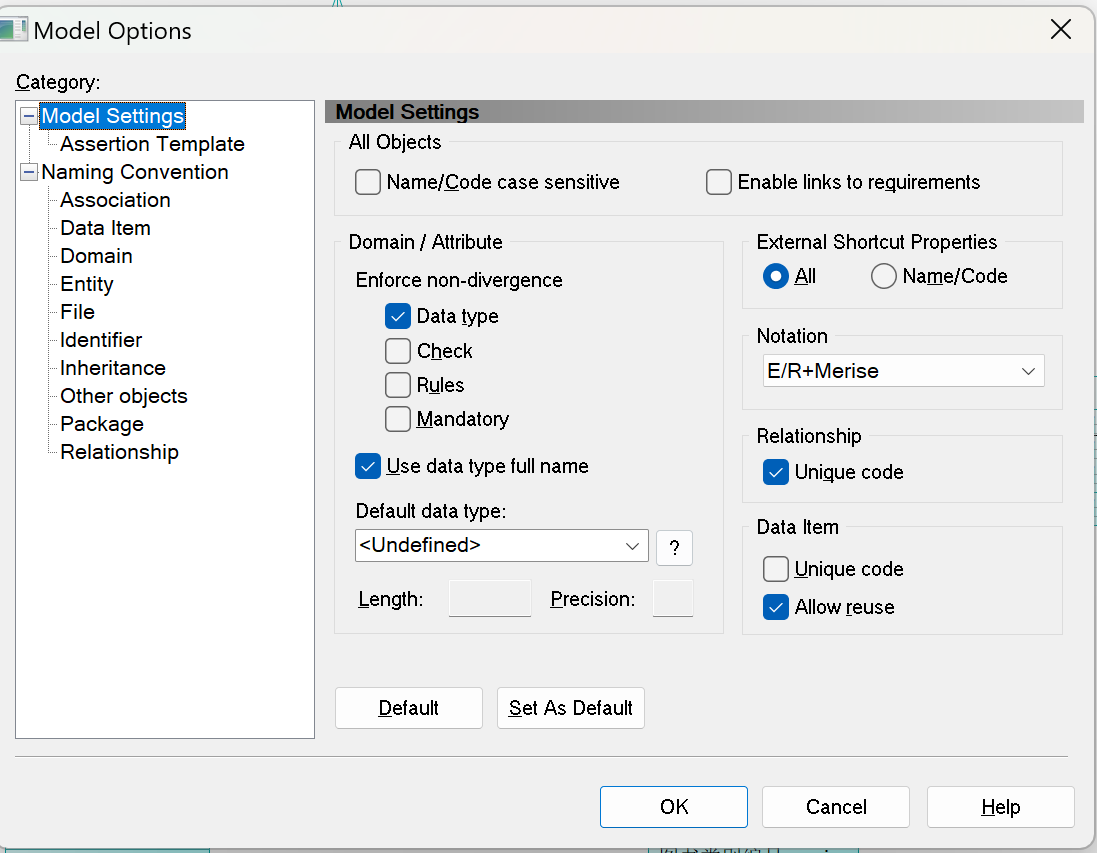
步骤 5：定义实体。用鼠标双击实体框，打开实体属性定义窗口，开始定义实体“用户”。

其中，General 选项卡用于设置实体名称、代码和注释等信息；Attributes 定义实体的字段或特性。每个属性对应一个数据表的列，包含有关实体的具体信息；Identifiers 标识符是用来唯一标识实体或属性的一部分，通常用于数据库中主键和外键的定义；Notes 备注字段用于添加关于实体、属性或模型的附加说明。

此例中，对 Name 栏输入“学生”，在 Code 输入“studentInfo”。

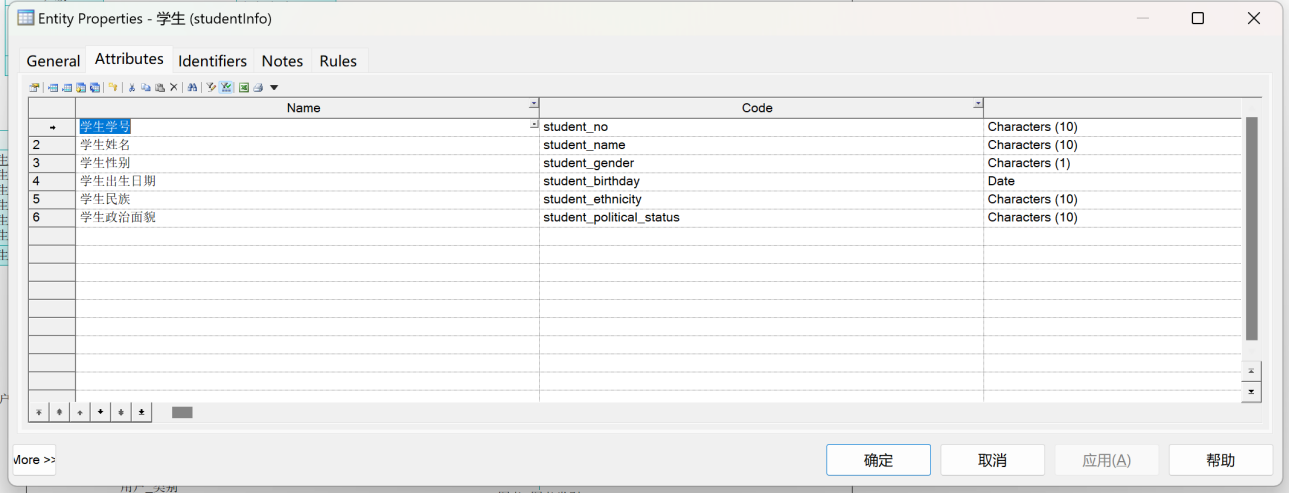
1、在PowerDesigner中，实体的 **Name** 和 **Code** 属性通常是联动的。也就是说，当你修改一个实体的 **Name** 时，PowerDesigner会自动更新该实体的 **Code**，确保两者保持一致。这种行为有时是方便的，但也可能会限制一些灵活性。如果想让它们不联动修改，可以在 Tools-General Options-Dialog 中取消 Name to Code mirroring 的勾选。

2、在PowerDesigner中，如果你在一个 ER 图中创建多个实体，且这些实体中存在具有相同名称的属性（例如，在不同的实体中都添加了名为 "code" 的属性），PowerDesigner 会默认将这些属性视为**全局唯一**的。如果你修改其中一个字段的类型或其他设置，其他相同名称的字段也会被同步修改。例如：book 实体里有一个属性 code 是 id，如果在 student 实体也新增一个属性 code 是 id，那么 PD 会认为两个 id 是同一个。当你修改一个 id 的 type 或者其他值时，另一个表的 id 也会被修改。如果你不想这样的情况出现，可以在 ToolsModel Options 找到 Data Item，去掉 Unique code 前面的勾。



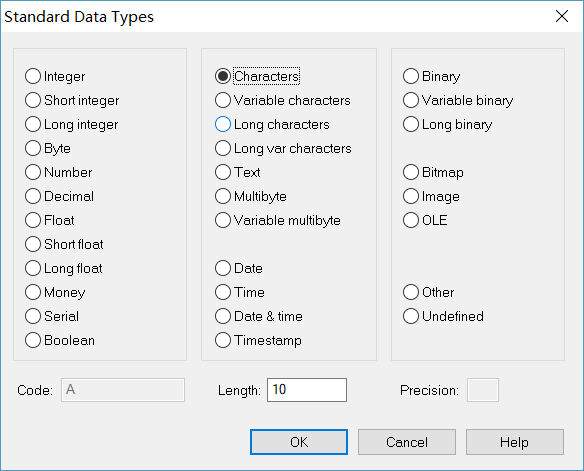
3、 code 不要使用 MySQL 的关键字，比如 return、describe 等。

步骤 6： 定义属性、属性的约束和算法。单击 Attributes 按钮，进入定义该实体的属性界面，如下图所示。



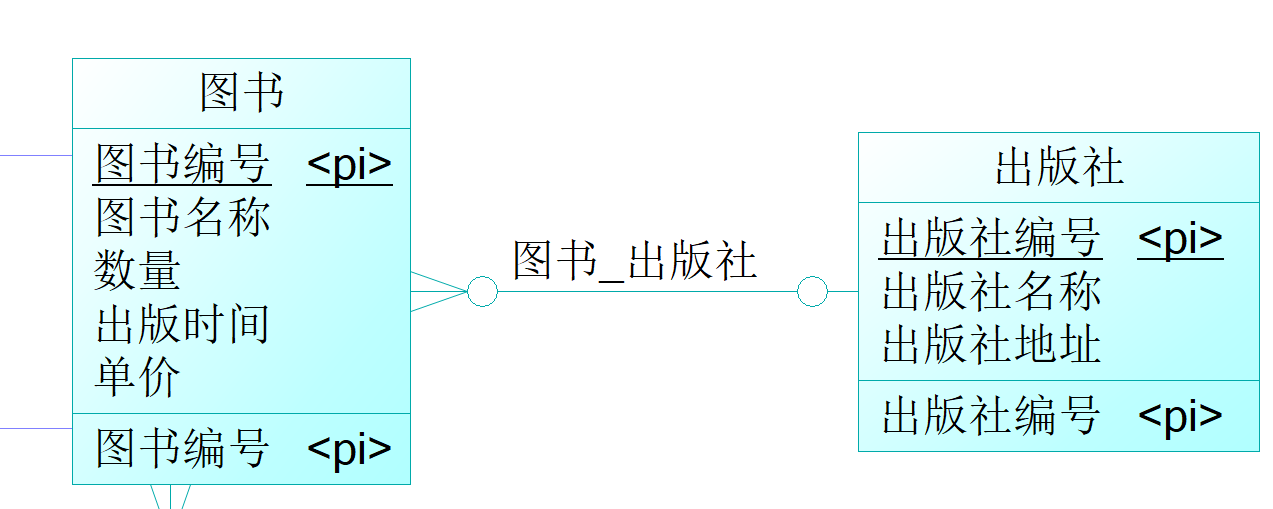
每一行定义一个属性，包括：

* Name：属性名称。
* Code：属性代码。
* Data Type：数据类型。
* Length：数据类型长度。
* Precision：数据类型精度。
* M(Mandatory)：强制，属性值是否允许为空。
* P(Primary Identifier)：表示是否为主标识符。
* D(Displayed)：表示是否在实体图形符号中显示该属性。
* Domain：表示应用到该属性上的域。

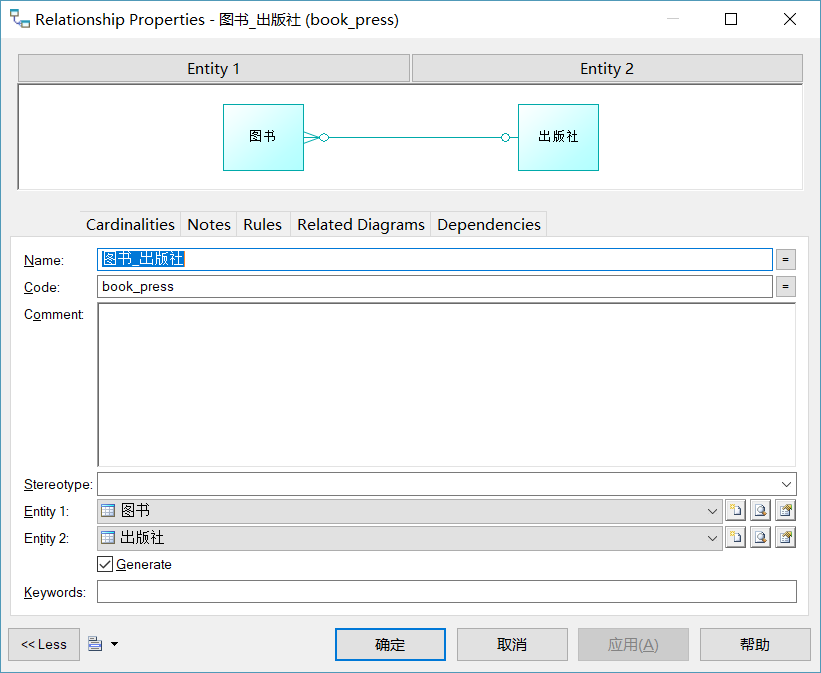
在定义类型的时候，可以通过单击“…”按钮显示全部类型选项并从中选择，如下图所示。

按照这个方法定义其他几个实体。

步骤 7：定义联系。定义好实体和属性后，接下来定义实体之间以及实体内部的联系。单击 Toolbox 中的 Relationship 工具选项，光标由指针形状变为该图标形状，在需要设置联系的两个实体中的一个实体图形符号上单击鼠标左键，并在保持按键的情况下将鼠标拖拽到另一个实体上，然后释放鼠标左键。这样就在两个实体之间创建了一个联系。如下图所示。



步骤 8：双击联系图形符号，打开联系属性设置窗口，如下图所示。设置 “图书”和“出版社”两个实体之间的联系，联系名称为“图书-出版社”。



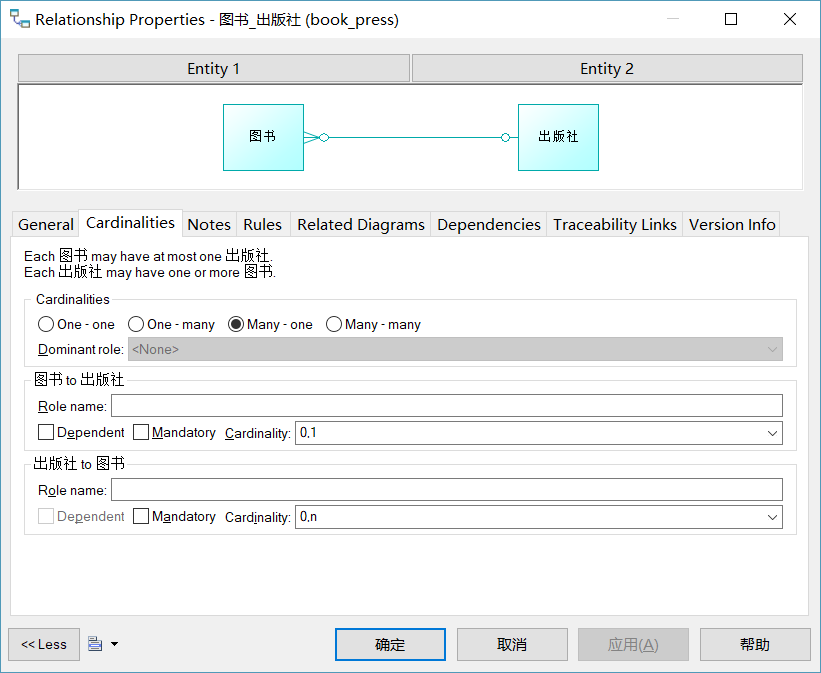
General 选项卡用于设置联系的基本信息，主要参数含义如下：

* Name：联系名称。
* Code：联系代码。
* Comment：注释。
* Entity 1 和 Entity 2：联系两端实体的名称。
* Generate ：在 CDM 生成 PDM 时， 将生成 PDM 中的参 照/ 引用

(Reference)。

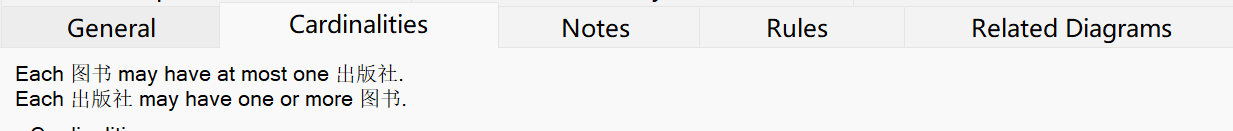
* Keywords：设置关键字。

Cardinalities 选项卡用于设置联系基数信息，如下图所示。设置“图书”和 “出版社”之间的联系为“n：1”联系；“图书 to 出版社”的联系基数为“0， 1”；“出版社 to 图书”的联系基数为“0，n”。

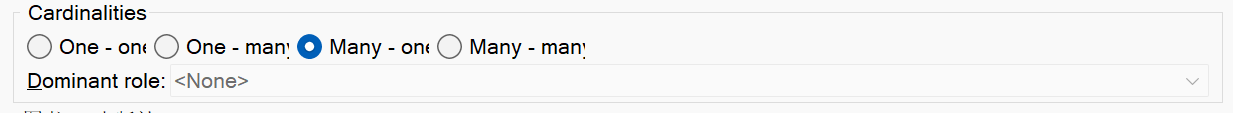


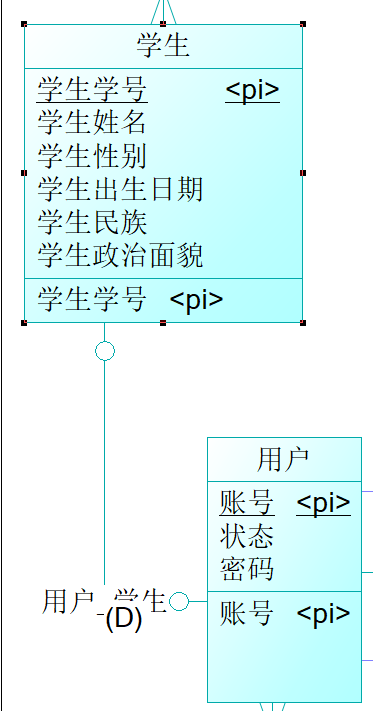
其中：

* Cardinalities ：联系的类型， 有 One-one 、One-many 、Many-one 、

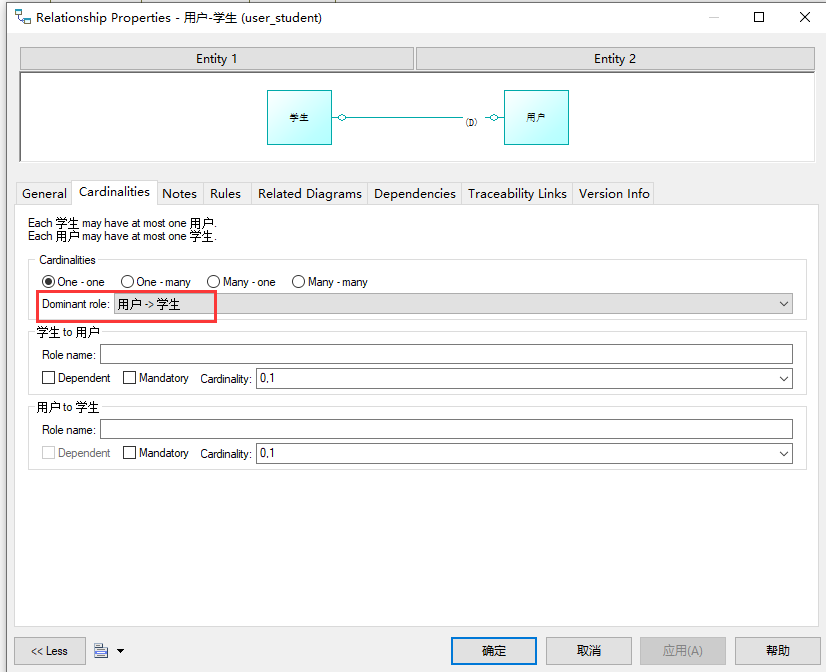
Many-many。

参考上方提示确定自己的选择是否正确。

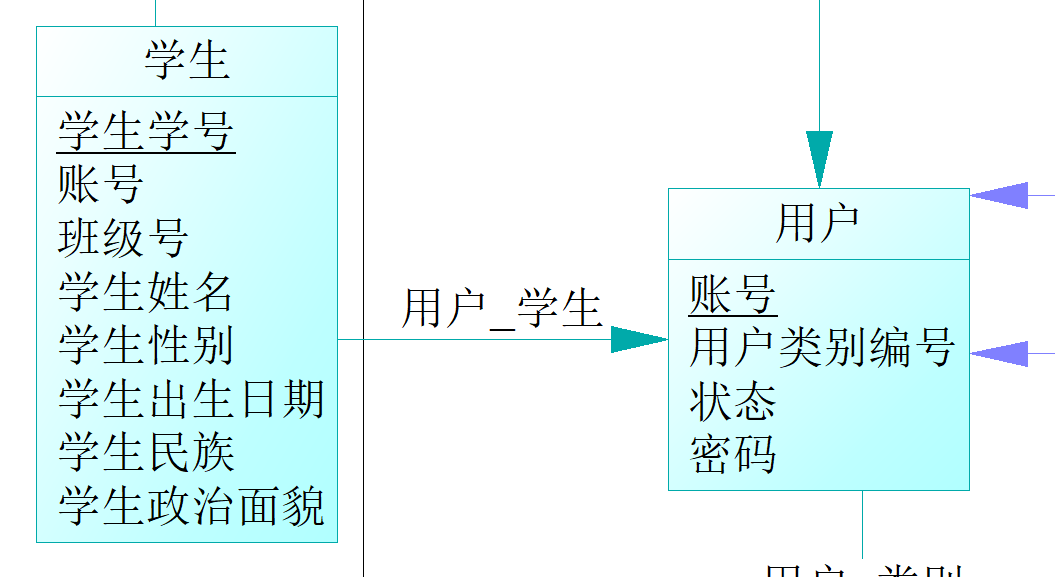
Cardinalities 设置为 One- many、Many-one，生成的 PDM 中会自动把一个实体的主键作为另外一个实体的外键。如果Cardinalities 设置为One-one，PowerDesigner会根据Dominant role的设置确定参照/引用。Dominant role参数是只针对“1：1”联系，用于定义该联系中起支配(主导)作用的角色。在 CDM 生成 PDM 时，如果定义该参数则在依赖实体对应表中生成一个参照/引用(Reference)；如果没有定义该参数则在两端实体对应表中分别生成一个参照/引用。



如果设置 Dominant role 为用户学生：

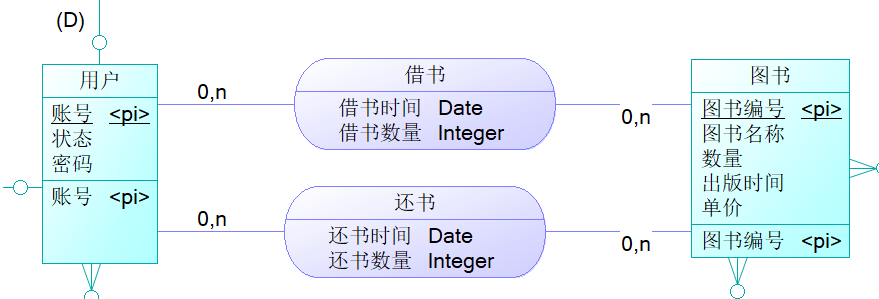


在生成的 PDM 中，学生实体对应的表中将加入引用(外键)-用户账号：

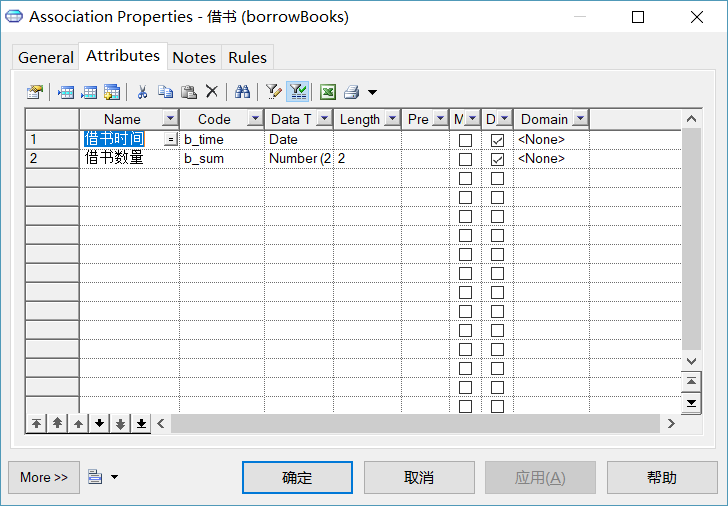


* Dependent：依赖关系。表示实体所包含的基本信息必须依赖于另一个实体的基本信息。如果定义了依赖关系，在 CDM 生成 PDM 时，依赖实体中将生成一个引用，并且该引用将作为依赖实体标识符的一部分。两个实体之间存在依赖关系的联系又称为标定联系；如果没有定义依赖，则称为非标定联系。
* Mandatory：强制。强制状态下联系的基数分为“1，1”和“1，n”两种。

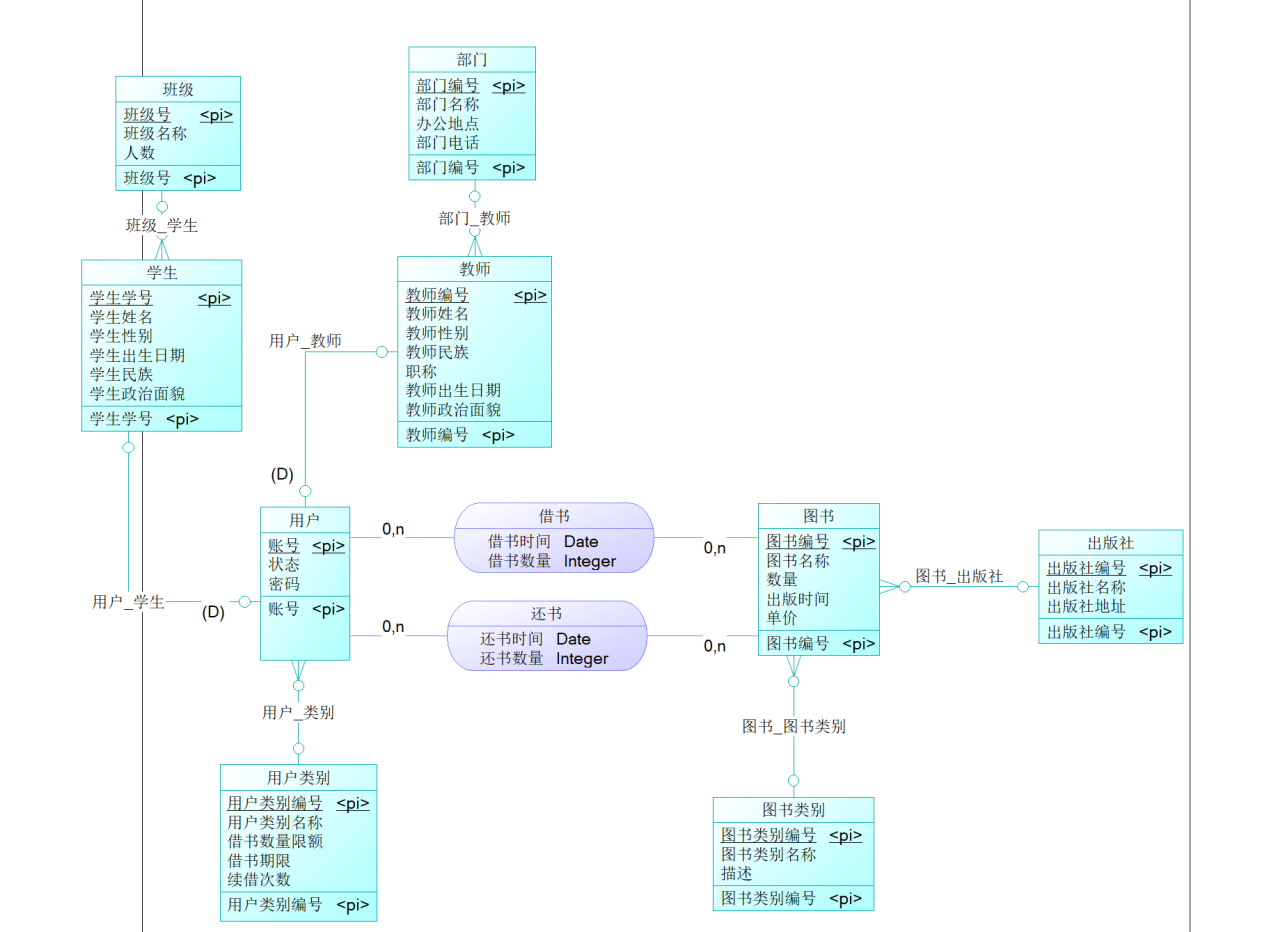
步骤 9：定义关联。PowerDesigner 提供了关联、关联链接用于描述本身有属性的关系。本例中实体“读者”和“图书”之间存在“借书”、“还书”的关系，而且“借书”、“还书”还包含属性。接下来定义实体之间的关联。单击

Toolbox 中的 Association 工具选项，光标由指针形状变为该图标形状，在需要设置关联的两个实体中间点一下，这样就在两个实体之间创建了一个关联。然后单击 Toolbox 中的 Association Link 工具选项，把“图书”与刚创建的关联与“读者”链接起来，如下图所示。

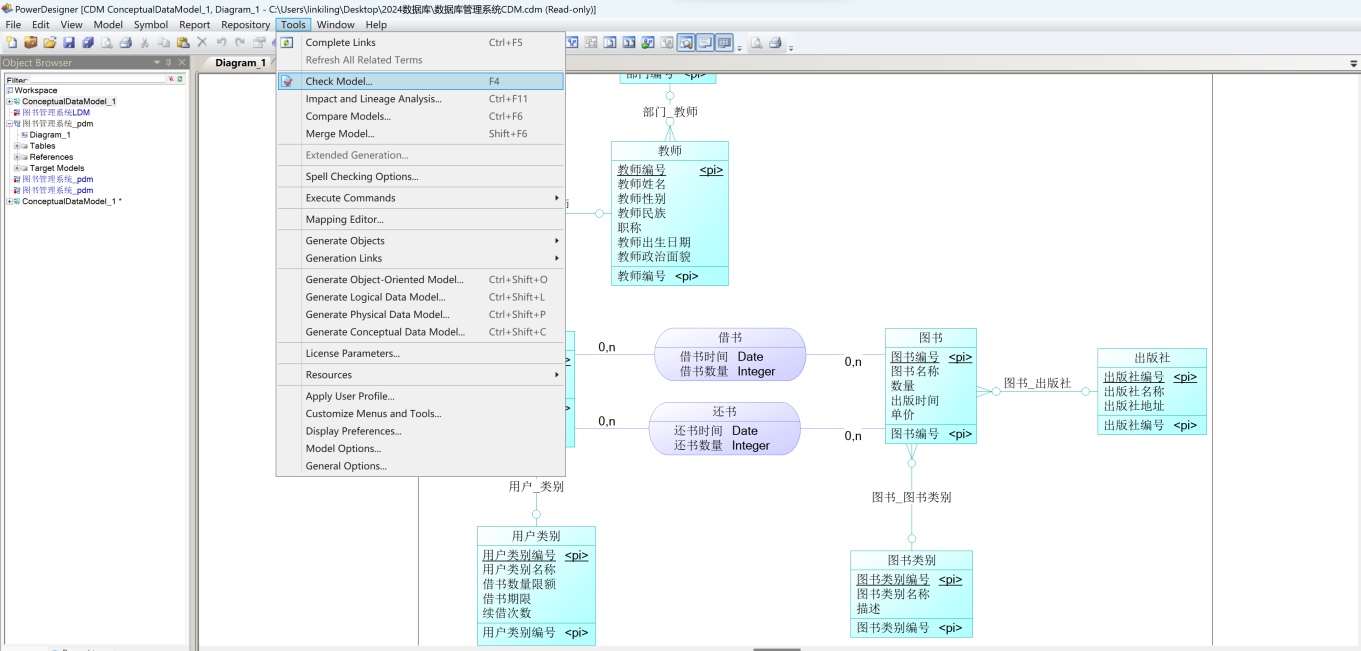
关联是需要设置属性的，如下图：

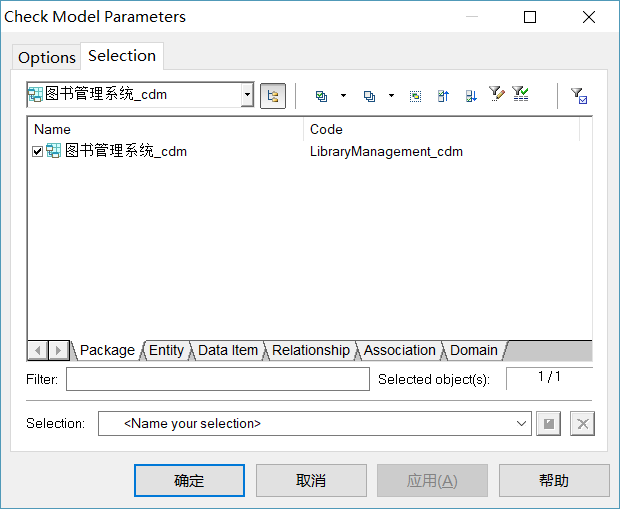


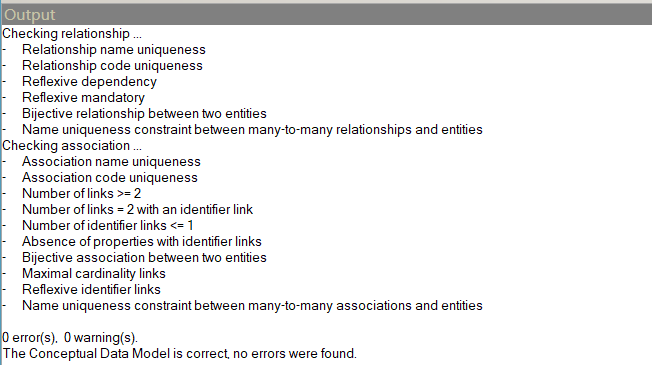
步骤 10：系统所有的实体、关系都定义完毕后，保存此 CDM。



步骤 11：在 Tools 菜单中单击 Check Model 命令来检查 E-R 图的错误。





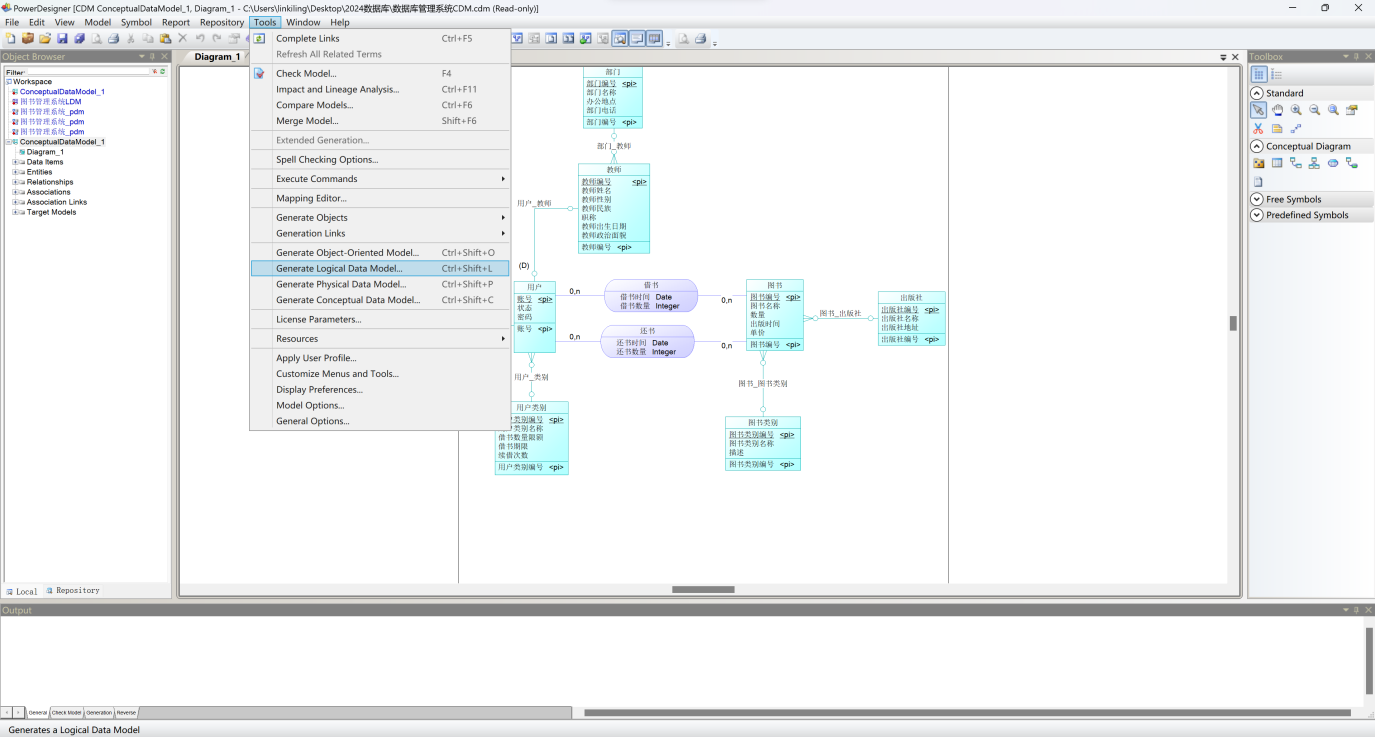


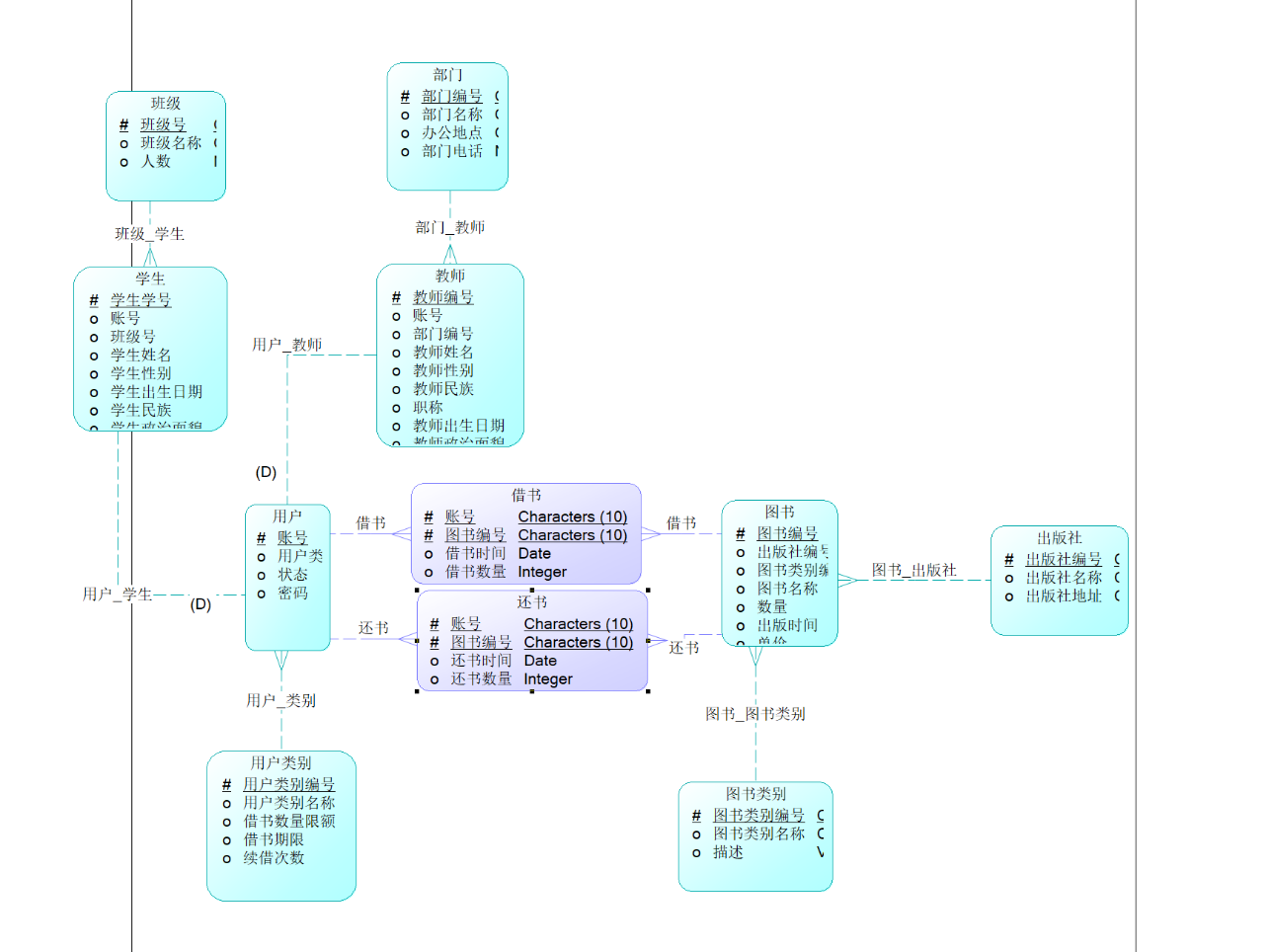
若没有错误error，则保存此CDM图。

## CDM 转成 LDM

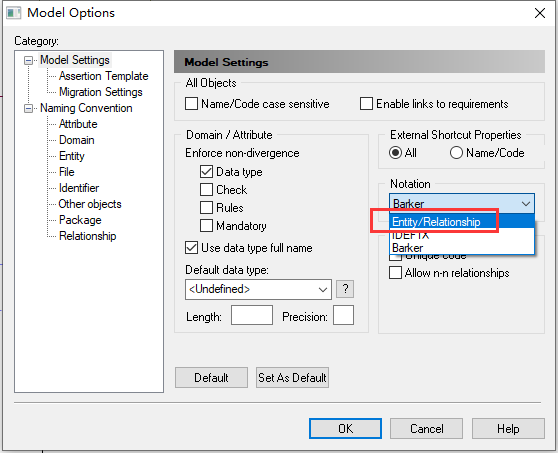
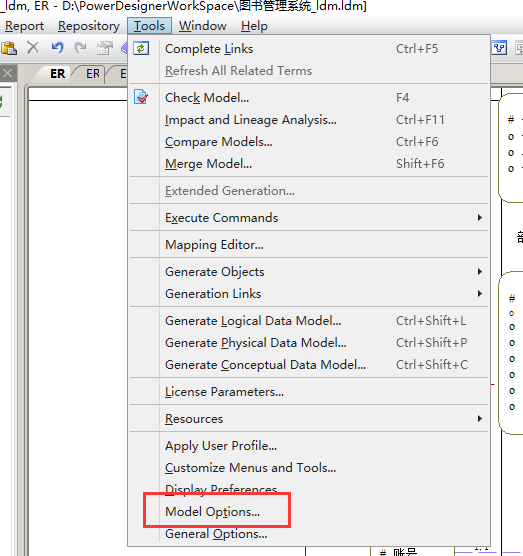
CDM 模型完成的是系统的概要设计，有了 CDM 模型之后，可以利用系统提供的自动转换功能将 CDM 模型转换成 LDM 模型，而不需要重新定义。

步骤 1：在 Tools 菜单中单击 Generate Logical Data Model 命令，打开逻辑数据模型设置窗口，如下图所示。

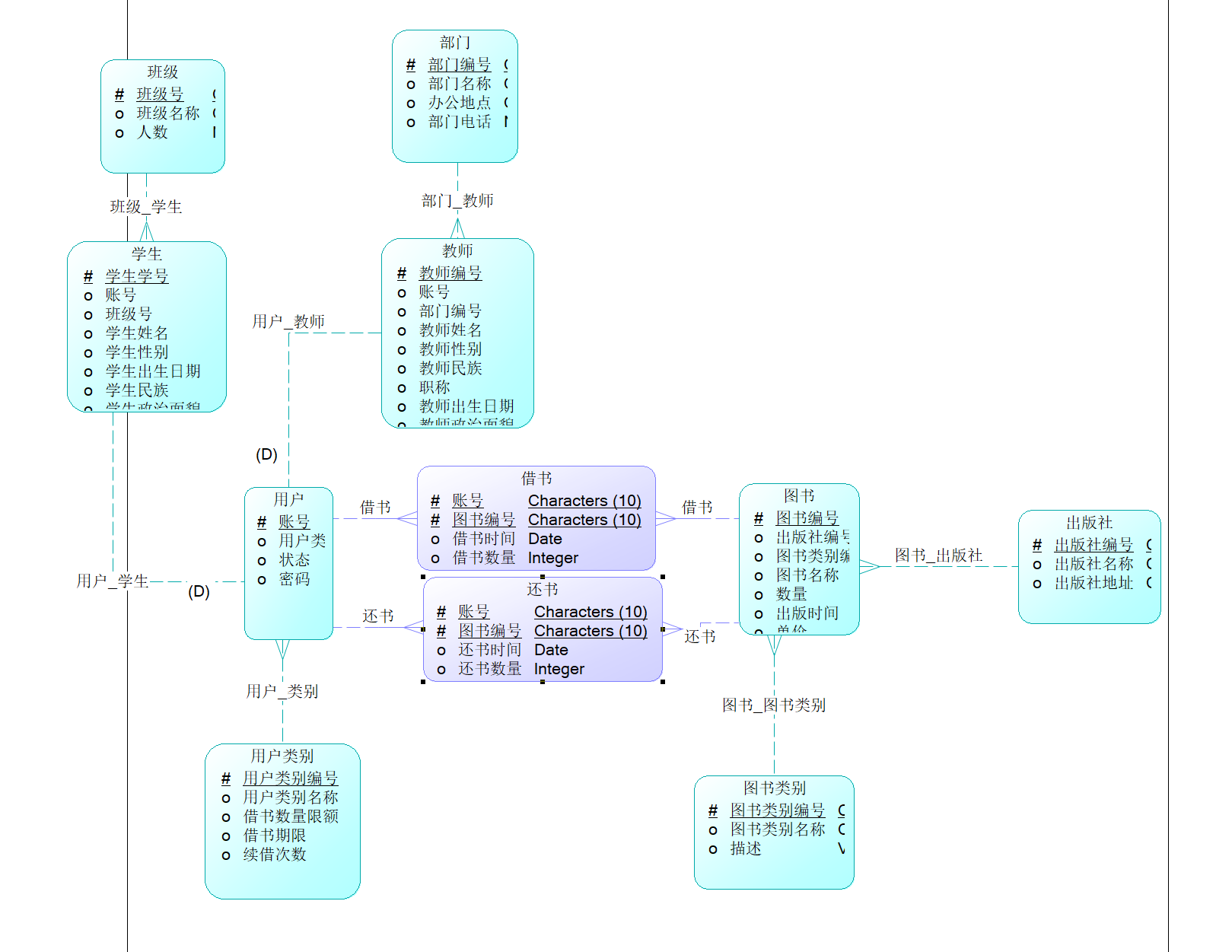


步骤 2： 默认转换成的 LDM，notation 是 Baker 方式：

我们可以修改成我们习惯的 Entity/Relationship 类型：

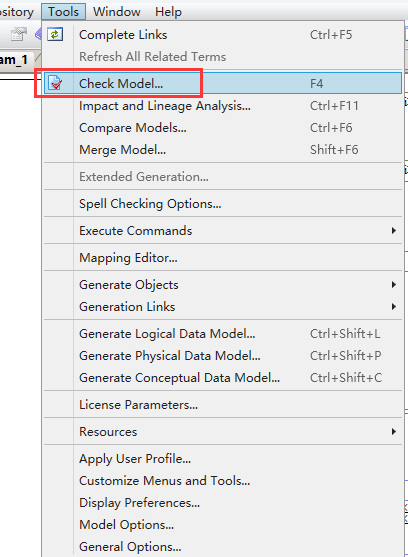


OK 之后：



步骤 3：可以利用鼠标拖动实体框和关联线，对 LDM 图形进行调整，直到图形整齐、美观为止，然后进行保存。

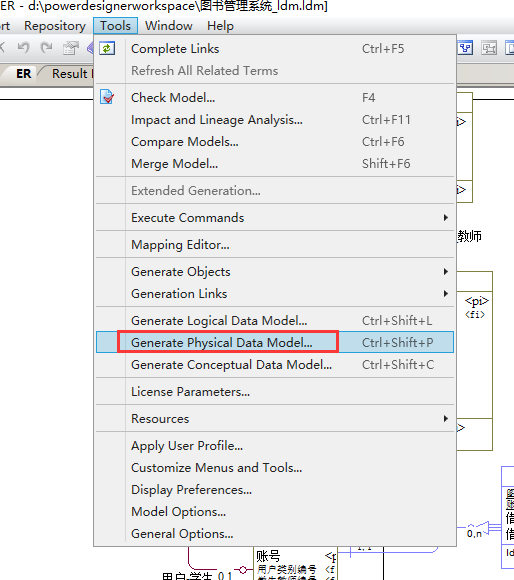
步骤 4：在 Tools 菜单中单击 Check Model 命令来检查 LDM 图的错误。

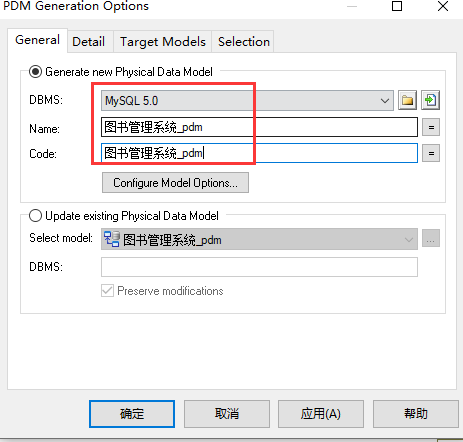


## LDM 转成 PDM

有了 LDM 模型之后，可以利用系统提供的自动转换功能将 LDM 模型转换成 PDM 模型，而不需要重新定义。(也可以由 CDM 图直接转成 PDM 图，再在 PDM 图基础上进行修改)

步骤 1：在 Tools 菜单中单击 Generate Physical Data Model 命令，打开物理数据模型设置窗口，如下图所示。





·Generate new Physical Data Model：选中此项，表示生成新的物理数据概念模型。

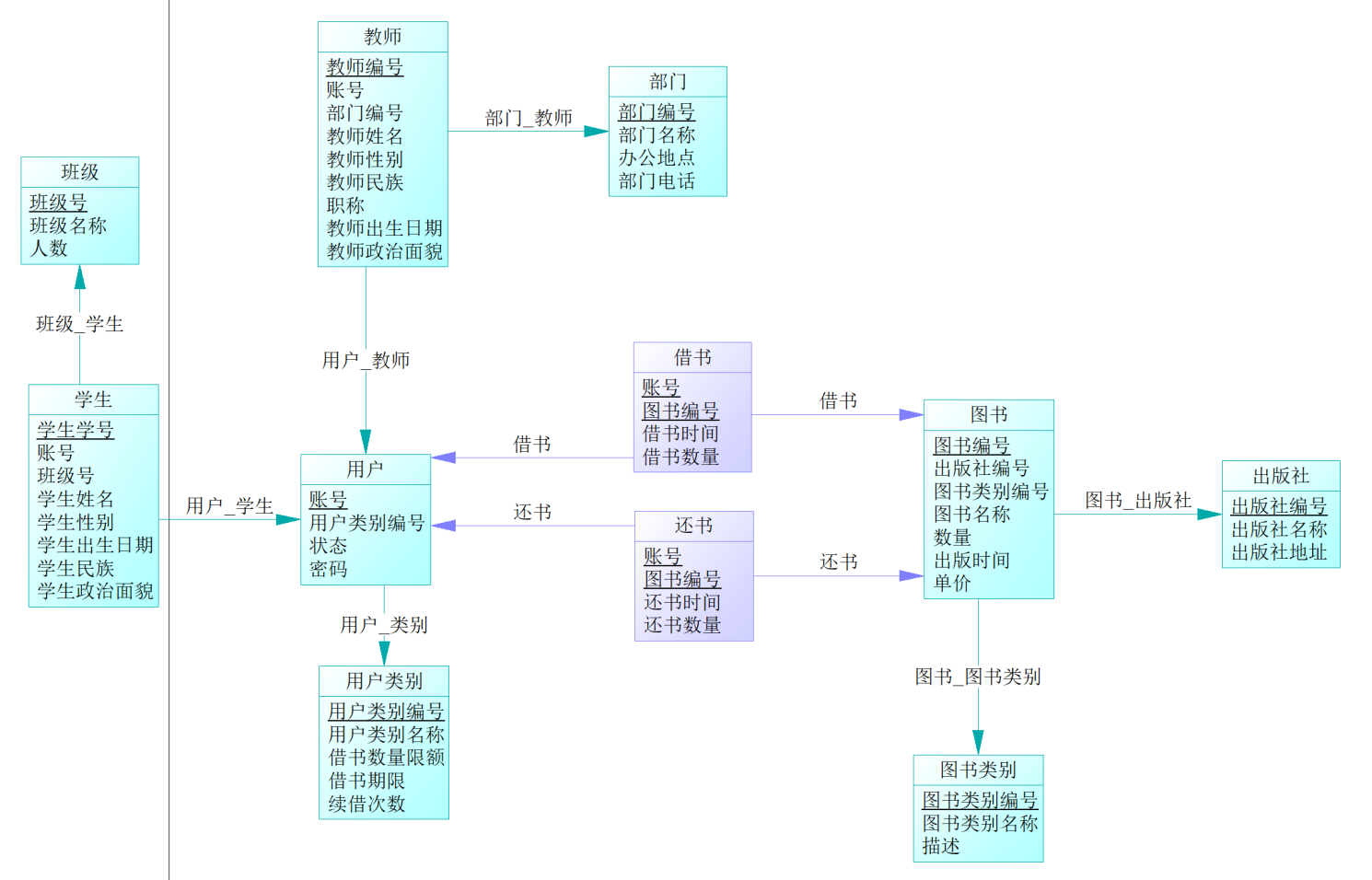
·Update existing Physical Data Model：选中此项，表示更新到原有的物理数据概念模型。

·DBMS：选择数据库类型，支持不同数据库的不同版本。我们选 **MySQL5.0**。

·Name：物理数据模型的名称。

·Code：物理数据模型的代码。

设置完毕后，单击“确定”按钮，开始生成物理数据模型，如下图所示。



步骤 2：可以利用鼠标拖动实体框和关联线，对 PDM 图形进行调整，直到图形整齐、美观为止，然后进行保存。从上图可见，PDM 与物理建表已经很接近。

## 在 PDM 上创建视图、索引、触发器、存储过程、存储函数

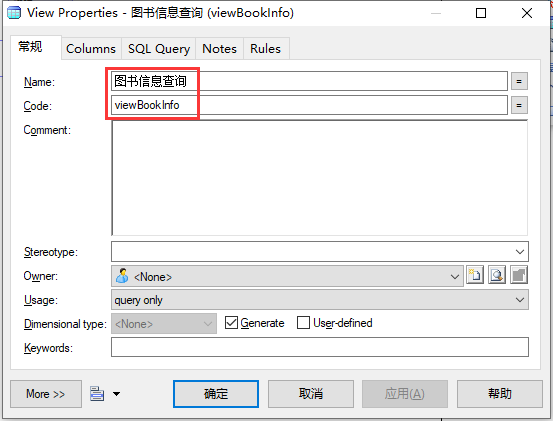
下面介绍如何创建视图、定义索引、定义触发器、定义存储过程和存储函数。

### 创建视图

1、单击Palette（有些版本是Toolbox）里面的视图图标:

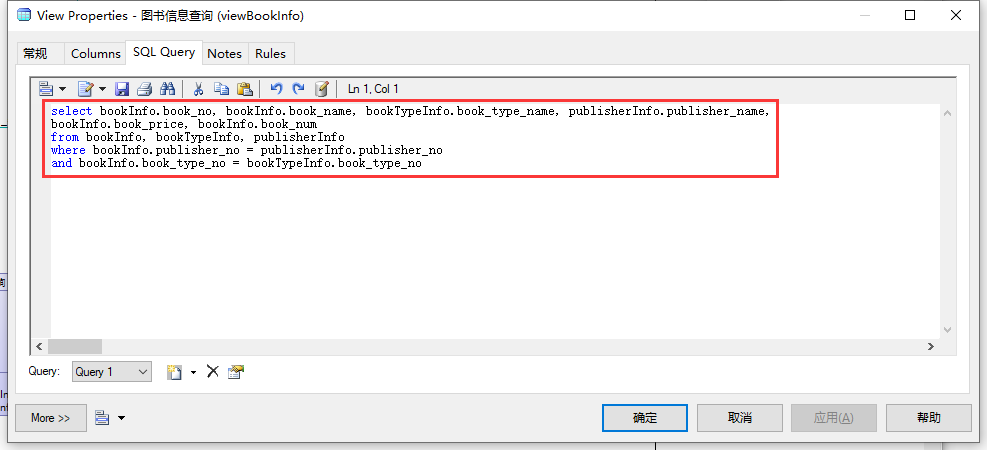


2、设置视图的名字、code 等属性：



注意倒数第三行的Usage参数，这是表示视图是只读的视图还是可更新的视图，如果我们只创建一般的视图，那么选择 query only选项即可。

3、切换到 SQL Query 选项卡，在文本框中可以设置视图定义的查询内容。在定义视图时最好不要使用\*，而是一一给出列名，这样在视图属性的 Columns 中才能看到每个列。如下图：



如果视图中没有出现对应名字的属性，查询的表也有缺少，可以检查一下SQL语句是否正确，比如这样：



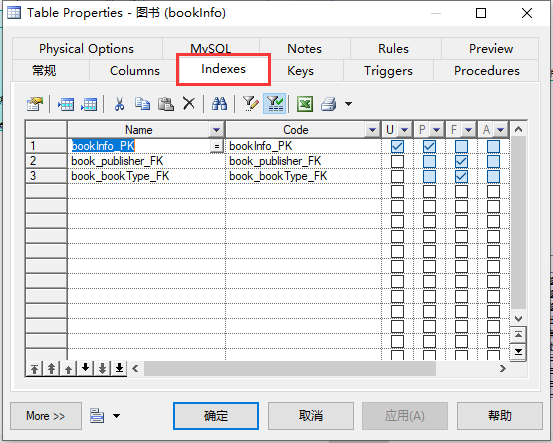
确认无误后视图就建好了：



### 定义索引

在 MySQL 中，主键、外键都会自动建索引，不需要额外设置。如果需要对非主键、外键字段建索引，具体方法如下：

1、在 PDM 模型中选择要建立索引的表， 打开表属性窗口， 选择Indexes 选项卡：



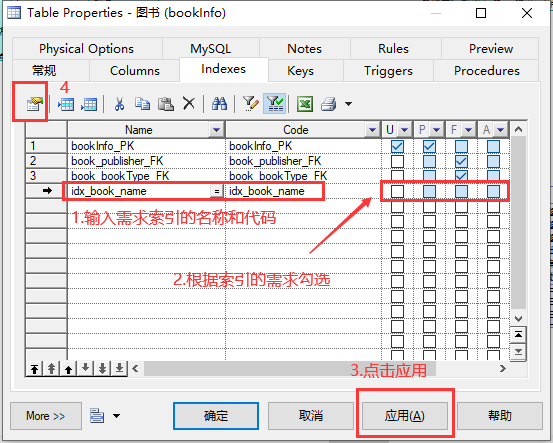
2、在空白行输入索引名称和代码等属性，根据需要勾选以下选项：

U：唯一索引。

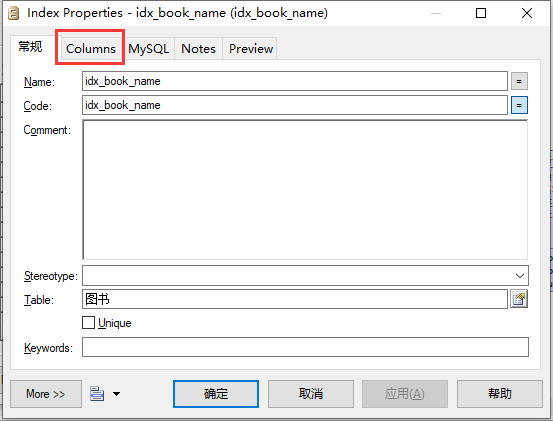
P：主键索引。

F：外键索引。

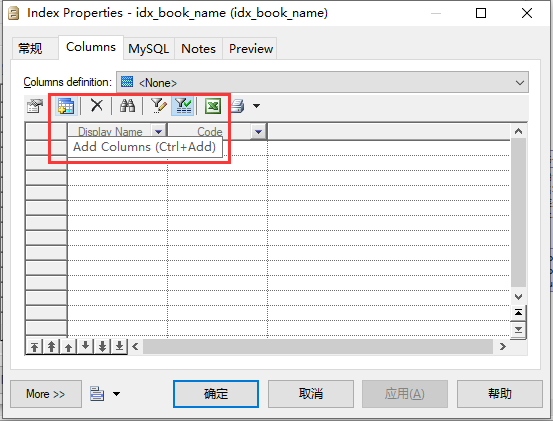
A：候选键索引。



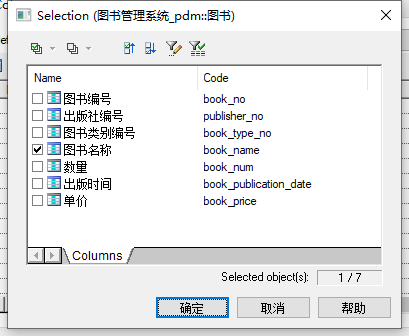
点击应用，然后点击 Properties。 3、 选择 Columns 选项卡：



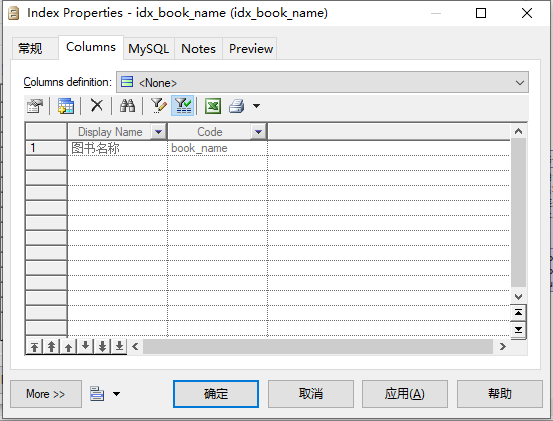
1. 点击 Add columns：



1. 勾选你想建索引的一列或者多列，然后点击 OK：



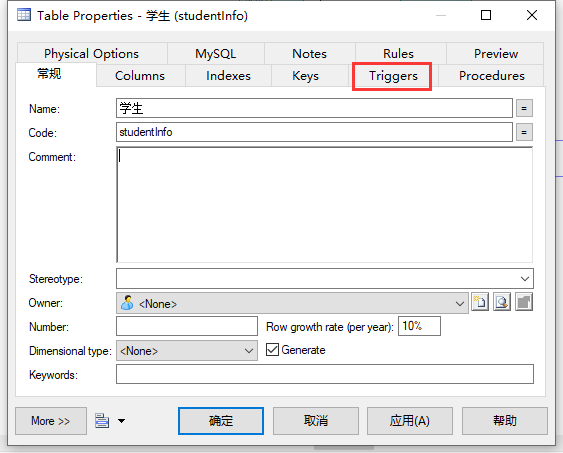
1. 点击确定：



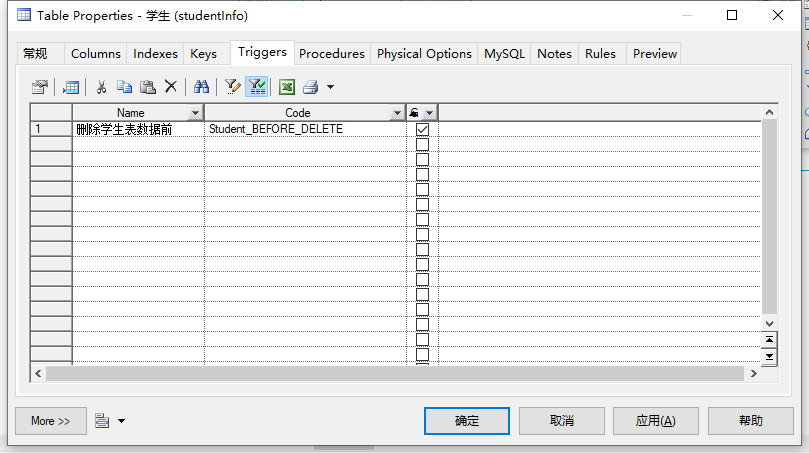
这样就在 book\_name 上建了一个索引。

### 定义触发器

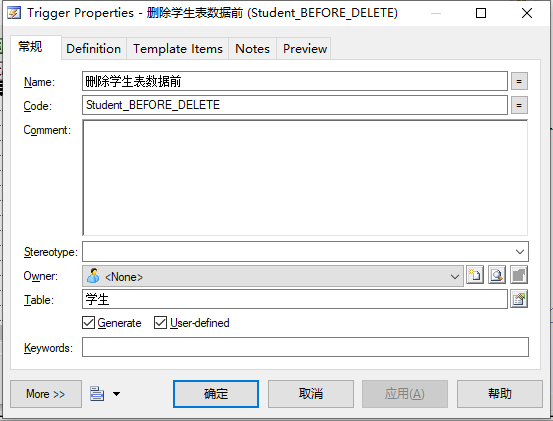
1、在PDM 中，找到需要建触发器的表，打开以后找到 Triggers 选项卡：



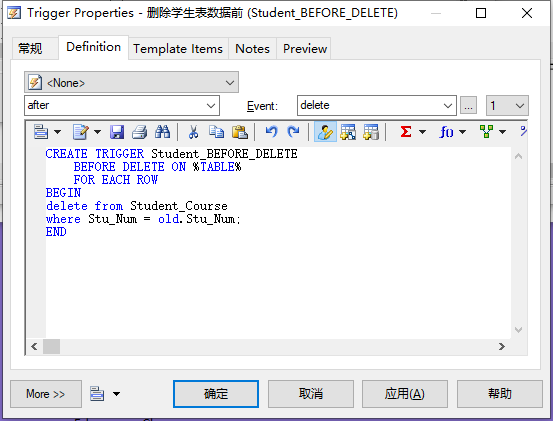
2、新增一行：



1. 双击trigger左侧序号，或者选中后点击左上角打开 trigger Properties 设置页面：



1. 在 Definition 中输入建立索引语句：

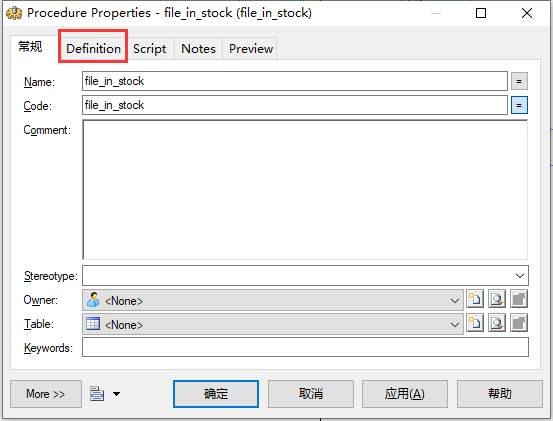


### 定义存储过程或存储函数

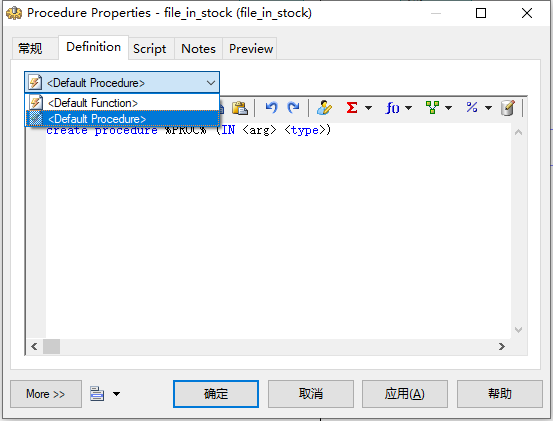
1、工具栏中单击 Procedure 按钮创建存储过程：



2、双击存储过程打开属性窗口→选择 Definition 选项卡：



3、在下拉列表框中，有 Default Procedure、Default Function 这两个选项，前者是定义过程，后者是定语函数：



4、编辑存储过程或存储函数的脚本：

图形用户界面, 文本, 应用程序, Word

描述已自动生成

5、写完以后看一下 Preview 页，核对一下 SQL 语句：

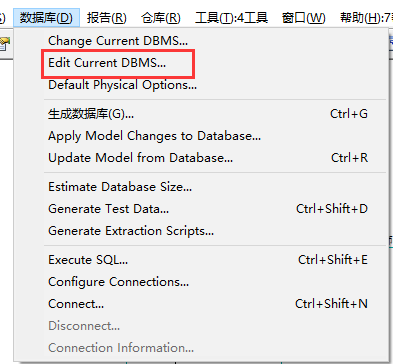
图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

如果没有问题就可以点击确定保存。

## PDM 转成 SQL 脚本

步骤 0：我们需要额外设置一下才能在生成的 SQL 脚本中出现 PDM 中设置的view。在 PDM 界面下，菜单 Database  Edit Current DBMS…

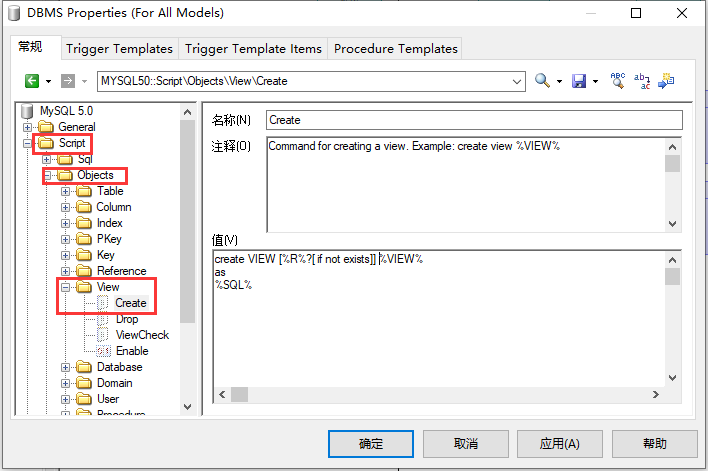


在左侧目录中按Script->Objects->View，进入Create，在value中输入

create VIEW [%R%?[ if not exists]] %VIEW%

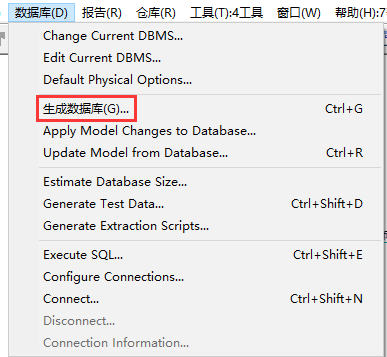
as

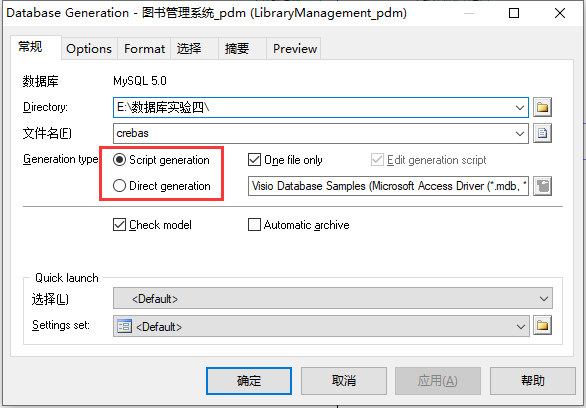
%SQL%



以及在Drop的value中输入drop table if exists %VIEW%

步骤 **1**：在 DataBase 菜单中单击 Generate Database（生成数据库） 命令，如下图所示。





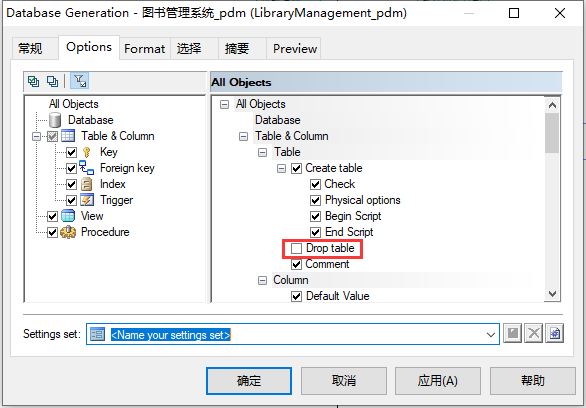
General选项卡：

* Directory：脚本文件路径。
* File Name：脚本文件名。
* Generation type：生成类型。
  + Script Generation：表示生成脚本文件，若同时选择 One file only 复选框，表示生成一个脚本文件；
  + Direct Generation ：表示直接生成到数据库中，若同时选择 Edit

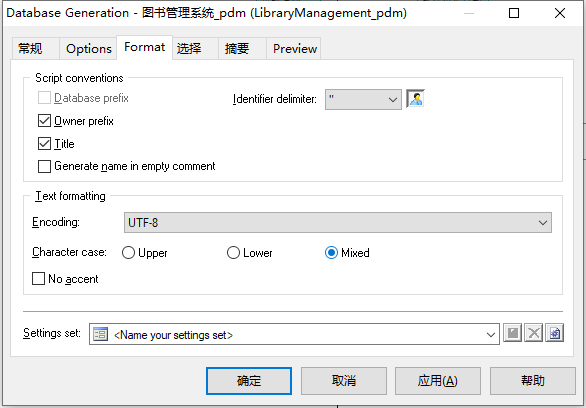
generation script 复选框，表示生成结束后可以编辑脚本文件。还可以使用右边的 Connect to a Data Source 工具，选择或配置数据源

* Check model：生成数据库时系统自动检查 PDM 的有效性。
* Automatic archive：自动归档。
* Selection：选择事先配置的生成对象。
* Setting set：选择生成对象类型。

Options选项卡：



Format 选项卡：

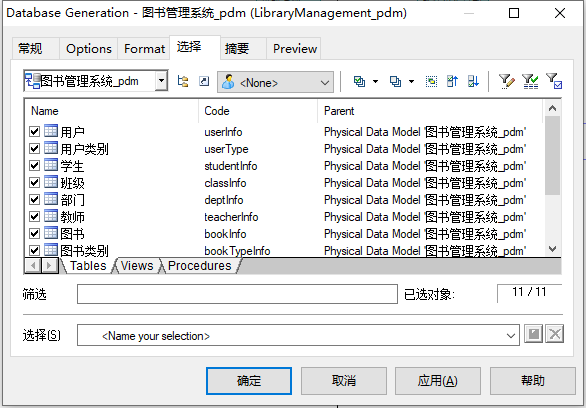


* Script Conventions
  + Database prefix：代码中表和视图名称的前面带有数据库名的前缀。
  + Owner prefix：代码中表和视图名称的前面带有所有者的名称。
  + Title：代码中每节前面包括一个注释， 例如： Database Name：

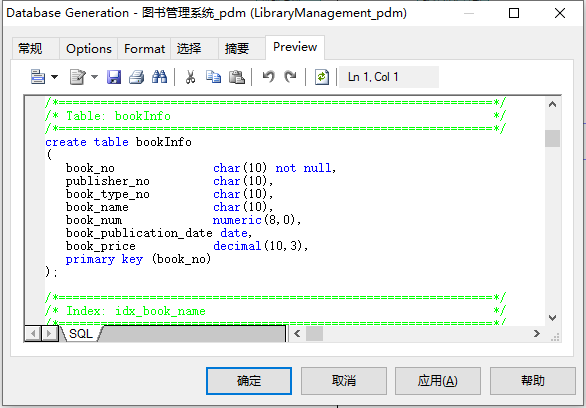
ORCL。

* + Generate name in empty comment：如果表、列、视图等模型对象的注释(Comment)为空，则生成的代码中将使用对象的名称作为注释。
* Encoding 改成 UTF-8，便于 MySQL Workbench 直接打开。

**Selection** 选项卡：

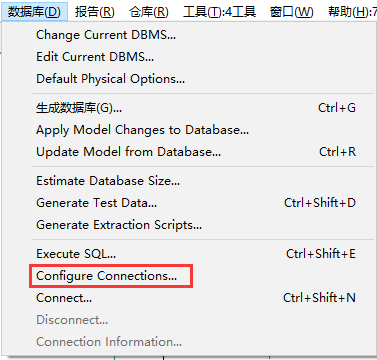


**Preview** 选项卡：可以在 Preview 中预览生成的 sql 内容

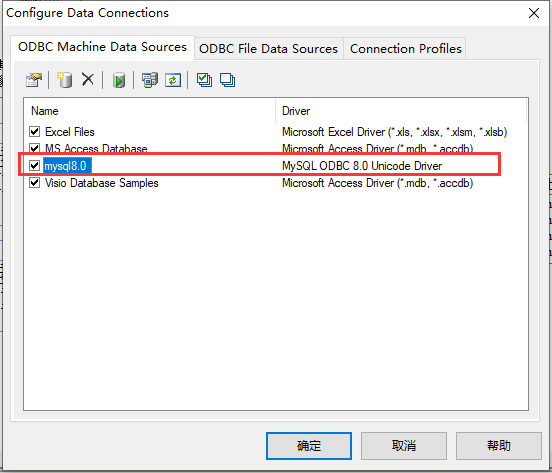


下面介绍如何在 Powerdesigner里连接数据库执行 sql 的方法

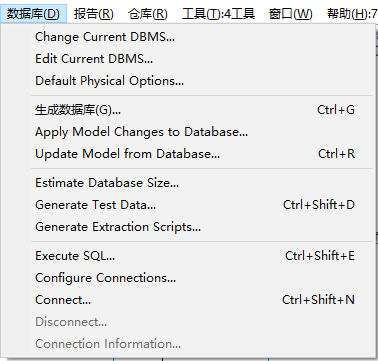
步骤1、在PowerDesigner中，配置数据库连接：



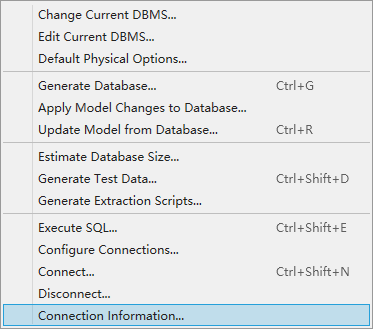
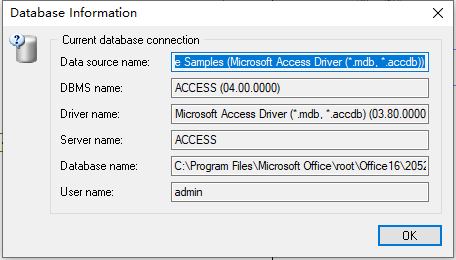
这里用 odbc 方式，先配置好了 odbc 数据源（步骤请自行百度）。



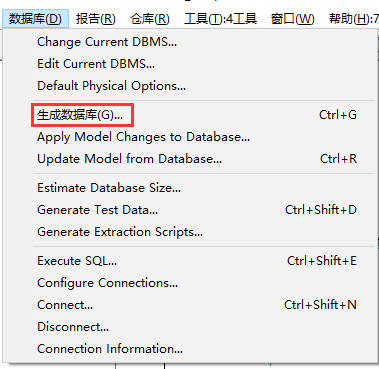
测试一下连接：

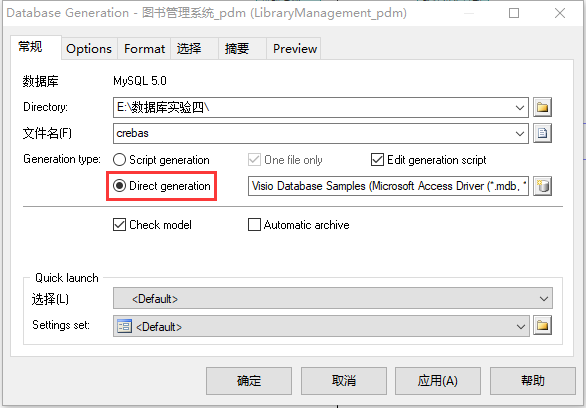


连接没有报错，然后看一下连接信息：

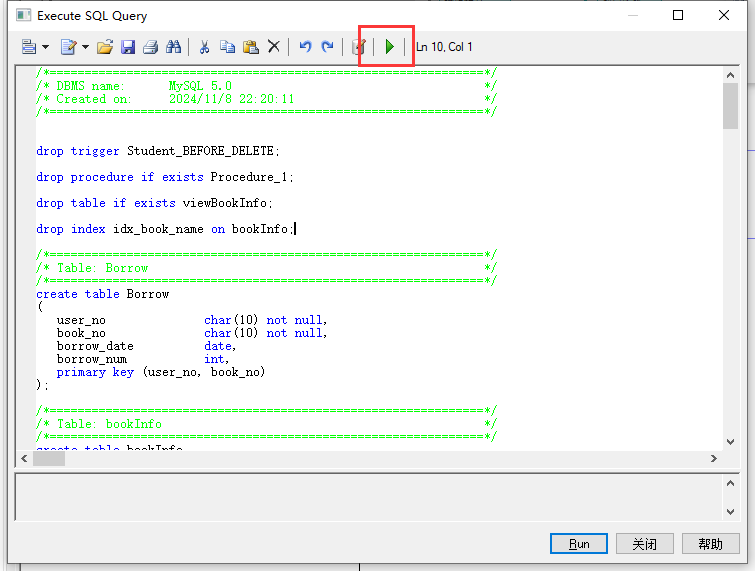


步骤 **2**：在 PowerDesigner 中，Generate Database：

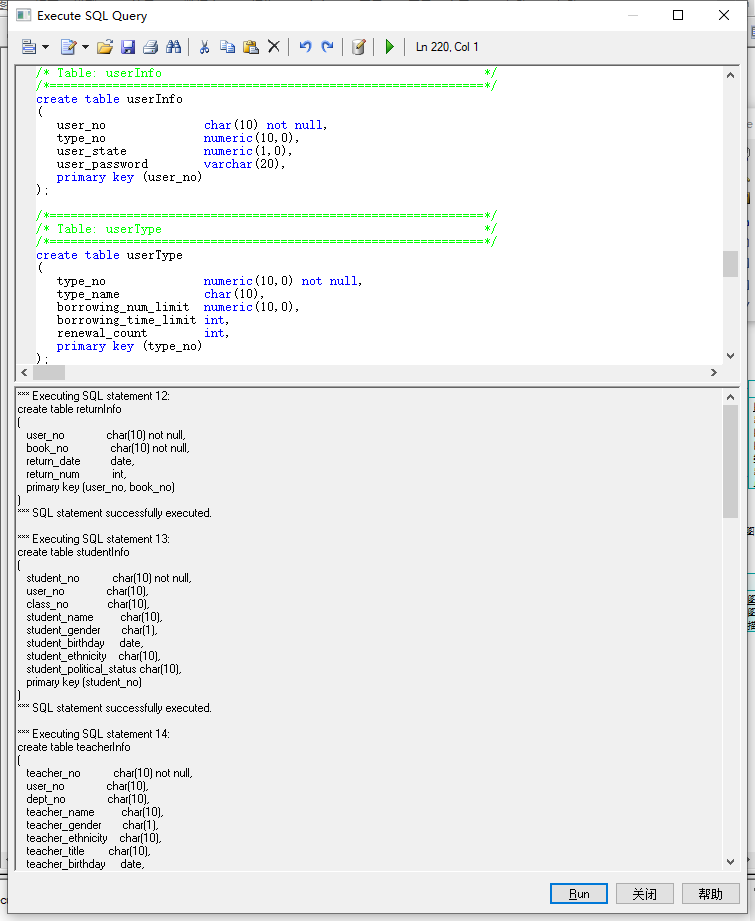




确定后出现 Execute SQL Query 交互框，可以修改 SQL 语句，比如增加建库语句：



运行完毕可以看到执行结果：



# 常见问题

* 生成的LDM或PDM缺少某个实体（例如：CDM中有实体，转成LDM或PDM实体消失），双击该实体，勾选“generate”方框， 既可以重新生成。
* PDM 存在表但不显示，在PDM 的 Diagram 上右键点击Show Symbols，在要显示的表前面打勾，点击OK即可。
* PDM 中 table 只显示字段的 name，想要显示 type 和其他信息，点击菜单栏“Tools”->“Preferences”,勾选“Data type”和其他信息，点击OK即可。
* 生成的 SQL 脚本在 MySQL 中执行报错，常见问题如下：
* 有重复的外键约束问题：可能ER 图 2 个实体是多对多的relationship，生成 pdm后新生成的table 两边的线条是同名的，所以生成 sql 语句后会报错。把其中一边的线条改成不重复的名字即可解决问题。
* 有重复的索引：问题可能出在：一个字段既是主键又是外键，因主键和外键都会生成索引，所以同一个字段有重复索引，删掉一个索引即可解决问题。
* 其他问题。可以查相关参考资料。

# 参考资料

Abraham Silberschatz, Henry F.Korth. 《数据库系统概念（第六版）》

《MySQL 中文参考手册》（MySQLBook.chm）