

哈尔滨工业大学（深圳）

数据库实验指导书

实验四：一个小型系统的设计与实现

目录

1	实验目的.....	3
2	实验环境.....	3
3	实验内容.....	3
3.1	实验任务.....	3
3.2	实验要求.....	3
3.3	选题说明.....	3
4	数据库设计.....	4
4.1	绘制 ER 图	5
4.2	CDM 转成 LDM	17
4.3	LDM 转成 PDM	23
4.3.1	创建视图.....	25
4.3.2	定义索引.....	27
4.3.3	定义触发器.....	31
4.4	PDM 转成 SQL 脚本.....	33
5	常见问题.....	45
5.1	PDM 存在表但不显示.....	45
5.2	生成的 SQL 脚本在 MySQL 中执行报错	46
5.3	PDM 中 table 只显示字段的 name，想要显示 type 和其他信息.....	47
5.4	生成的 SQL 执行报错：有重复的外键约束.....	48
6	参考资料.....	48

1 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。

2 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。

3 实验内容

3.1 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。

3.2 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、外键索引）、1 个触发器；
- 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。

3.3 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
实验室管理平台	1、 用户注册 （普通教师、实验室管理员）； 2、实验室管理员进行实验室维护（实验室名称、地址、实验室管理员、面积、台套数、电脑配置、安装的软件列表……）； 3、实验室管理员进行课程维护（课程名称、所属学院、授课教师、学时、人数、排课教室、课程所需软件列表……）； 4、实验室管理员进行软件维护（软件名称、类别、版本、软件架构、所需空间……）； 5、 普通教师预约实验室 （填写预约申请，包括实验室编号、预约时间、用途、联系方式）； 6、 实验室管理员收到预约申请后进行审批，通过或驳回申请。
校园食堂外送点餐系统	1、 用户注册 （普通用户、食堂管理员、商家）； 2、食堂管理员可进行食堂信息维护（新增、修改、删除）； 3、商家可进行商铺维护； 4、商家可进行菜品维护； 5、 普通用户下订单 （选择某个食堂、某个商家的某些菜品，设置数量，设置外送地址，设置联系方式……）； 6、 商户接单 （修改订单状态）。
校园猫管理平台	1、 用户注册 ； 2、查询、维护所有猫信息； 3、新增猫的信息（昵称、品种、颜色、经常出现的位置、性格……）； 4、 猫猫出现位置打卡 ； 5、 猫猫投喂登记 （选择猫猫、设置时间、选择地点、填写投喂物品）。
校园志愿者招募平台	1、 用户注册 （普通用户、管理员）； 2、管理员发布活动信息（时间、地点、人数、要求……）； 3、普通用户查看活动信息； 4、 普通用户申请参加活动 ； 5、 管理员可查询自己发布的活动的申请人清单，并对申请人进行审核 （通过或拒绝）； 6、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； 7、过期的活动设置为关闭状态。

4 数据库设计

备注：以下是 **PowerDesigner** 作为数据库设计工具进行示范。

PowerDesigner 是 Sybase 的企业建模和设计解决方案，采用模型驱动方法，将业务与 IT 结合起来，可帮助部署有效的企业体系架构，并为研发生命周期管理提供强大的分析与设计技术。

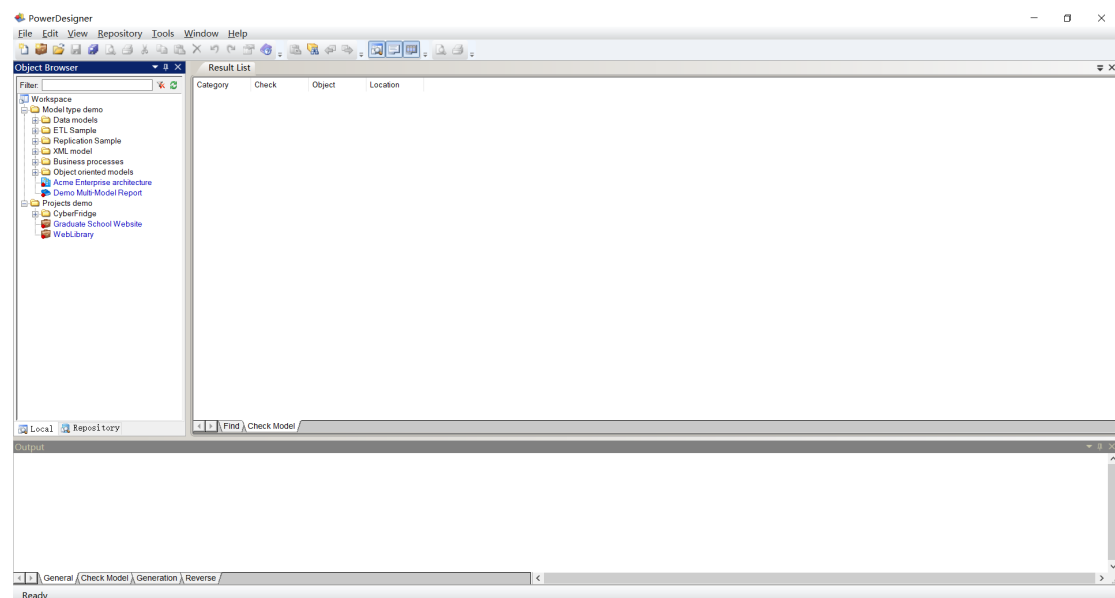
PowerDesigner 包含 Data Architect、Process Analyst、App Modeler 和 ODBC Administrator。其中 Data Architect 是一个强大的数据库设计工具，使用 Data Architect 可利用实体-关系图为一个信息系统创建“概念数据模型”-CDM（Conceptual Data Model）。并且可根据 CDM 产生“逻辑数据模型”-LDM（Logical Data Model）和基于某一特定数据库管理系统的“物理数据模型”-PDM(Physical Data Model)。还可由 PDM 产生为特定 DBMS 创建数据库的 SQL 语句并可以文件形式存储，以便在其他时刻运行这些 SQL 语句创建数据库。另外，Data Architect 还可根据已存在的数据库反向生成 PDM、LDM、CDM 及创建数据库的 SQL 脚本。

4.1 绘制 ER 图

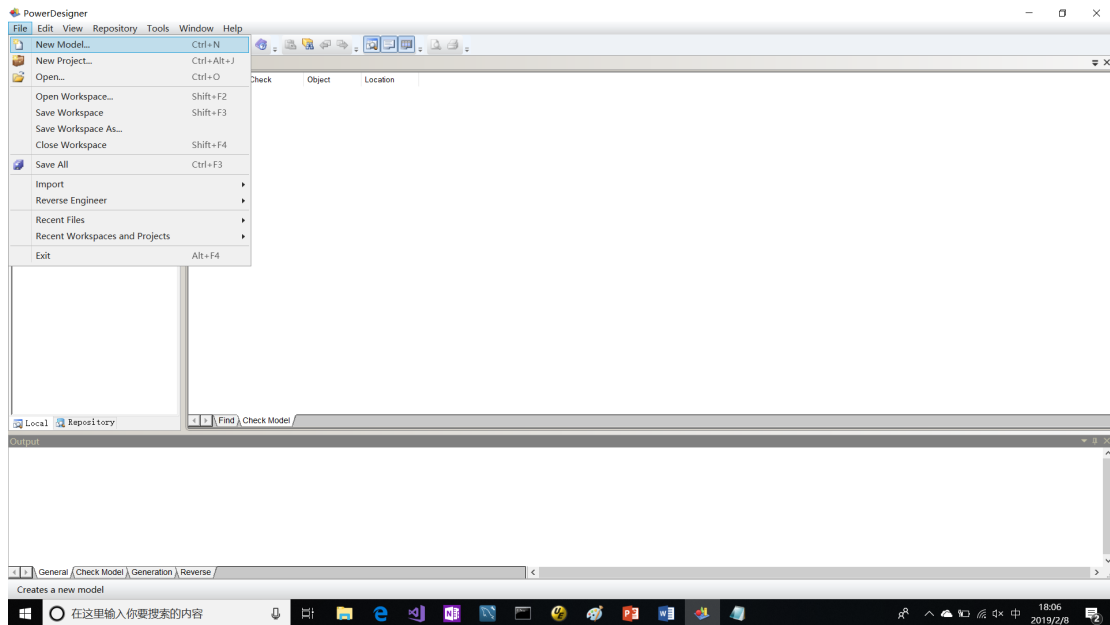
概念数据模型(conceptual data model，简称 CDM)，主要用在系统开发的数据库设计阶段，是按用户的观点来对数据和信息进行建模，并用实体-联系图(E-R 图)来体现。CDM 静态地描述系统中的各个实体以及相关实体之间的关系，系统分析员通过 E-R 图来表达对系统静态特征的理解。E-R 图实际上相当于对系统的初步理解所形成的一个数据字典，系统的进一步开发将以此为基础。

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，应该充分理解项目的软件需求说明书，区分实体和联系，为设计 E-R 图做好准备。这一步是数据库分析与设计的基本功。

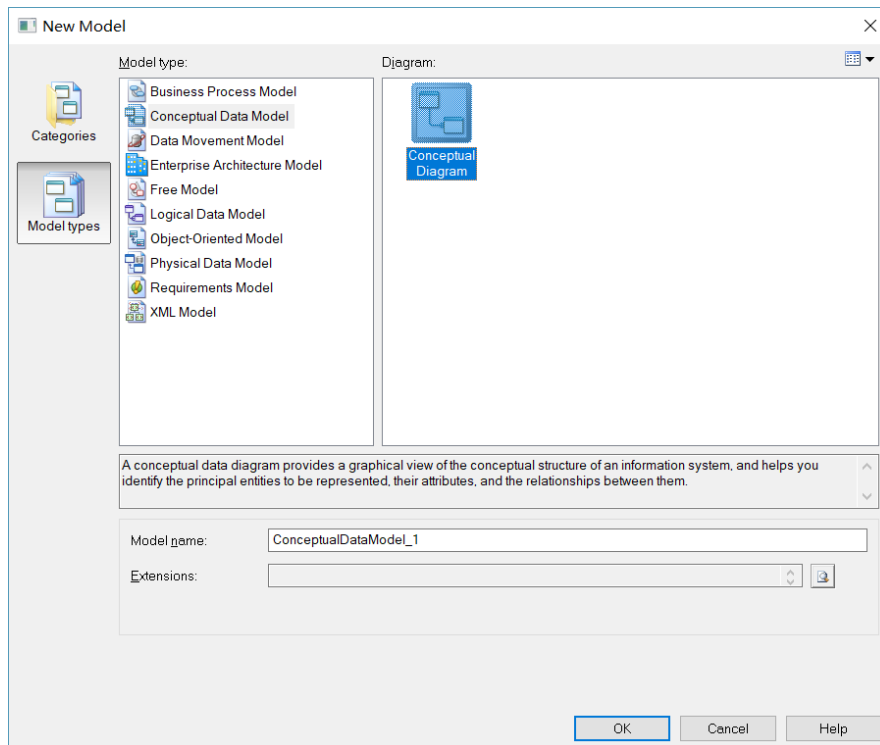
步骤 1. 启动 PowerDesigner，进入 PowerDesigner 的操作主界面，如下图所示。



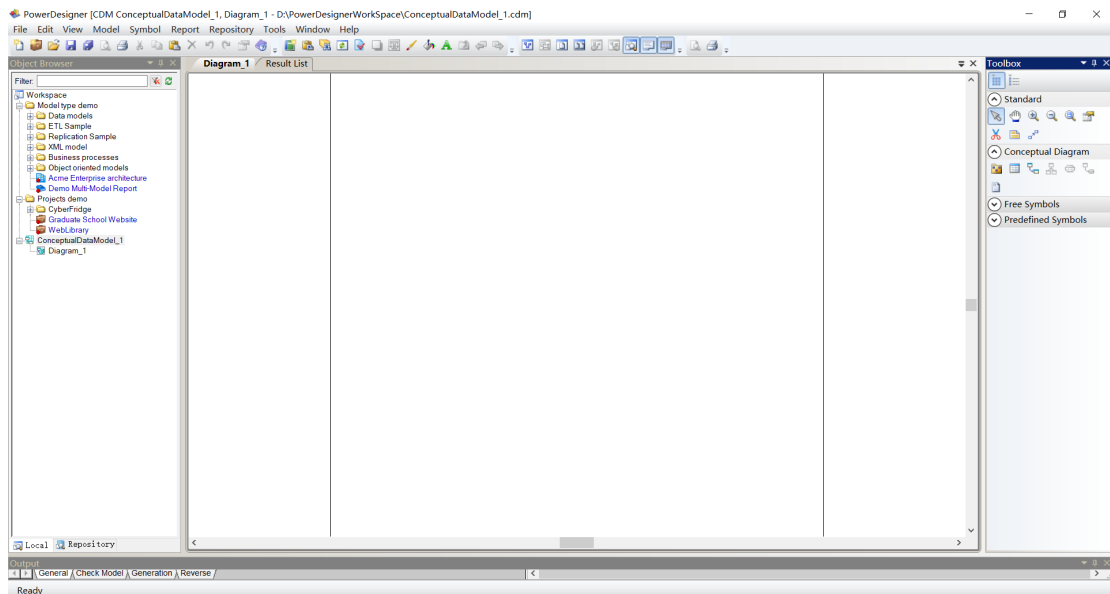
步骤 2: 在 File 菜单中单击 New Model…，如下图所示。



在 Model types (模型类型) 框中，可以看到：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。在此，我们选择 Conceptual Data Model 并单击“OK”按钮。

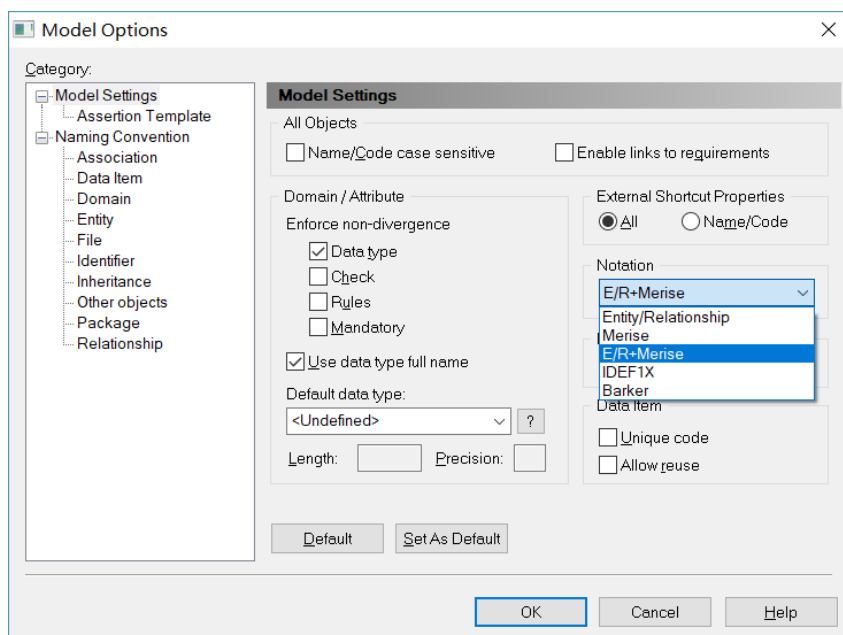


步骤 3：进入 PowerDesigner 的 CDM 操作窗口，如下图所示。




最常用的工具图标在 Toolbox 栏中，包括实体、关系、放大、缩小、移动等。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，画实体关系连线)即可。

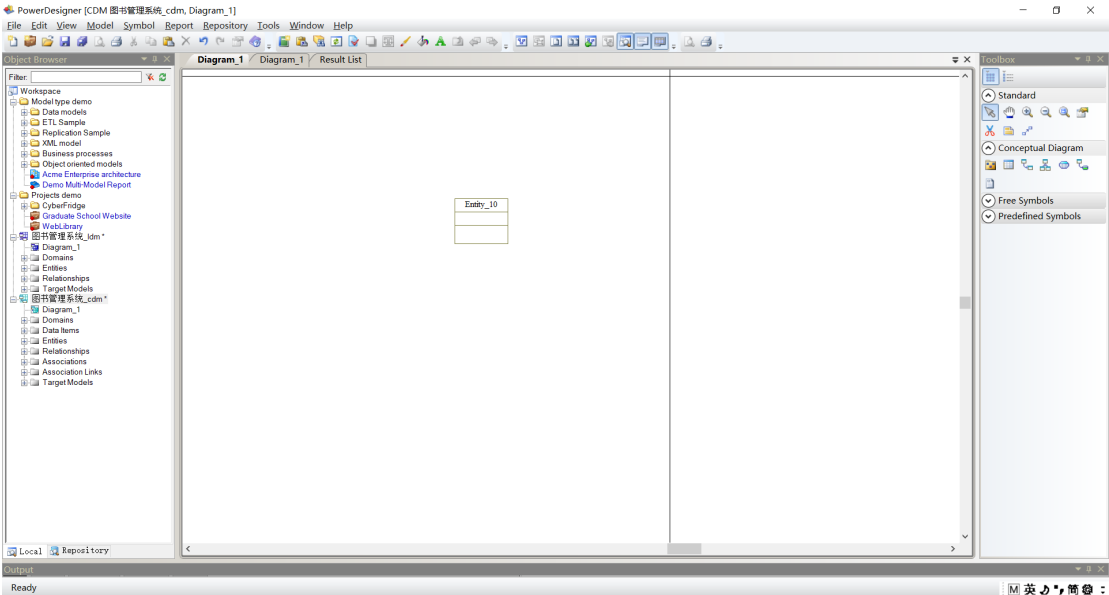
注意：PowerDesigner 16 中，模型选项 Notation 的设置不同，对象显示样式不同。可以通过 Tools→Model Options→Model Settings→Notation 对其进行修改。如下图。



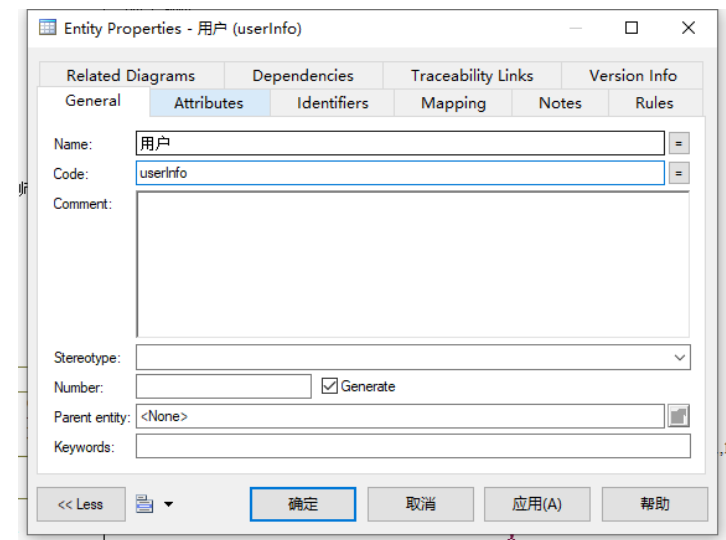
一般我们使用 ER 图作为概念模型图，所以建议选择 Entity/Relationship 或者 E/R+Merise, 这两者的区别是后者在 ER 模型的基础上还提供了 Merise 建模理论，允许在概念模型中使用 Association 和 Association Link。另外三个 Notation 中，Merise 使用 Association 完全代替了 Relationship；IDEF1X 是 IDEF 系列方法中 IDEF1 的扩展版本，是在 E-R(实体关系)方法的原则基础上, 增加了一些规则，使语义更为丰富的一种方法，其表现方式与一般的 ER 图表示有一点区别；Barker 比 ER 模型还要简化，只能使用“实体”和“关系”两个组件进行建模，

不能使用“继承”组件。一般情况下，使用 E/R 模型就够了，不过为了更好的表现实体之间的业务关系，有些时候还是使用 Association 来代替实体还是有一定的必要的，所以更建议选择 E/R+Merise 模型。

步骤 4：新增实体。Toolbox 中单击实体图标(Entity) ，然后在操作界面中单击 1 下，得到 1 个实体框。此时，这个实体还是空的，其名字可以临时任意选取，并且还没有属性。接下来，要逐步对实体的名字及其属性进行定义。



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 选项卡用于设置实体包括的属性(字段)信息；Identifiers 选项可用于设置实体标识符；Notes 选项卡用于设置实体的描述信息；Rules 选项卡用于设置与该实体相关的业务规则。

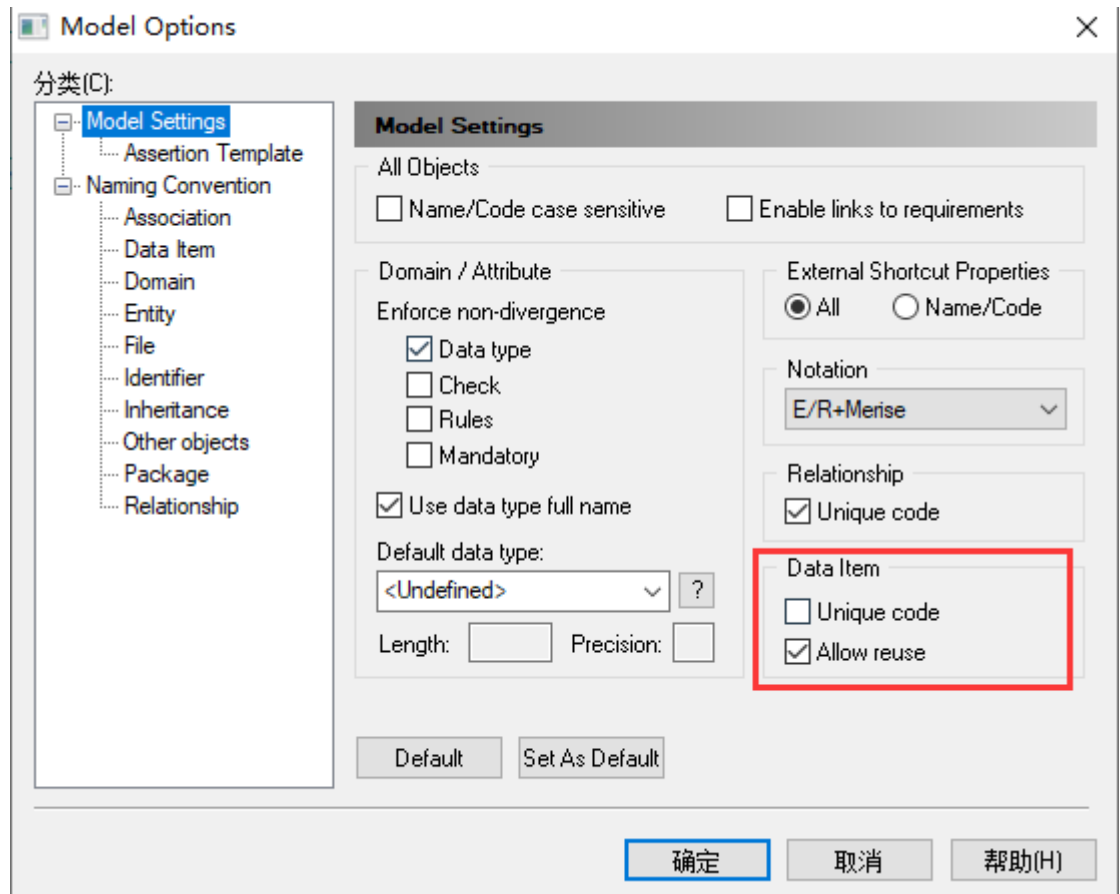
此例中，对 Name 栏输入“用户”，在 Code 处输入“userInfo”。

注意：

1、默认 Code 与 Name 是联动修改的，修改了 Name，Code 随之修改。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、默认一个 ER 图中所有实体的属性是全局唯一的。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。

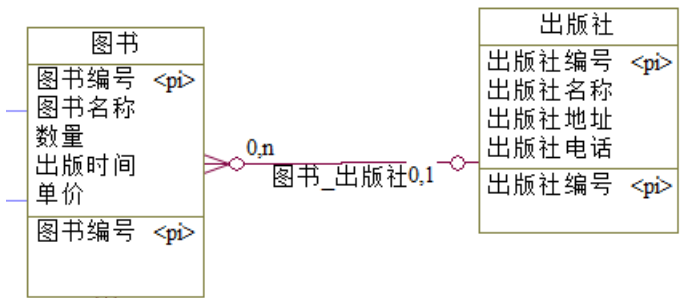


- 3、code 不要使用 MySQL 的关键字，比如 return、describe 等。可以使用下划线“_”，不要使用“-”。

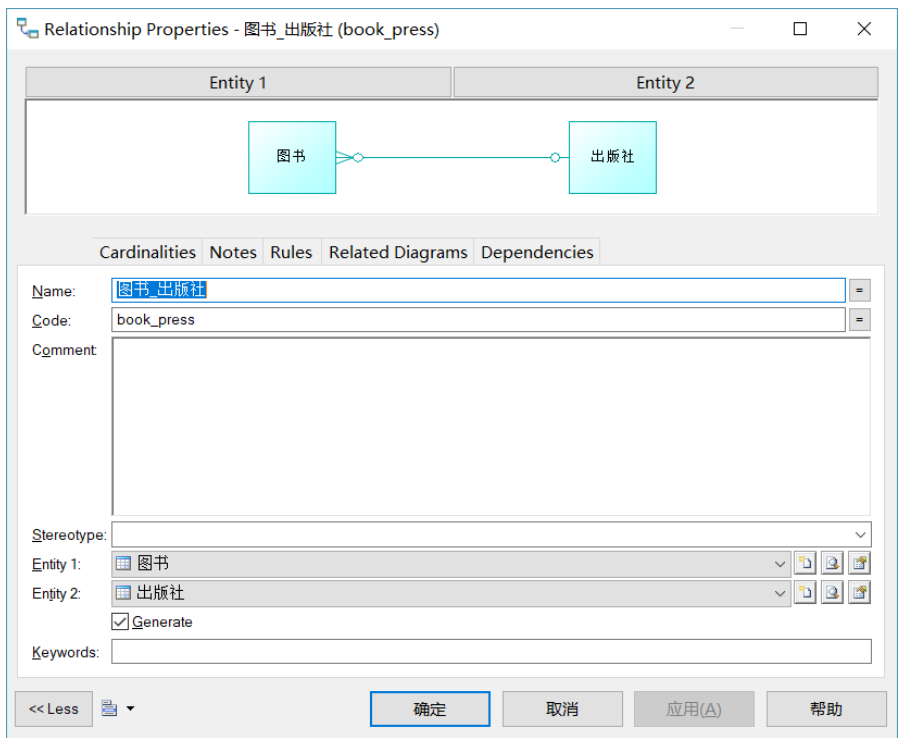
步骤 6： 定义属性、属性的约束和算法。单击 Attributes 按钮，进入定义该实体的属性界面，如下图所示。

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”。

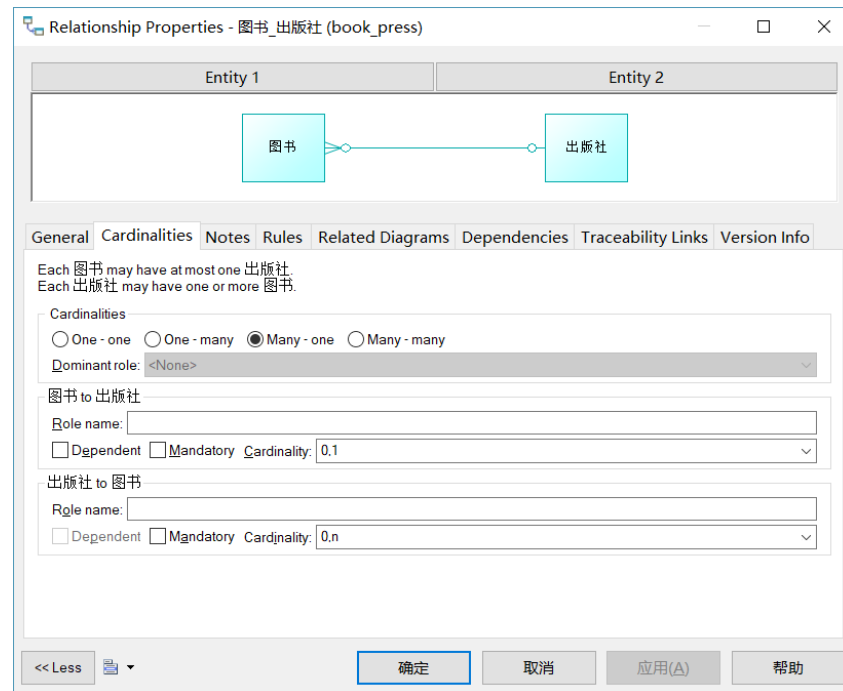


General 选项卡用于设置联系的基本信息，主要参数含义如下：

- 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

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”。



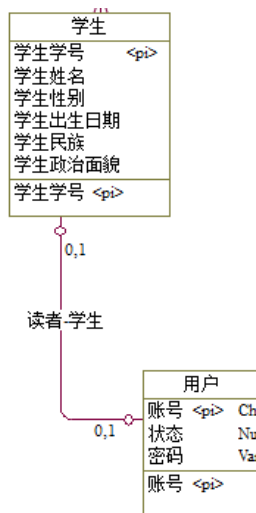
其中：

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

注意：Each 图书 may have at most one 出版社.
Each 出版社 may have one or more 图书. 这个地方会根据你的选择而改变，可以通过读这个信息来确认联系和基数是否选择恰当。

这里提醒一下大家，两个实体间设置了联系，如果 Cardinalities 设置为 One-many、Many-one，在你生成的 PDM 中会自动把一个实体的主键作为另外一个实体的外键！如果是 One-one，请看下面 Dominant role 设置。如果是 Many-many，哈哈，你能猜到转成 PDM 是什么样子的吗？猜不到那就把它转成 PDM 看看。（这里有个小技巧，同学们可以边画 CDM，边转成 PDM，验证自己的 CDM 的正确性）

- Dominant role：该参数只针对“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。
比如 CDM 中有学生和用户的关系如下图：



如果设置 Dominant role 为用户→学生:

Relationship Properties - 用户-学生 (user_student)

Entity 1: 学生 Entity 2: 用户

General Cardinalities Notes Rules Related Diagrams Dependencies Traceability Links Version Info

Each 学生 may have at most one 用户.
Each 用户 may have at most one 学生.

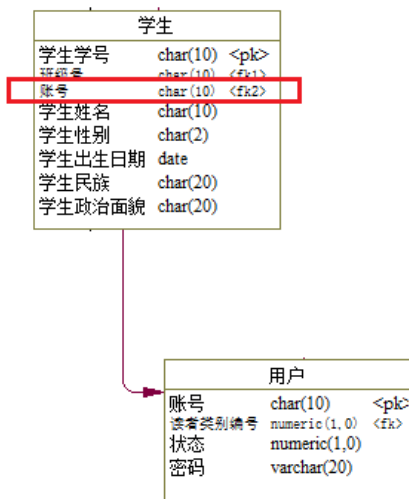
Cardinalities
☒ One - one ☐ One - many ☐ Many - one ☐ Many - many
 Dominant role: 用户 -> 学生

学生 to 用户
 Role name:
☐ Dependent ☐ Mandatory Cardinality: 0,1



用户 to 学生
 Role name:
☐ Dependent ☐ Mandatory Cardinality: 0,1

<< Less 确定 取消 应用(A) 帮助

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:

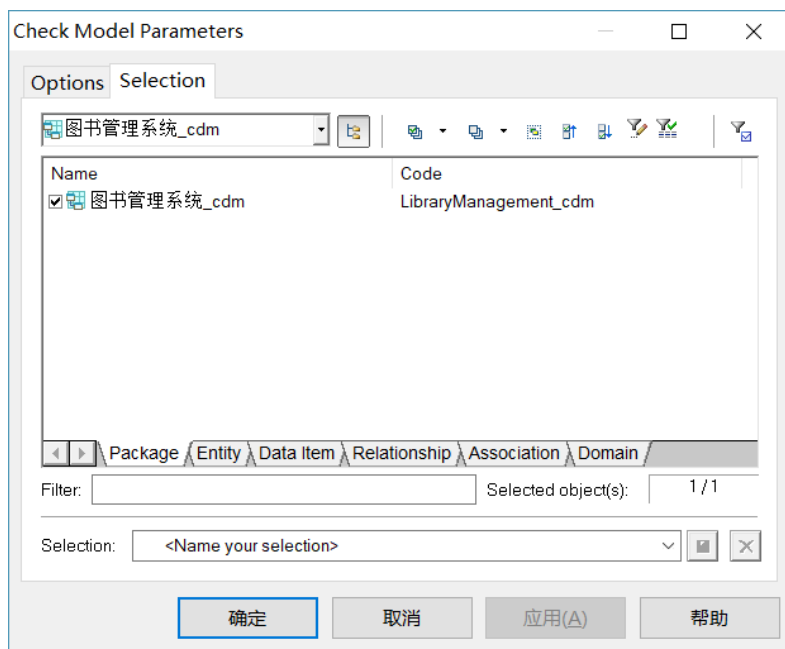
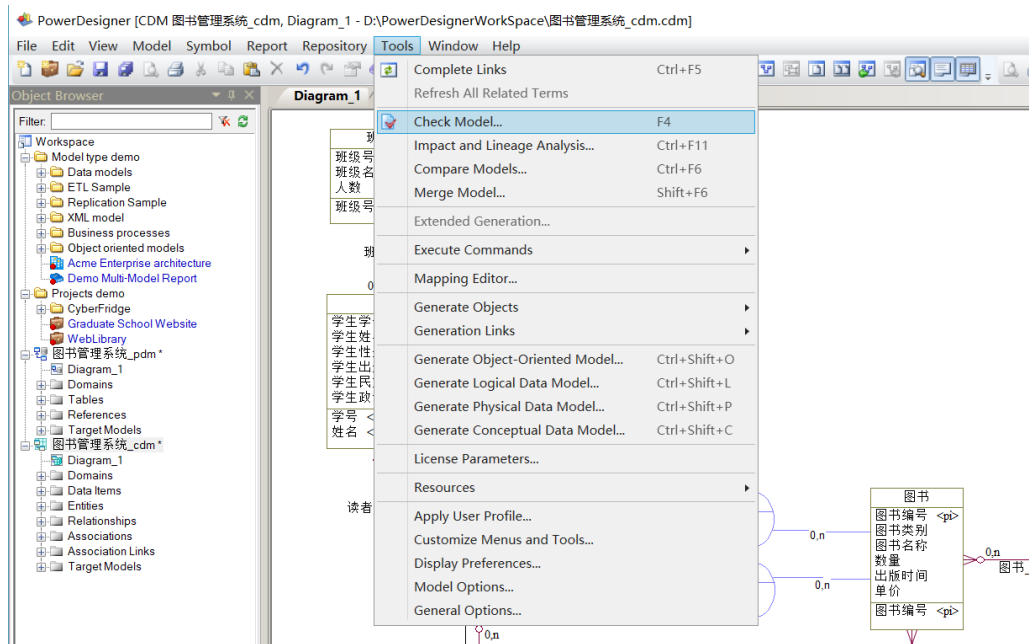


- **Dependent:** 依赖关系。表示实体所包含的基本信息必须依赖于另一个实体的基本信息。如果定义了依赖关系，在 CDM 生成 PDM 时，依赖实体中将生成一个引用，并且该引用将作为依赖实体标识符的一部分。两个实体之间存在依赖关系的联系又称为标定联系；如果没有定义依赖，则称为非标定联系。
- **Mandatory:** 强制。强制状态下联系的基数分为“1, 1”和“1, n”两种。

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：




```
Output
Checking relationship ...
- Relationship name uniqueness
- Relationship code uniqueness
- Reflexive dependency
- Reflexive mandatory
- Bijective relationship between two entities
- Name uniqueness constraint between many-to-many relationships and entities
Checking association ...
- Association name uniqueness
- Association code uniqueness
- Number of links >= 2
- Number of links = 2 with an identifier link
- Number of identifier links <= 1
- Absence of properties with identifier links
- Bijective association between two entities
- Maximal cardinality links
- Reflexive identifier links
- Name uniqueness constraint between many-to-many associations and entities

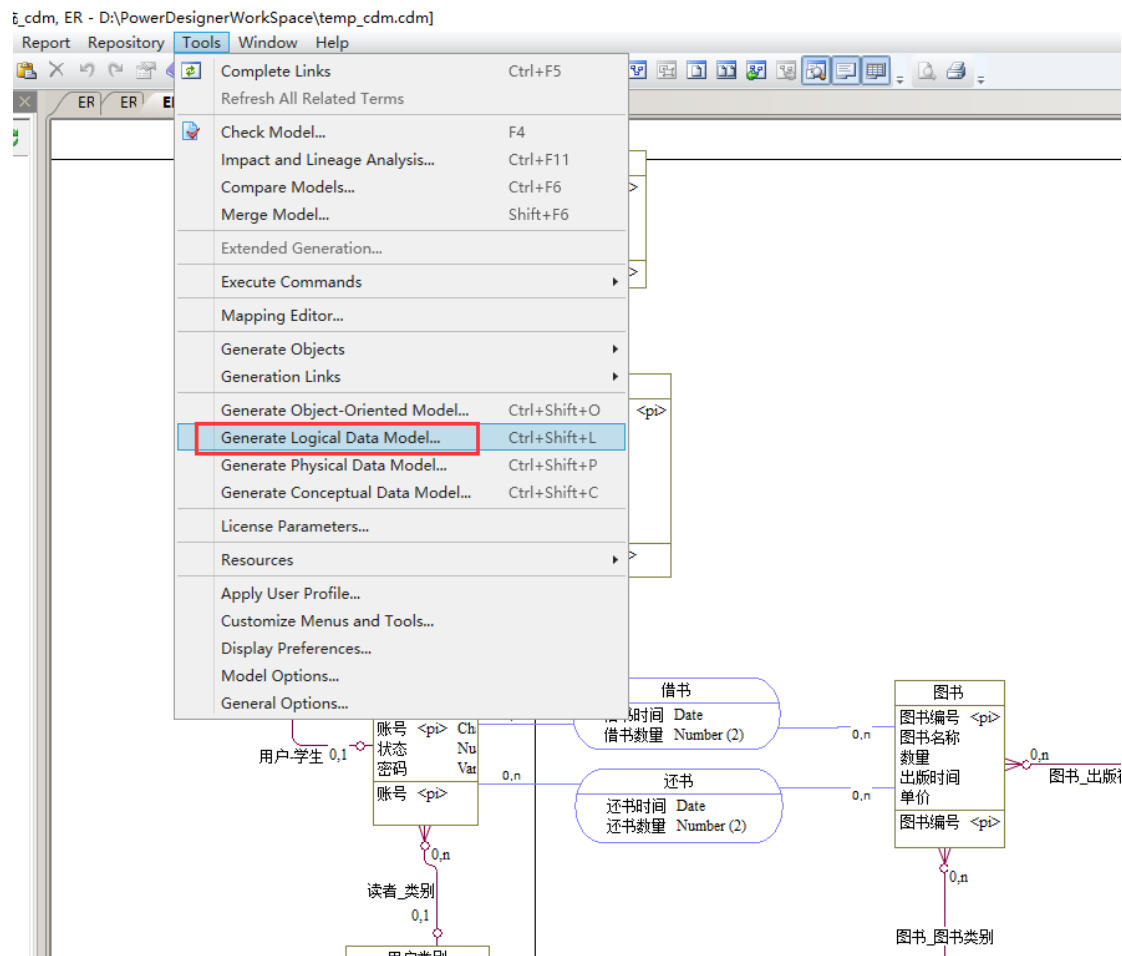
0 error(s). 0 warning(s).
The Conceptual Data Model is correct, no errors were found.
```

检查结果分为没有错误、错误和警告三类：错误是必须要改正的，警告（例如，一个实体有外键而无主键）可以改正也可以不改正。若没有错误，则保存此 E-R 图。至此，概念数据模型 CDM 已经完成。

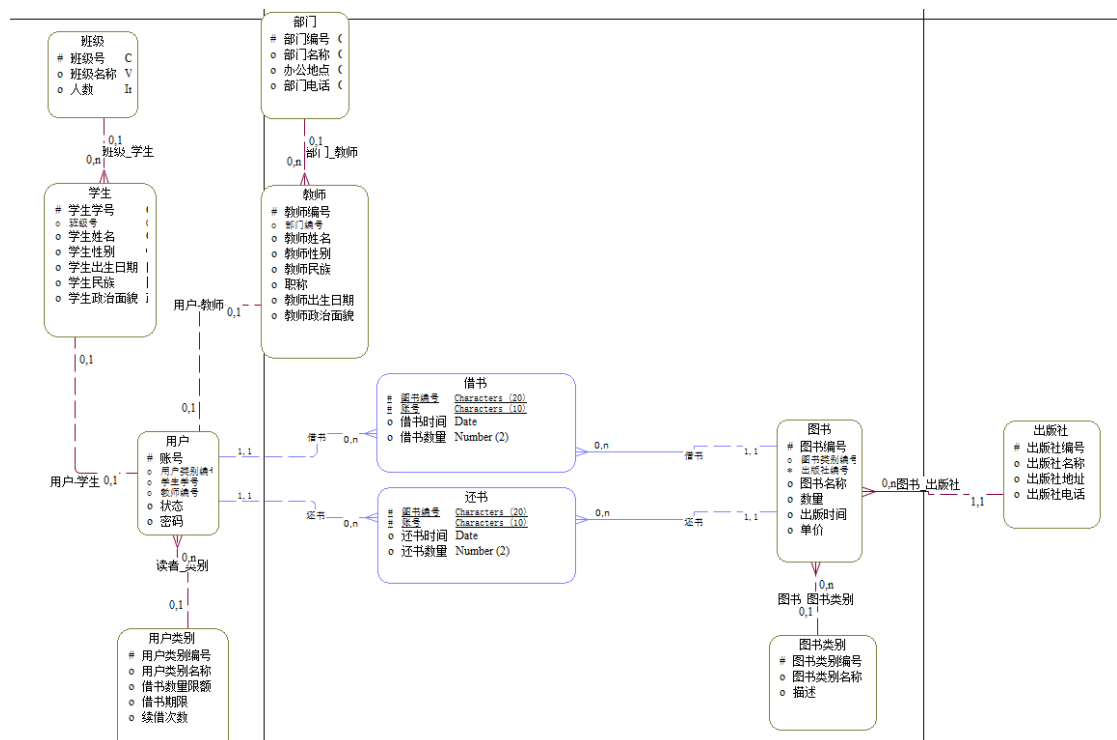
4.2 CDM 转成 LDM

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。

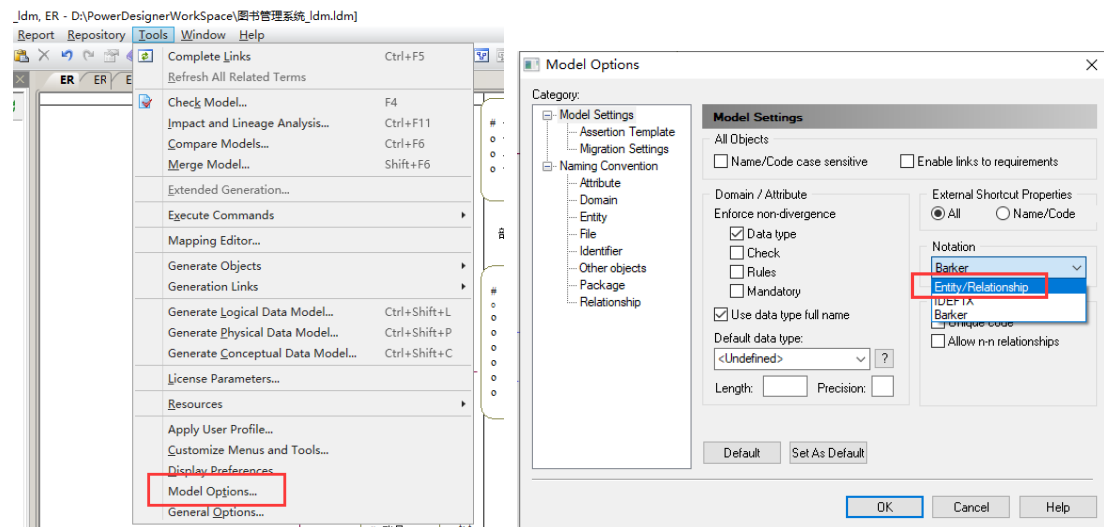
步骤 1：在 Tools 菜单中单击 Generate Logical Data Model 命令，打开逻辑数据模型设置窗口，如下图所示。



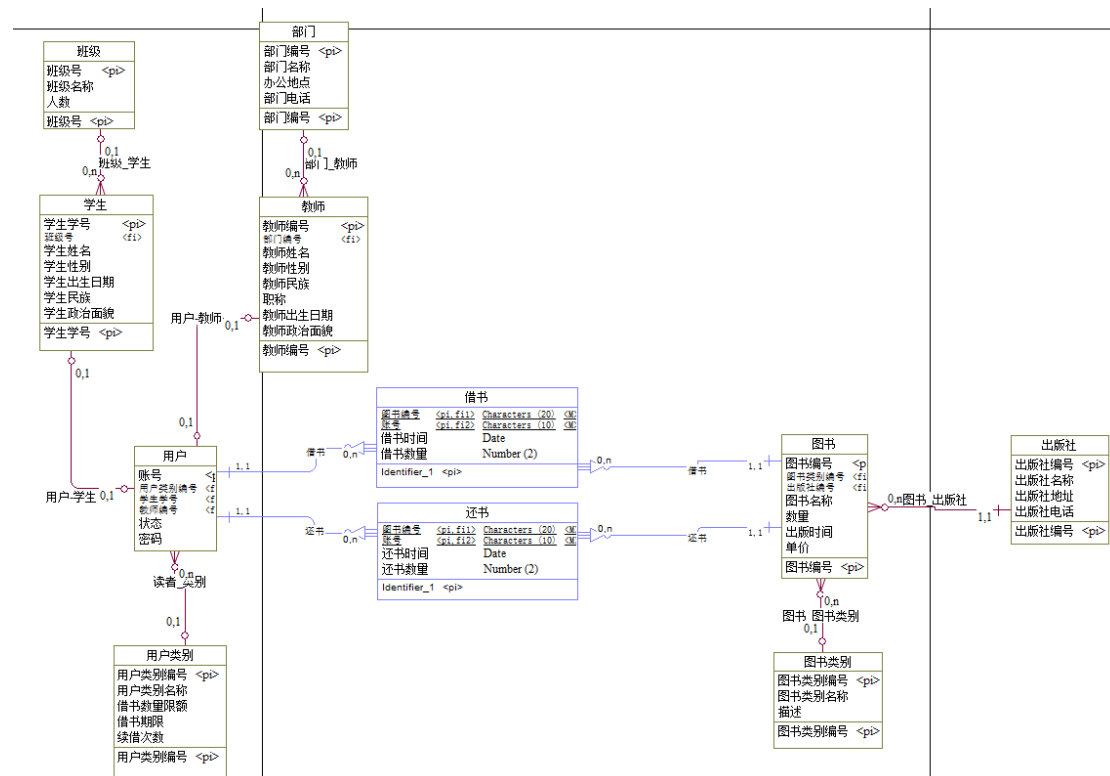
步骤 2: 默认转换成的 LDM, notation 是 Baker 方式:



我们可以修改成我们习惯的 Entity/Relationship 类型:

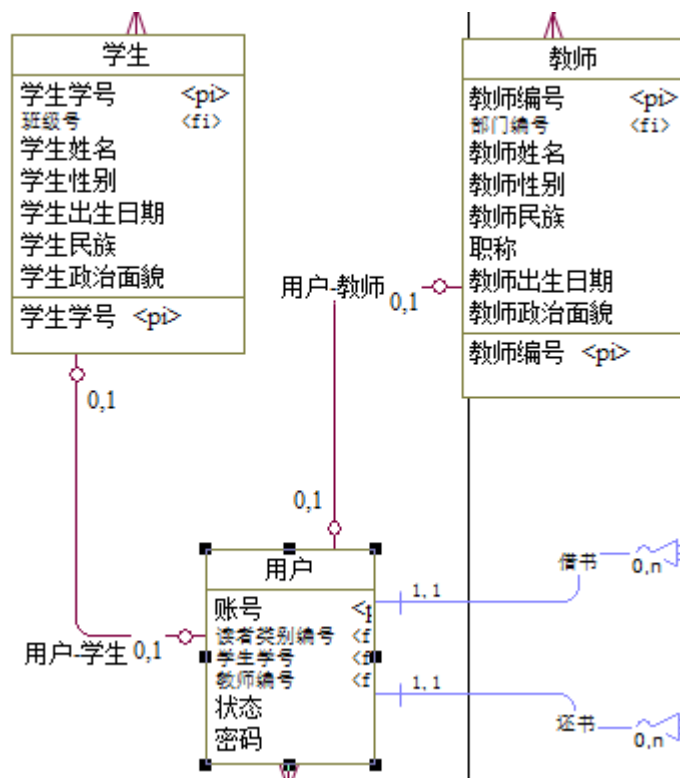


OK 之后：



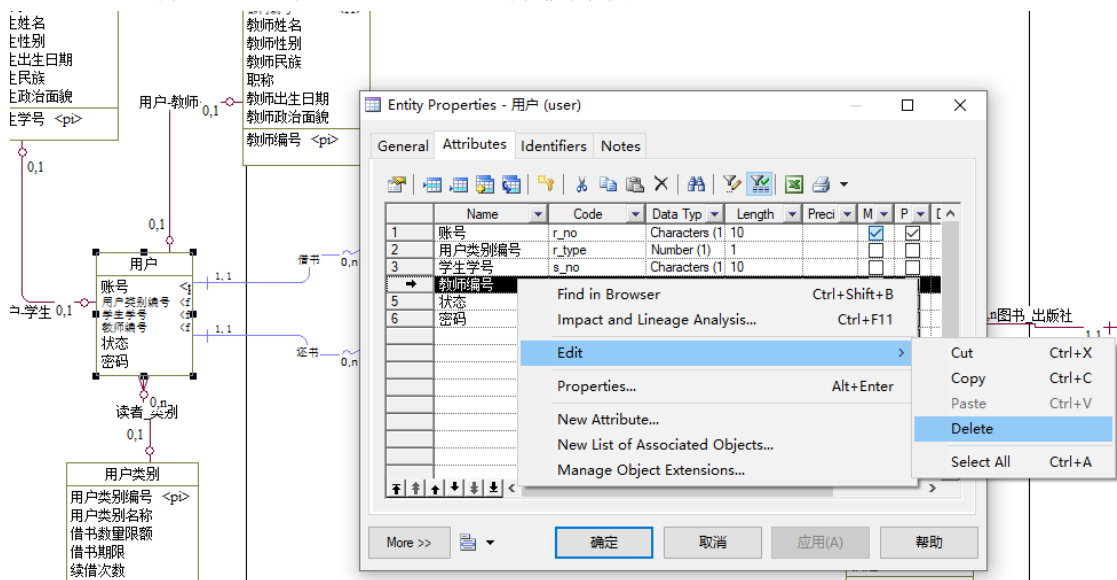
步骤 3：可以利用鼠标拖动实体框和关联线，对 LDM 图形进行调整，直到图形整齐、美观为止，然后进行保存。

步骤 4：我们检查 LDM 图，注意到用户实体的属性出现了学生学号、教师编号：

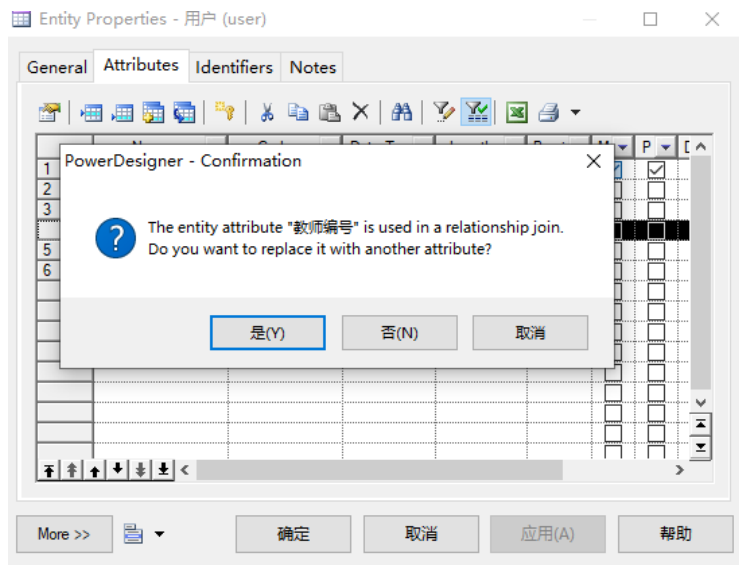


实际上这里可以把这 2 个属性合并成一个编号，如果用户类别是学生，那么编号存储的是学生学号；如果用户类别是教师，那么编号存储的是教师编号。我们可以这样操作：

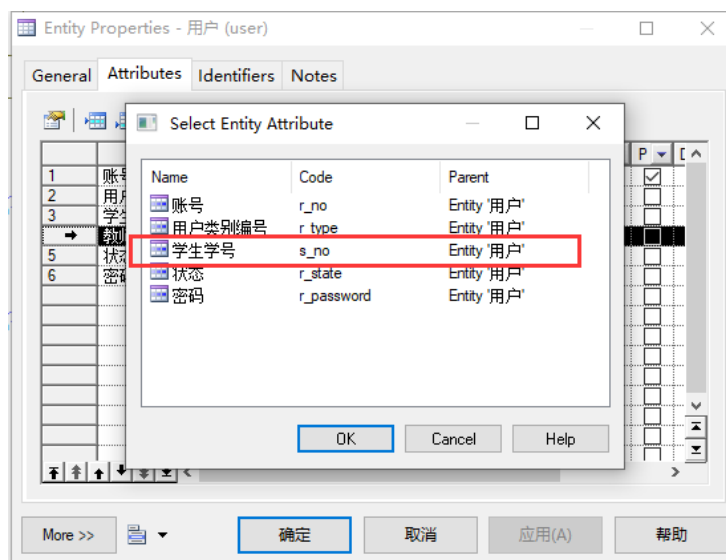
1. 我们删掉其中一个属性，比如删掉教师编号：



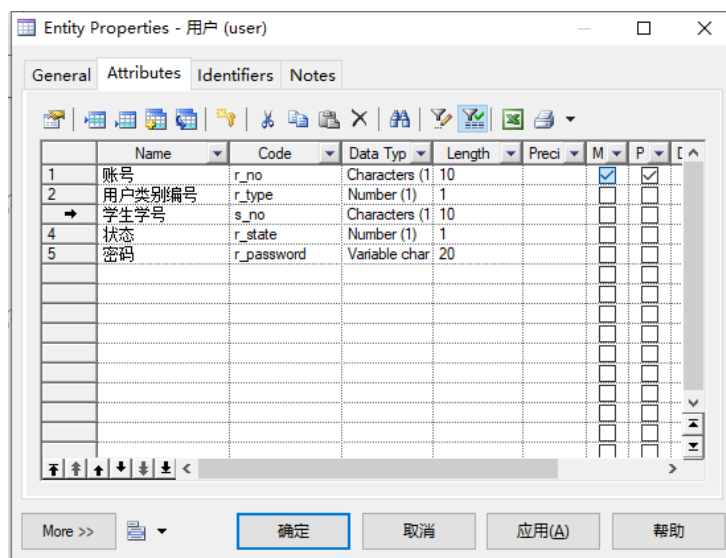
2. 因为教师编号是有关联到教师实体的，所以删除的时候会提示是否要关联到另外一个属性，选择“是”：



把原来关联到教师编号的 relationship join 到学生学号上去:



Ok:



3. 把学生学号改成学生教师编号:

Entity Properties - 用户 (user)

General Attributes Identifiers Notes

	Name	Code	Data Type	Length	Preci	M	P	
1	账号	r_no	Characters (1: 10			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	用户类别编号	r_type	Number (1)	1		<input type="checkbox"/>	<input type="checkbox"/>	
3	学生教师编号	st_no	Characters (1: 10			<input type="checkbox"/>	<input type="checkbox"/>	
4	状态	r_state	Number (1)	1		<input type="checkbox"/>	<input type="checkbox"/>	
5	密码	r_password	Variable char: 20			<input type="checkbox"/>	<input type="checkbox"/>	

More >> 确定 取消 应用(A) 帮助

4. 我们检查用户与学生实体间的关联关系, 可以看到学生-用户之间的关联关系表达正确:

Relationship Properties - 用户-学生 (user_student)

Entity 1: 用户 Entity 2: 学生

General Joins Cardinalities Notes

Parent: 学生学号

Parent Attribute	Child Attribute
学生学号	学生教师编号

More >> 确定 取消 应用(A) 帮助

我们检查用户与教师实体间的关联关系, 可以看到教师-用户之间的关联关系表达正确:

Relationship Properties - 用户-教师 (user_teacher)

Entity 1: 用户 Entity 2: 教师

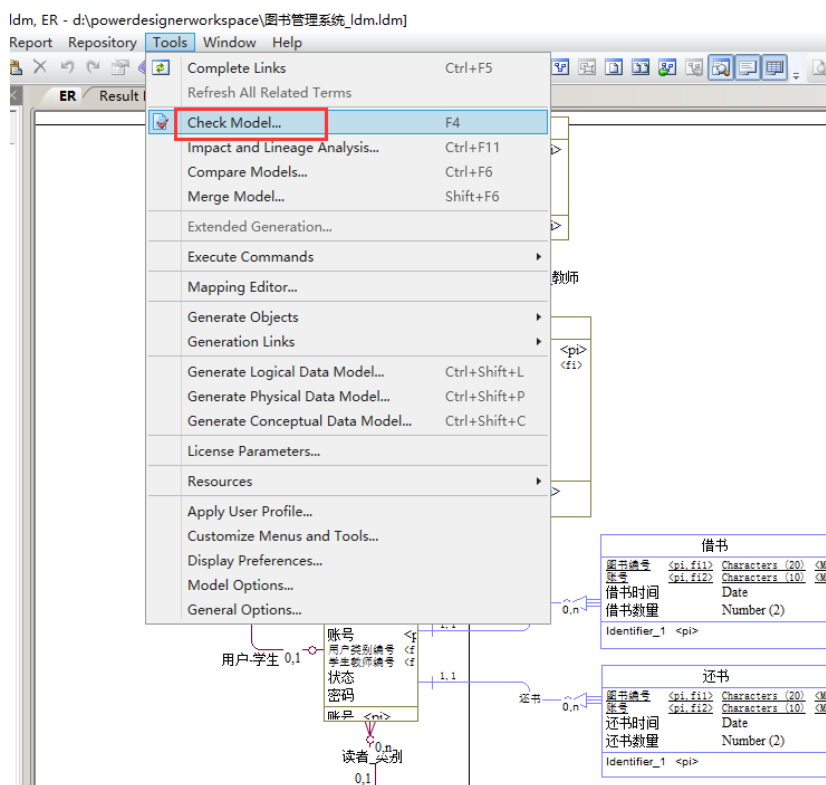
General Joins Cardinalities Notes

Parent: 教师编号

Parent Attribute	Child Attribute
教师编号	学生教师编号

More >> 确定 取消 应用(A) 帮助

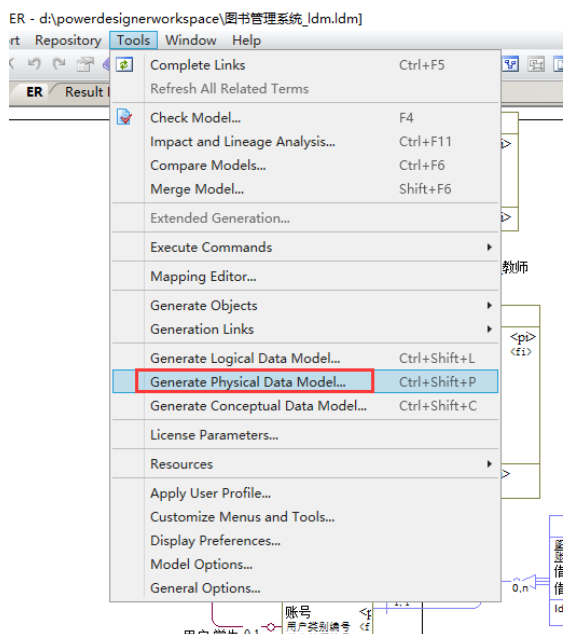
步骤 5: 在 Tools 菜单中单击 Check Model 命令来检查 LDM 图的错误。

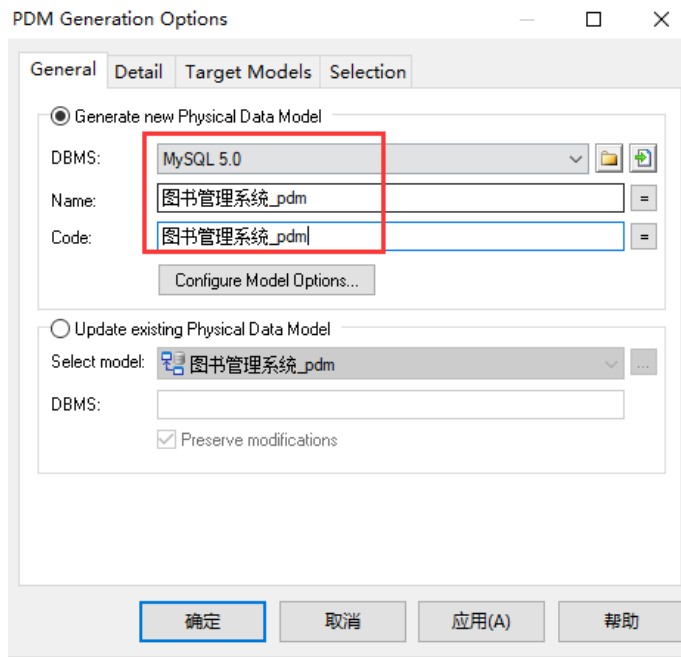


4.3 LDM 转成 PDM

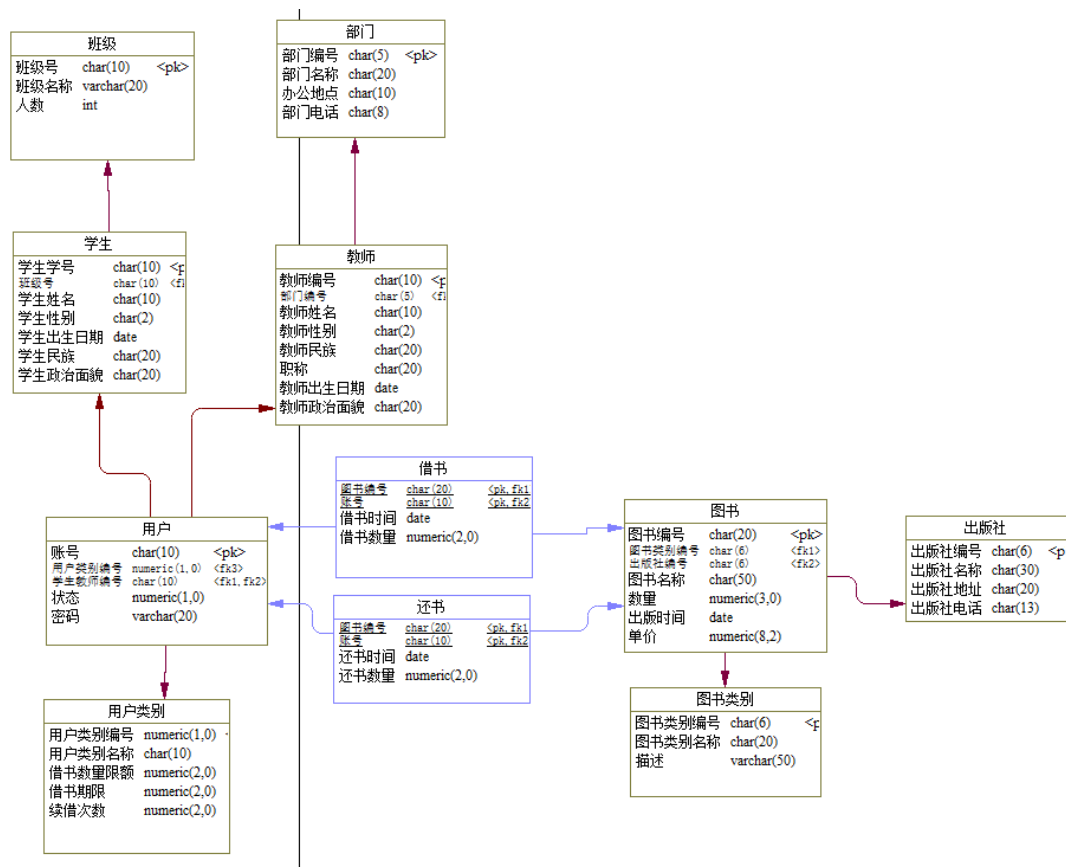
有了 LDM 模型之后，可以利用系统提供的自动转换功能将 LDM 模型转换成 PDM 模型，而不需要重新定义。（也可以由 CDM 图直接转成 PDM 图，再在 PDM 图基础上进行修改）

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。
 - 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。

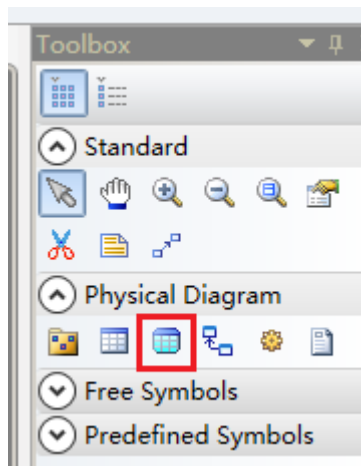


步骤 2：可以利用鼠标拖动实体框和关联线，对 PDM 图形进行调整，直到图形整齐、美观为止，然后进行保存。从上图可见，PDM 与物理建表已经很接近。
注意：同学们可以认真分析 CDM、LDM、PDM 图的区别，仔细看看 CDM 里的实体、关系是如何转成 PDM 中的表。

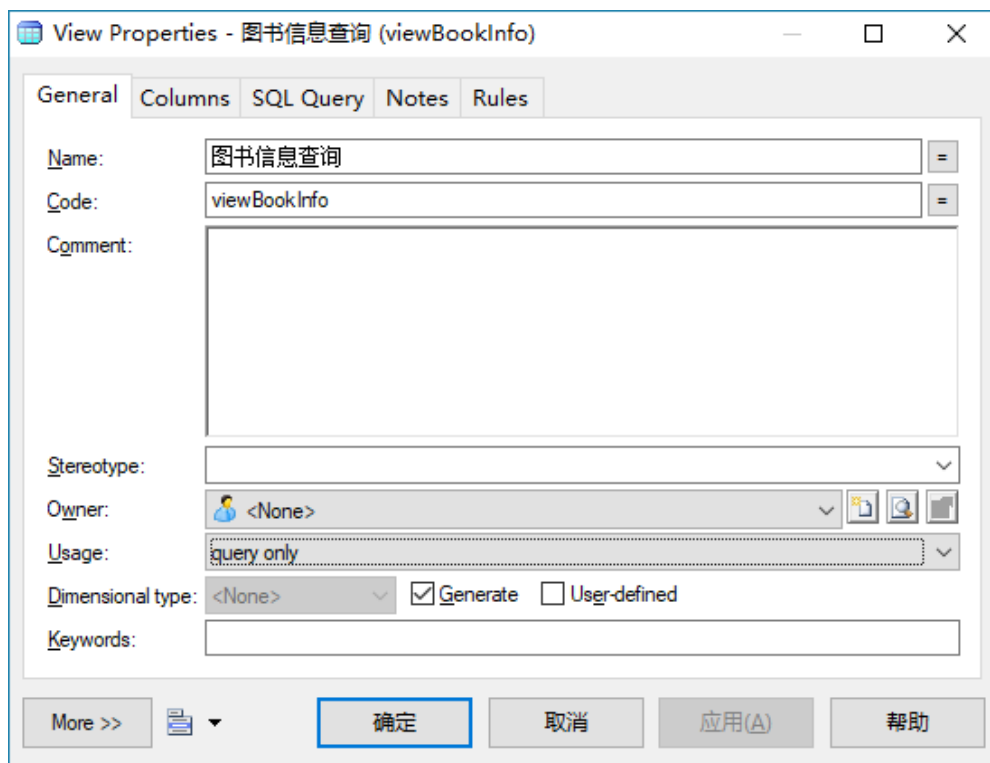
下面介绍如何创建视图、定义索引、定义触发器。

4.3.1 创建视图

1、单击 Toolbox 里面的视图图标：

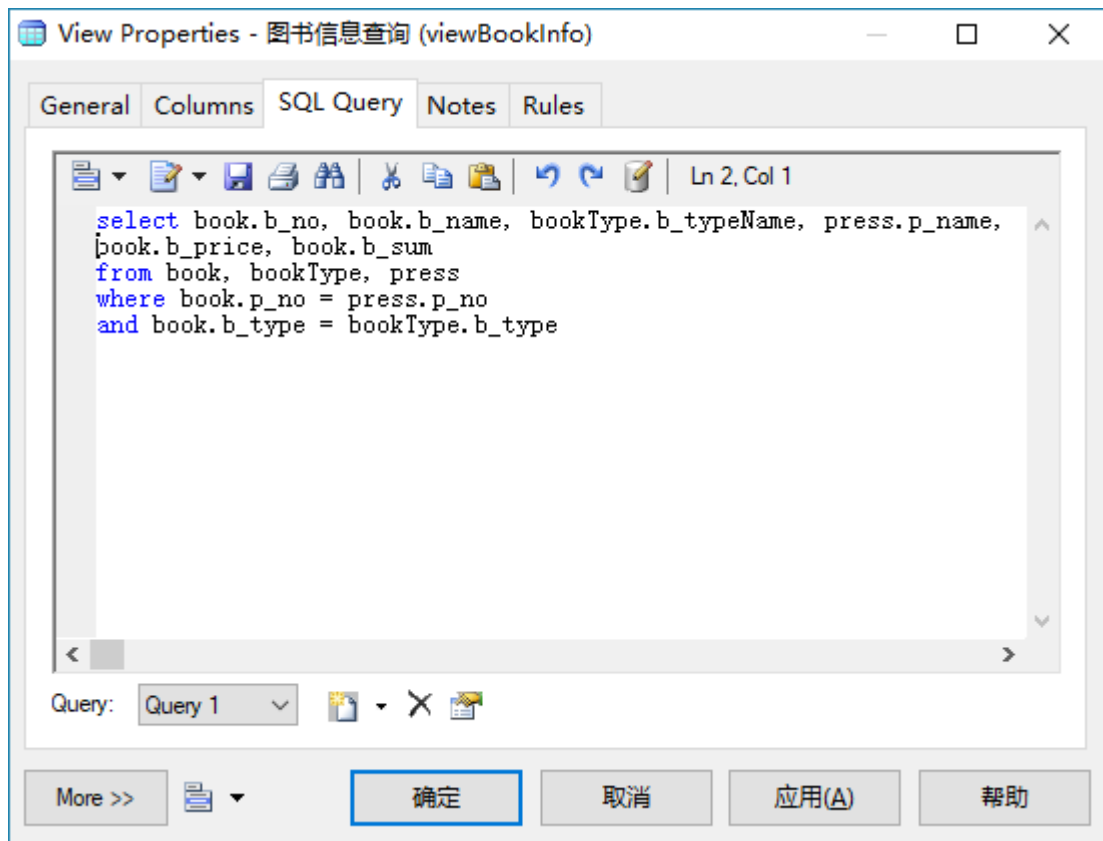


2、设置视图名字、code 等属性：

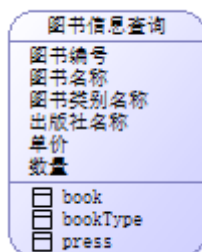


注意：Usage 是表示视图是只读的视图还是可更新的视图，如果我们只创建一般的视图，那么就选择 query only 选项。

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*，而是使用列名，这样在视图属性的 Columns 中才能看到每个列。如下图：



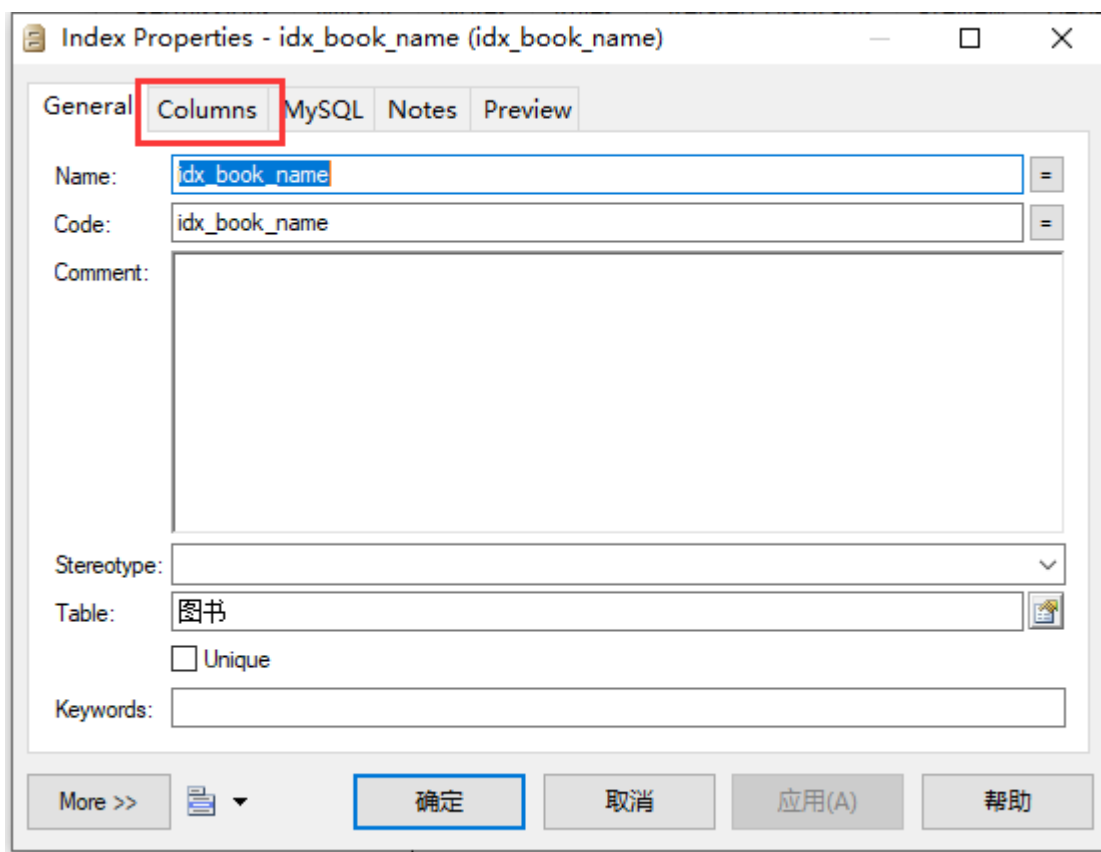
确定后视图建好了：



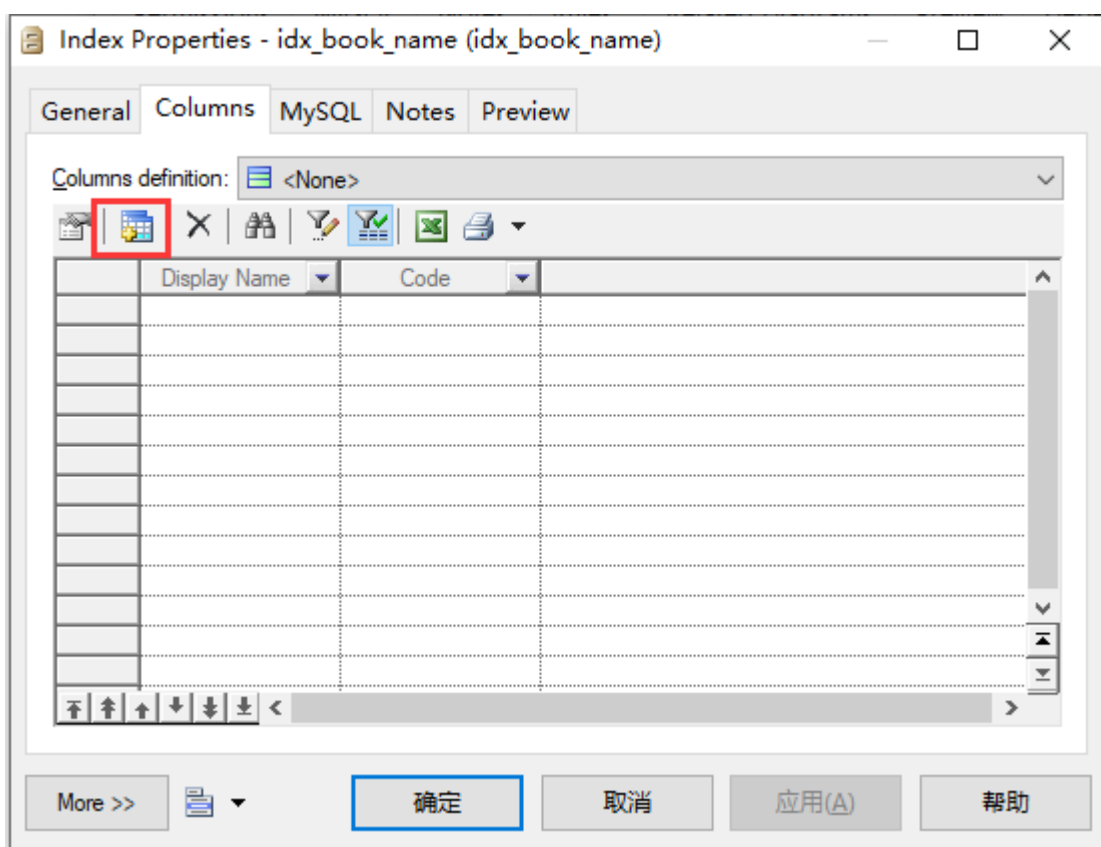
4.3.2 定义索引

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：



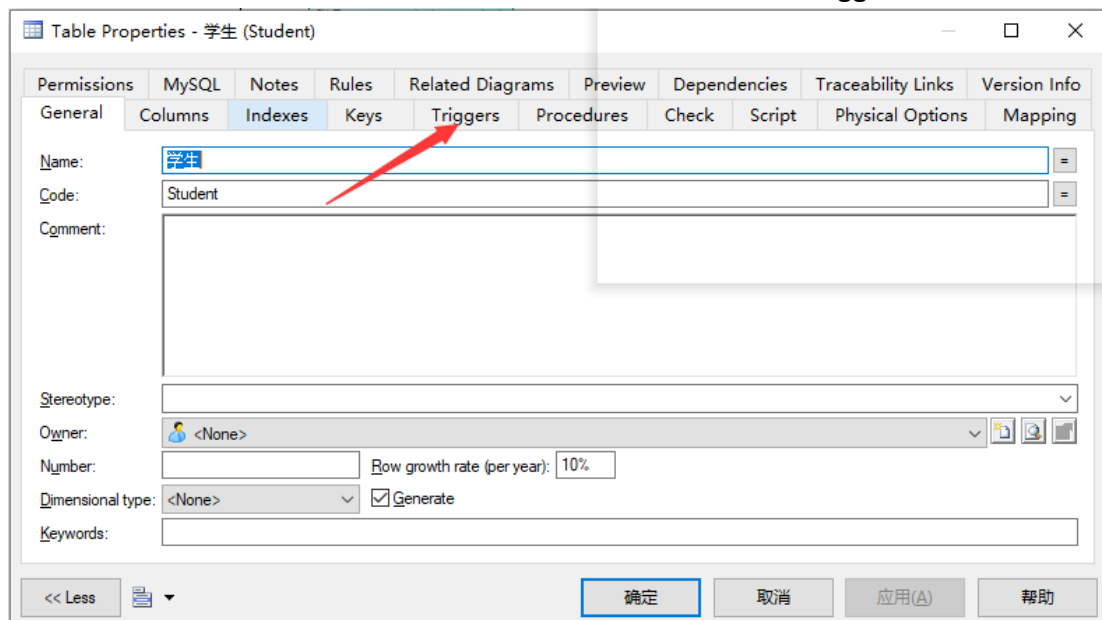
4、 点击 Add columns:



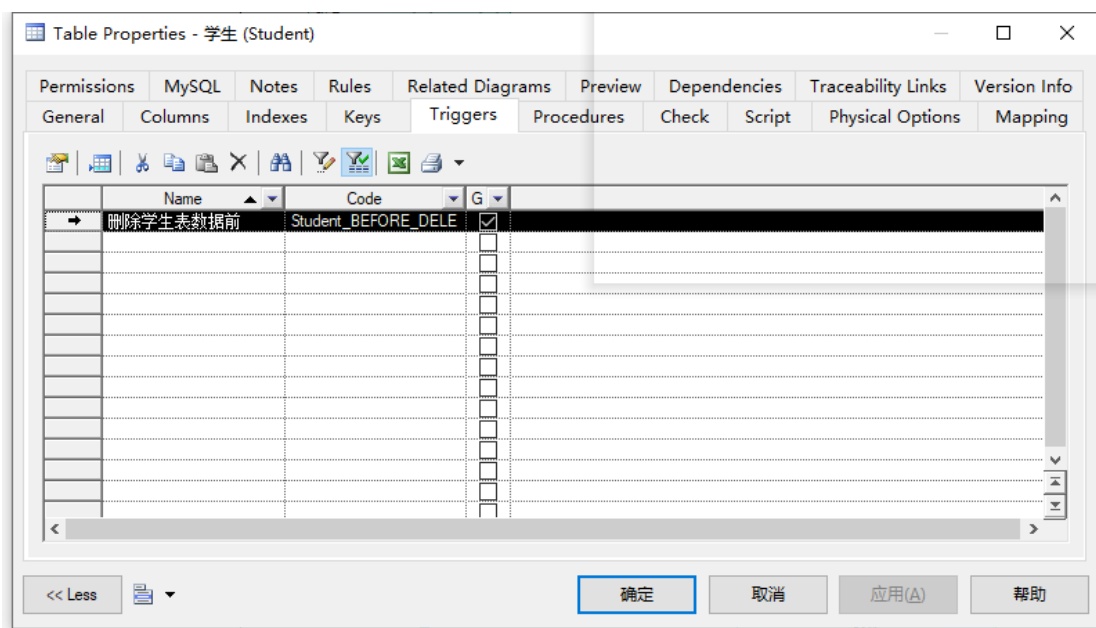
5、 勾选你想建索引的一列或者多列，然后点击 OK:

4.3.3 定义触发器

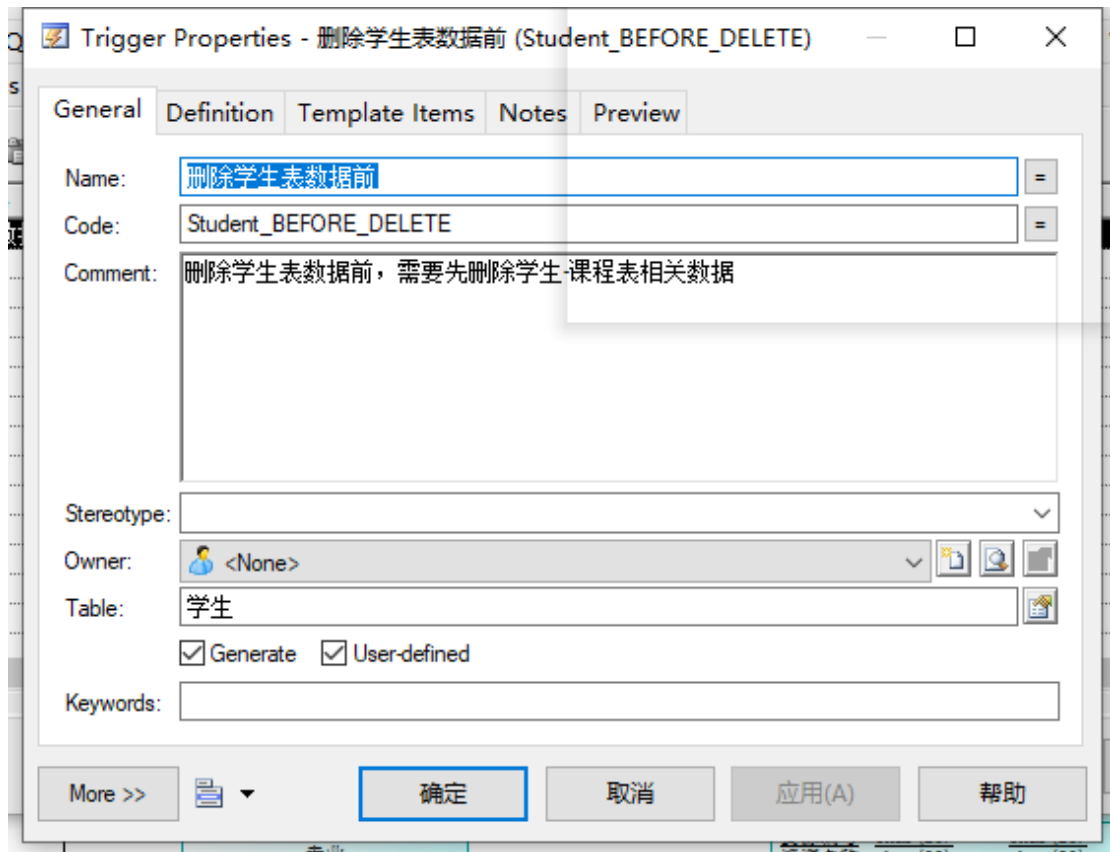
1、PDM 中，找到需要建触发器的表，打开以后找到 triggers 选项卡：



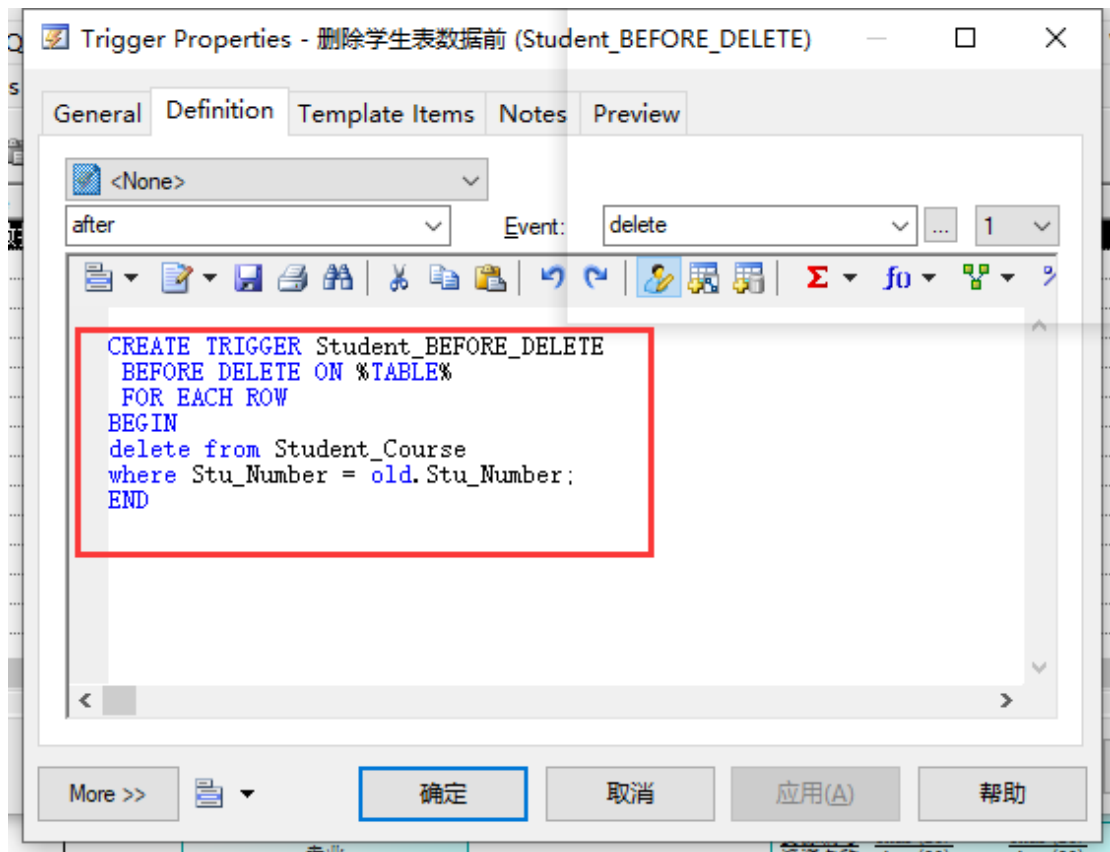
2、新增一行：



3、双击打开 trigger Properties 设置页面：



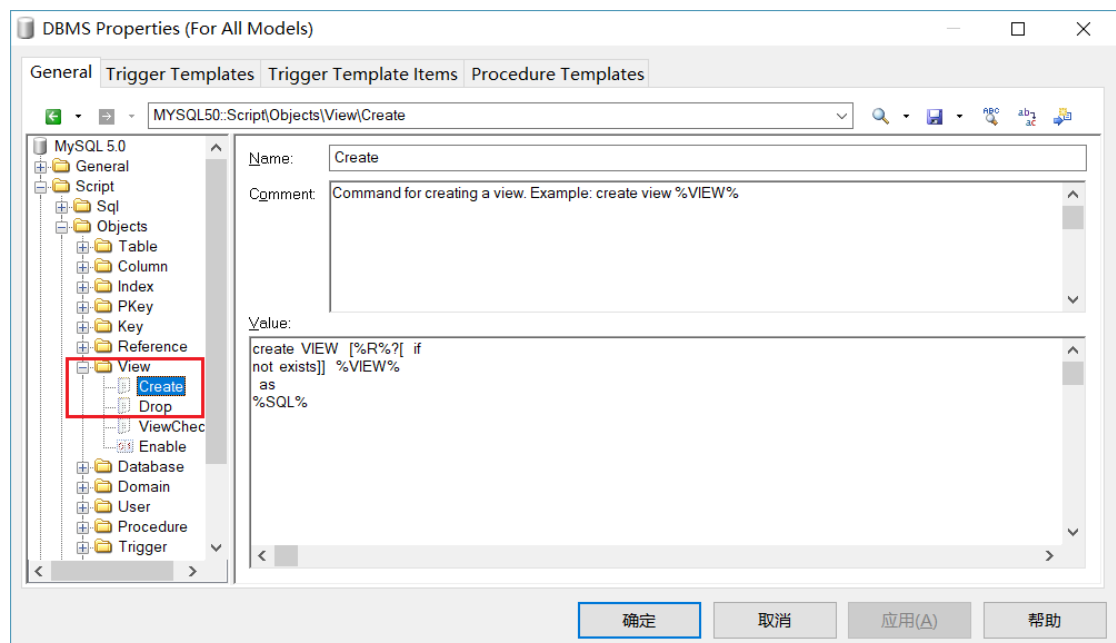
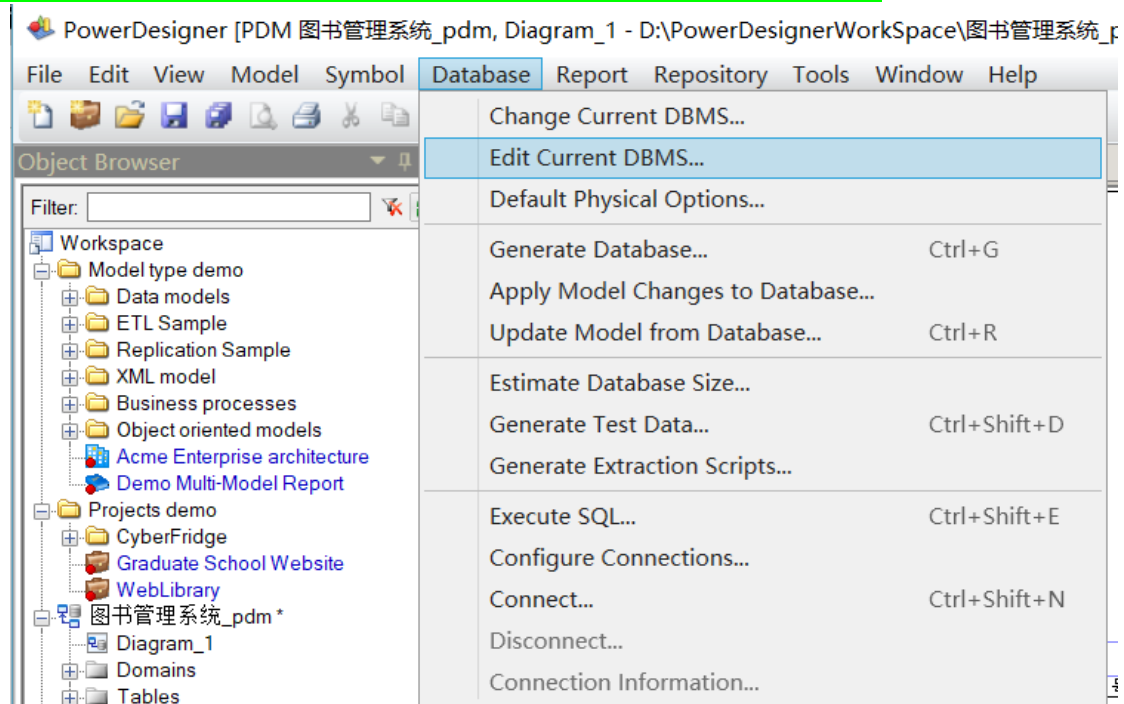
4、在 Definition 中填入 create trigger.....建索引语句:



为确保语句有效，可先在 MySQL 尝试执行。

4.4 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...



在左侧树状视图中选择

Script → Objects → View，选中 Create，在右侧的 Value 中输入：

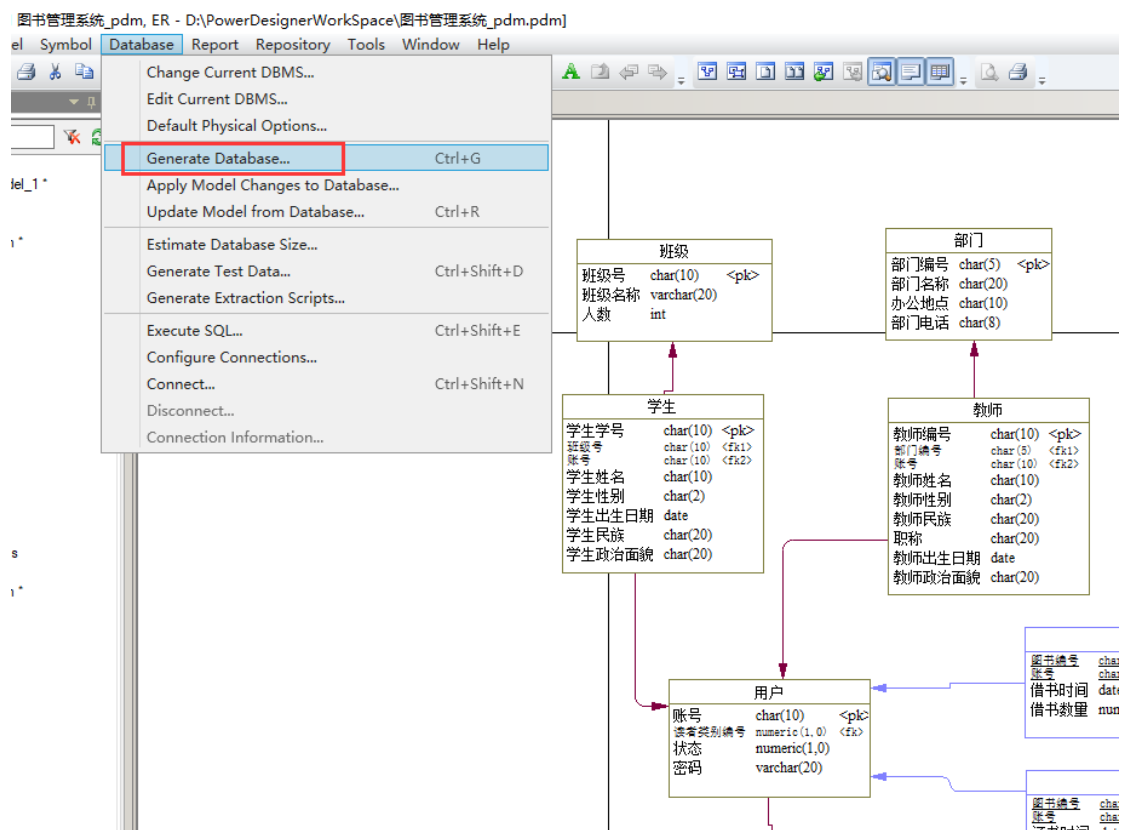
```
create VIEW [%R%?[% if  
not exists]] %VIEW%
```

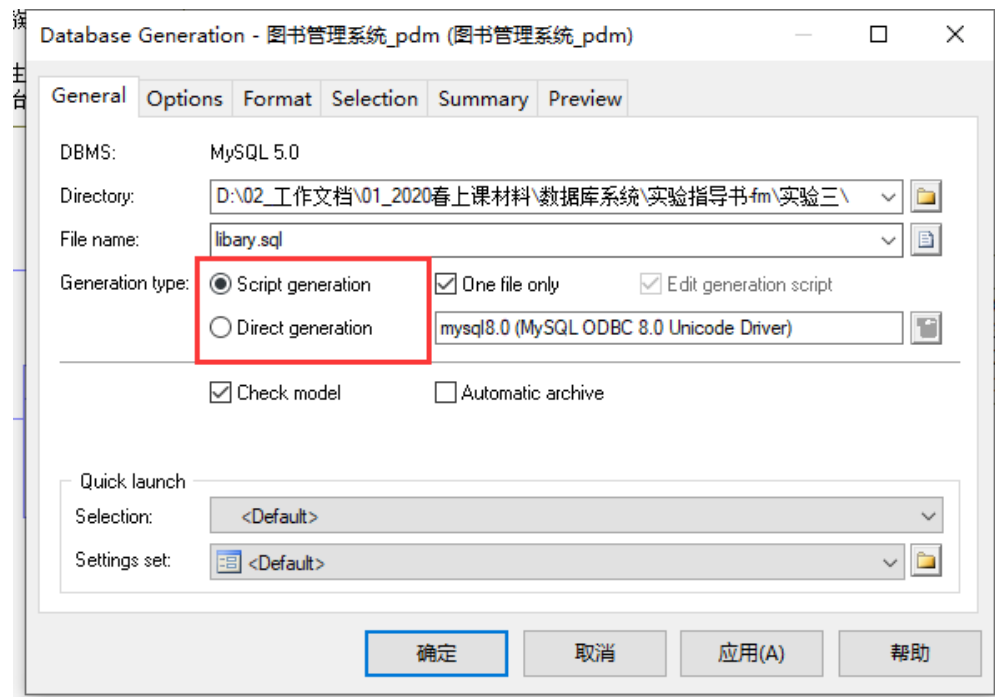
as
%SQL%

然后选中 Drop，在右侧的 Value 中拷贝下列文字
drop
table if exists %VIEW%

确定。

步骤 1: 在 DataBase 菜单中单击 Generate Database 命令，如下图所示。

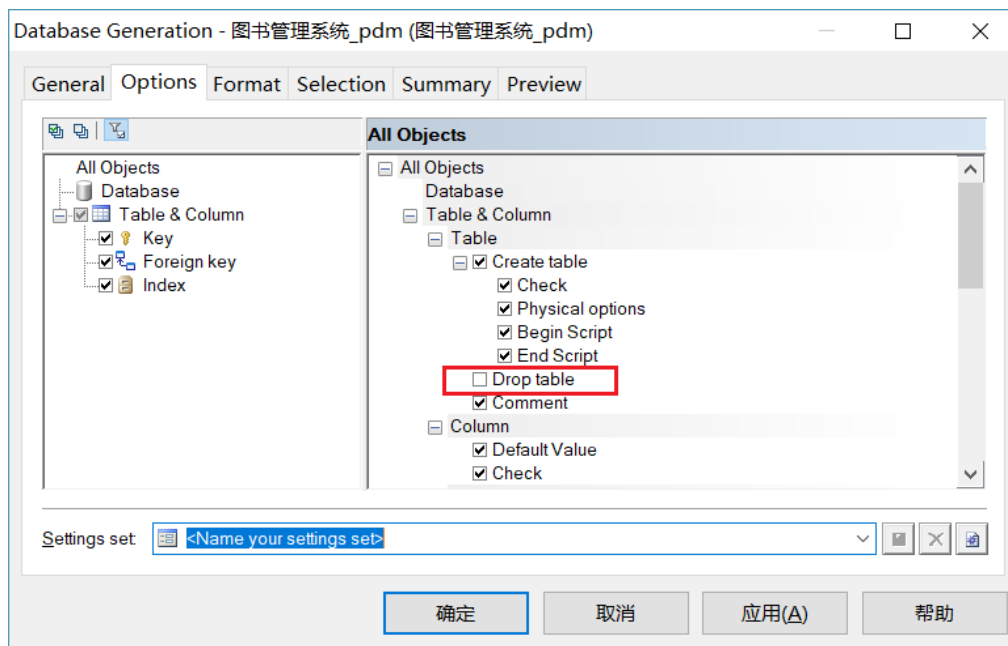




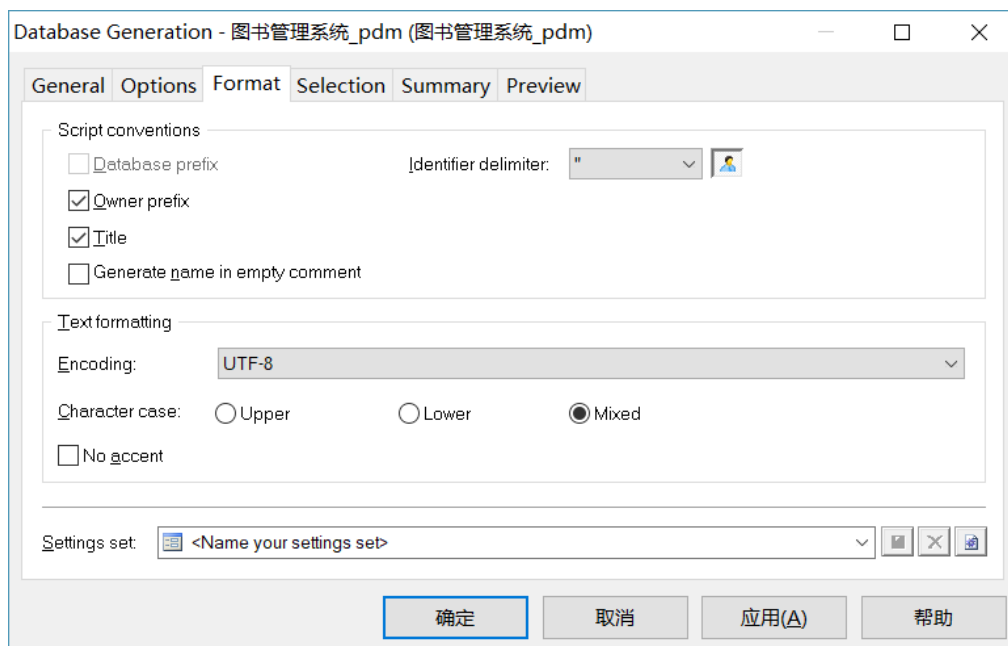
General 选项卡:

- 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:

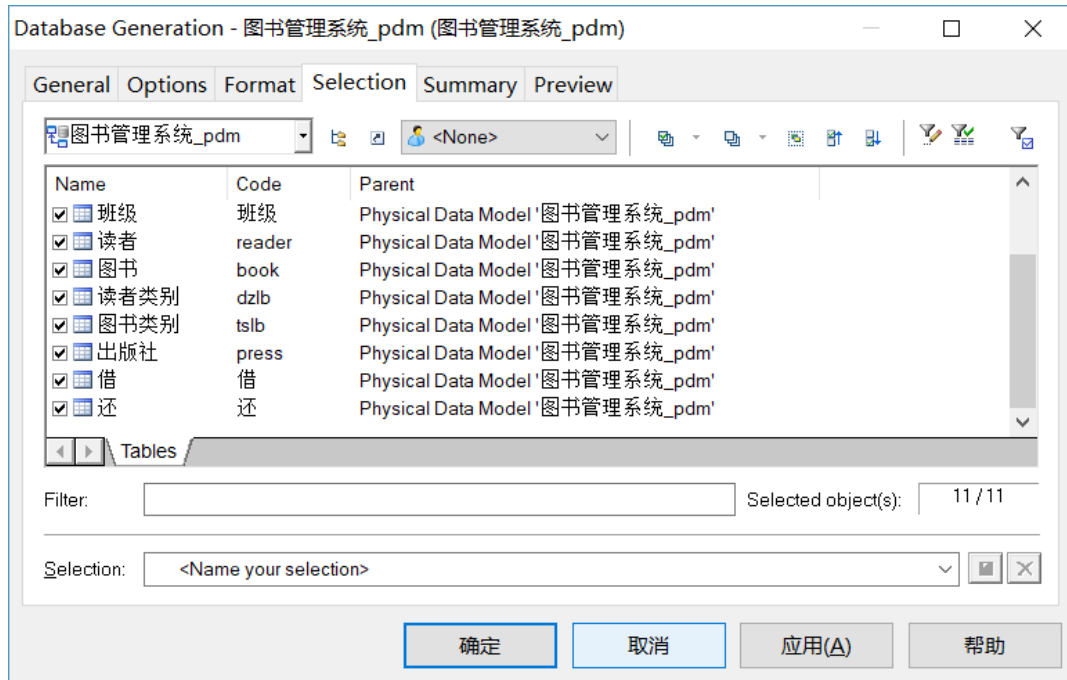


Format 选项卡:

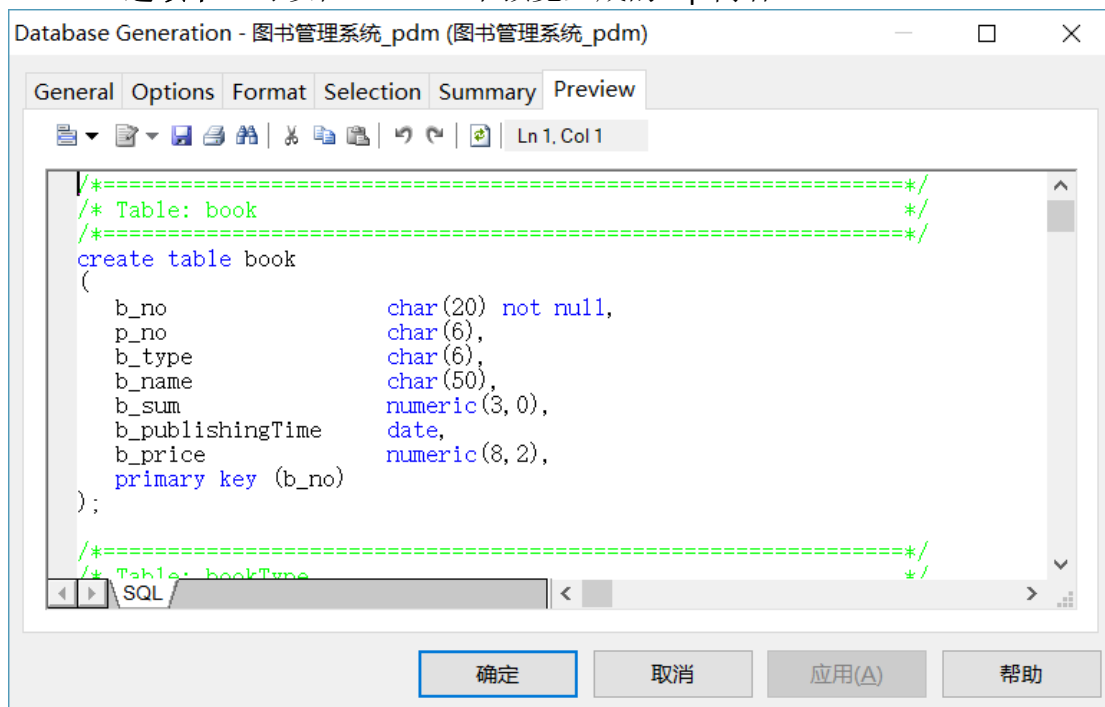


- Script Conventions
 - Database prefix: 代码中表和视图名称的前面带有数据库名的前缀。
 - Owner prefix: 代码中表和视图名称的前面带有所有者的名称。
 - Title: 代码中每节前面包括一个注释，例如：Database Name: ORCL。
 - Generate name in empty comment: 如果表、列、视图等模型对象的注释(Comment)为空，则生成的代码中将使用对象的名称作为注释。
- Encoding 改成 UTF-8，便于 MySQL Workbench 直接打开。

Selection 选项卡:

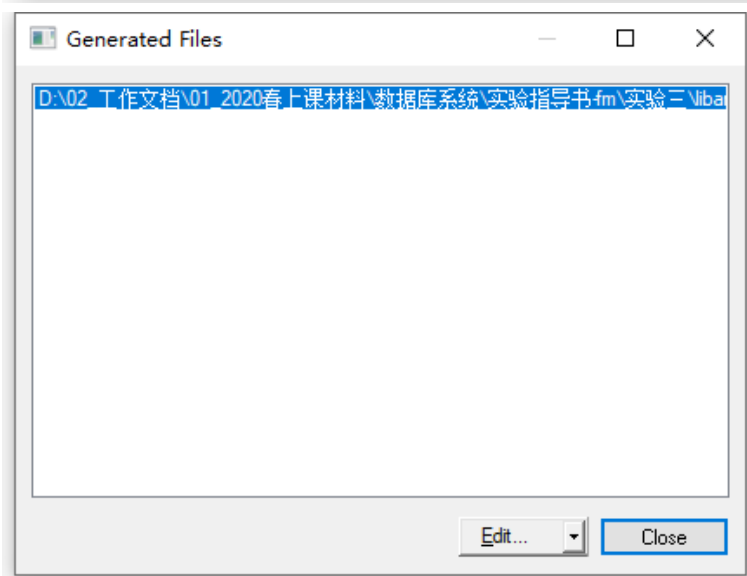
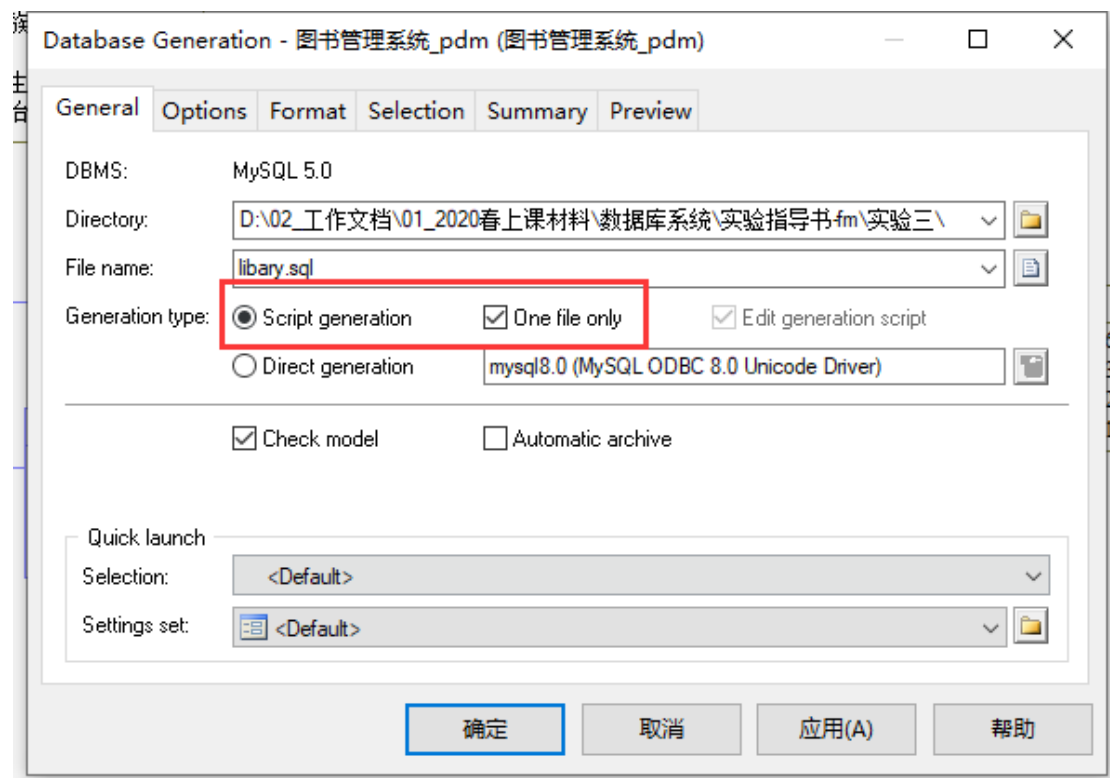


Preview 选项卡：可以在 Preview 中预览生成的 sql 内容

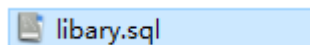


以下分别介绍生成 sql 文件在 **mysql workbench** 里执行、直接在 **Powerdesigner** 里连接数据库执行 sql 两种方式。

方式一，生成 sql 文件在 **mysql workbench** 里执行：

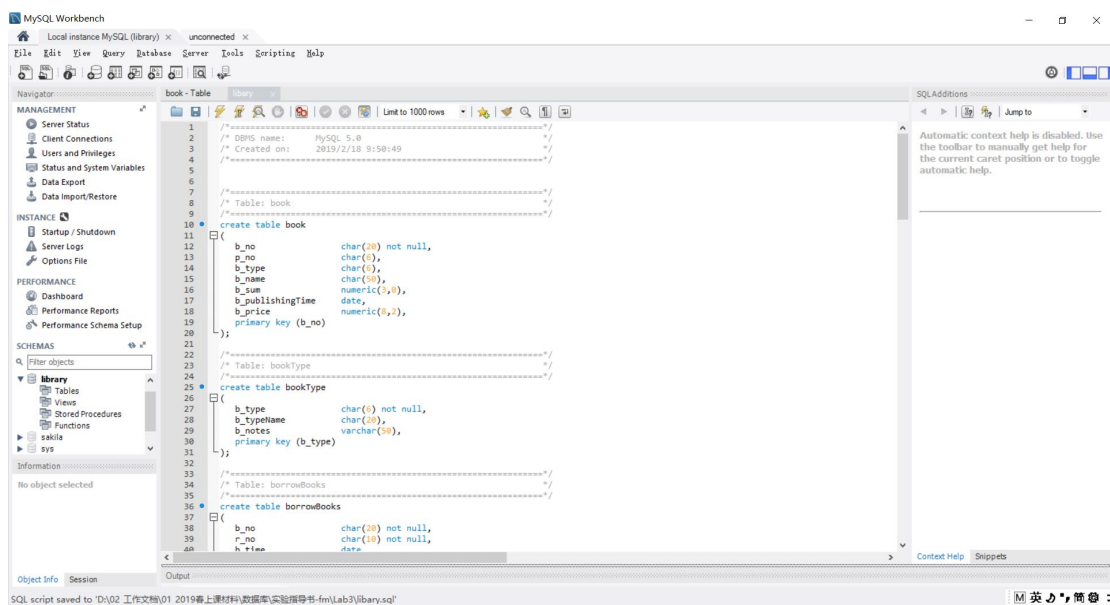
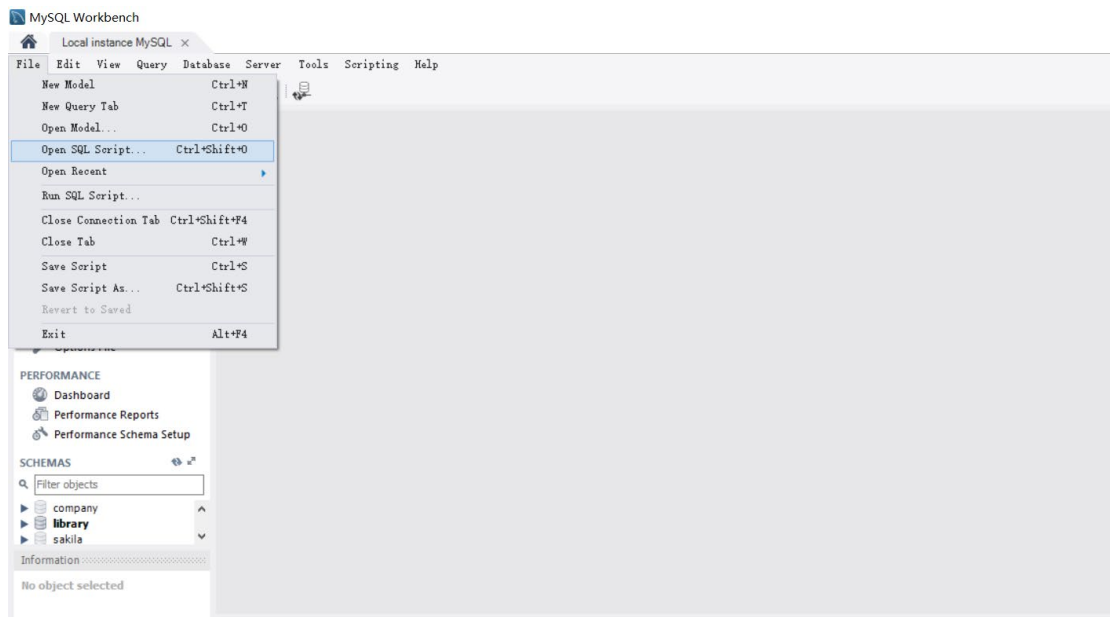


可以在指定目录下找到生成的 sql 文件。

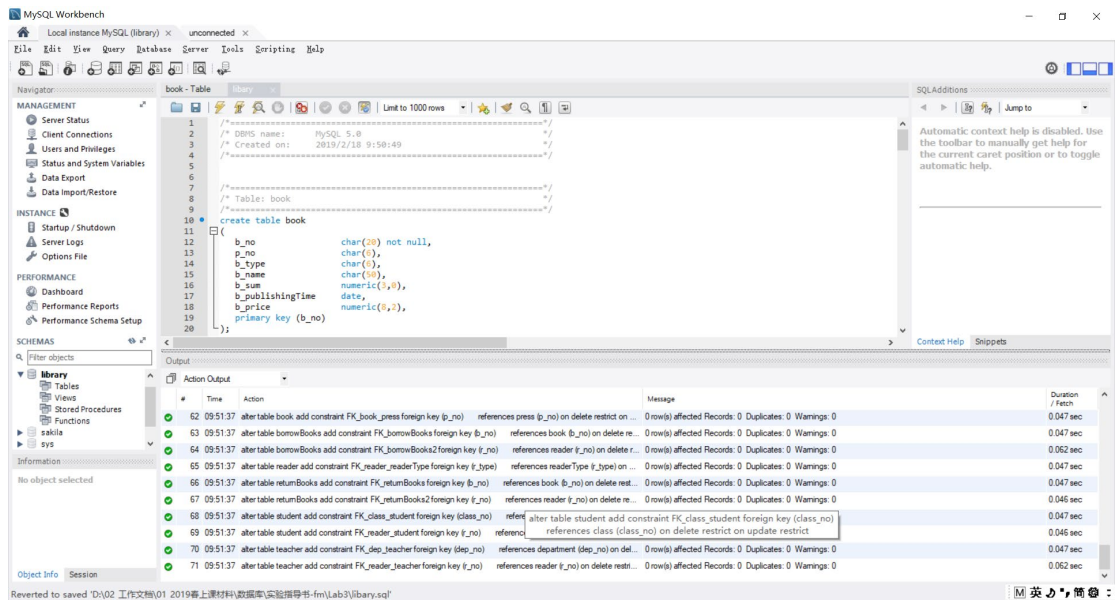


步骤 2：打开 **MySQL Workbench**，**CREATE** 一个数据库。

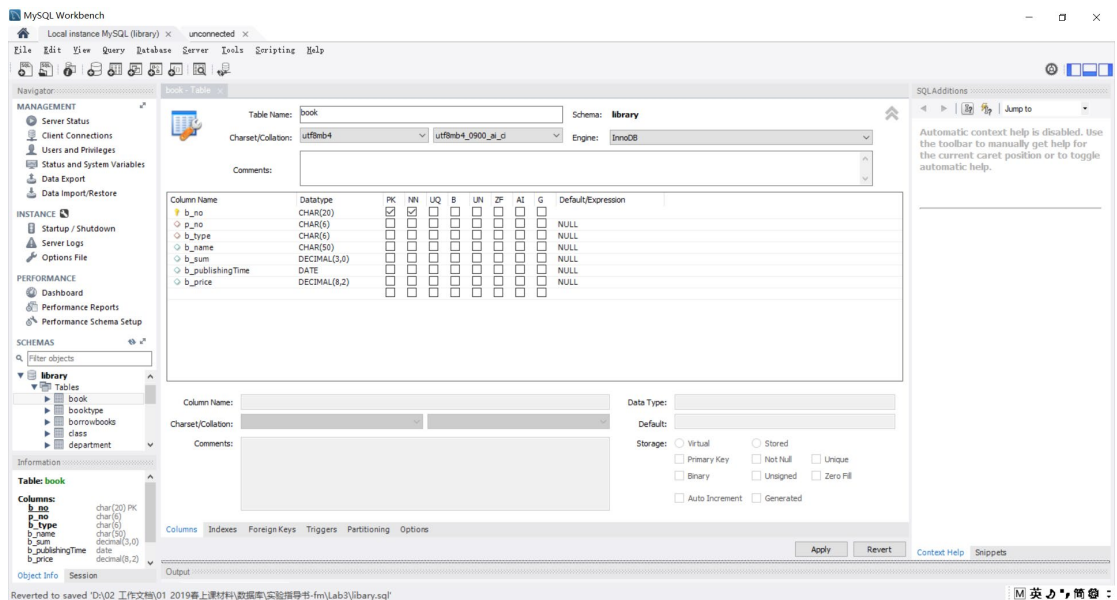
步骤 3：在 **MySQL Workbench** 中，打开 **File→Open SQL Script**



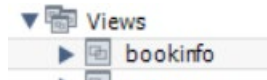
步骤 4: 在 MySQL Workbench 中，执行 SQL 脚本，在 Output 窗口看执行结果



至此，数据库 library 建好了。可以打开表检查是否正确：

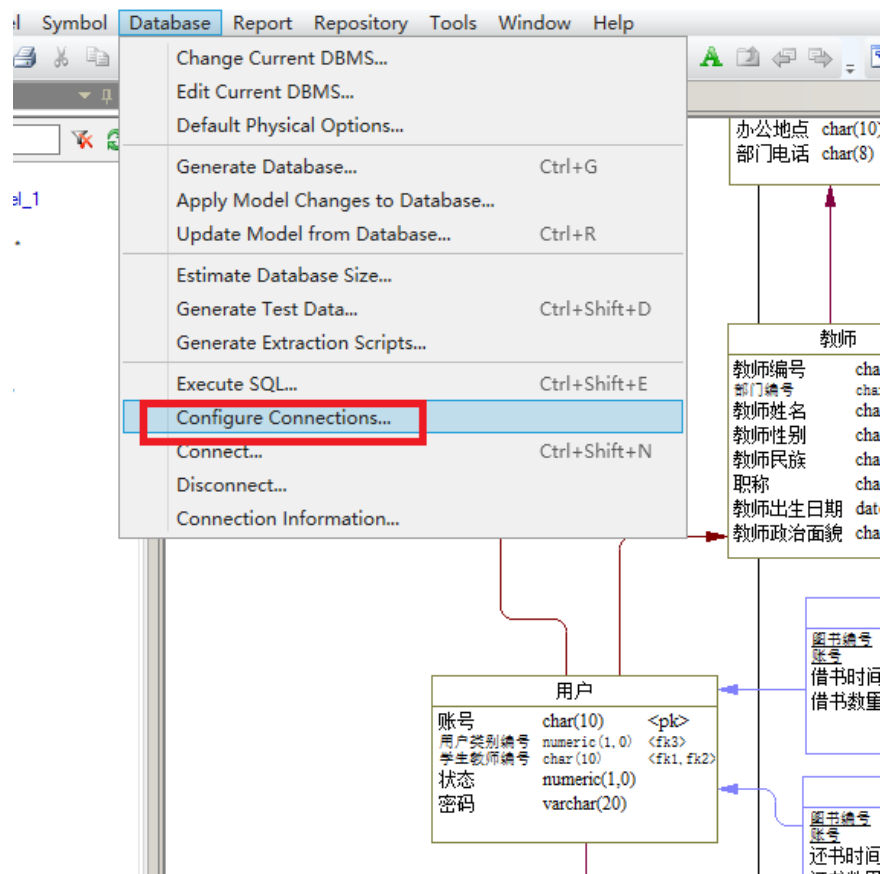


检查视图：

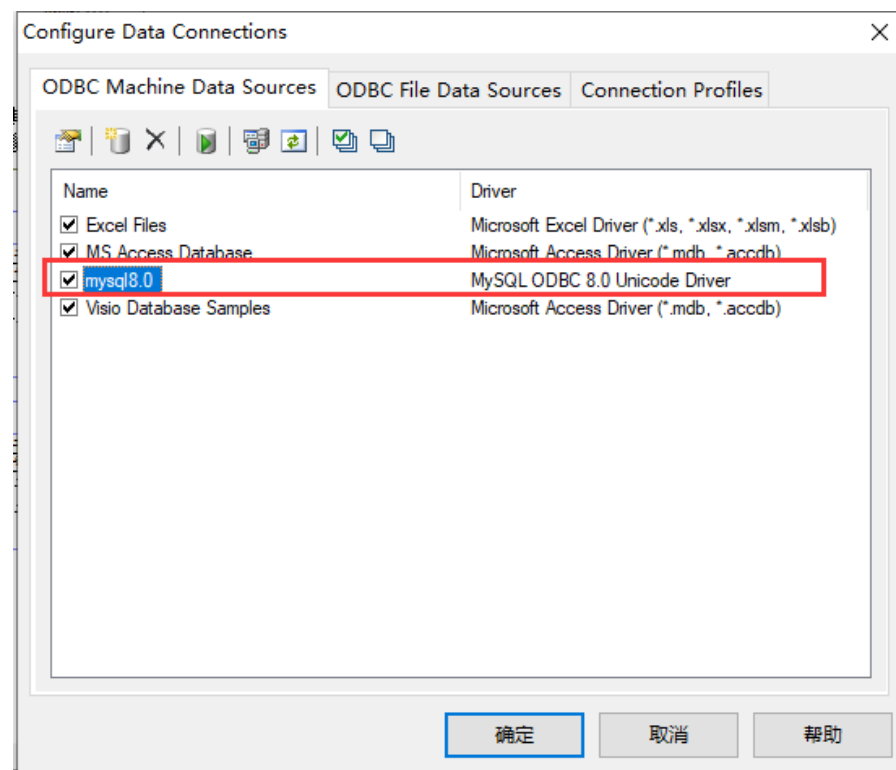


方法二，直接在 Powerdesigner 里连接数据库执行 sql 建表：

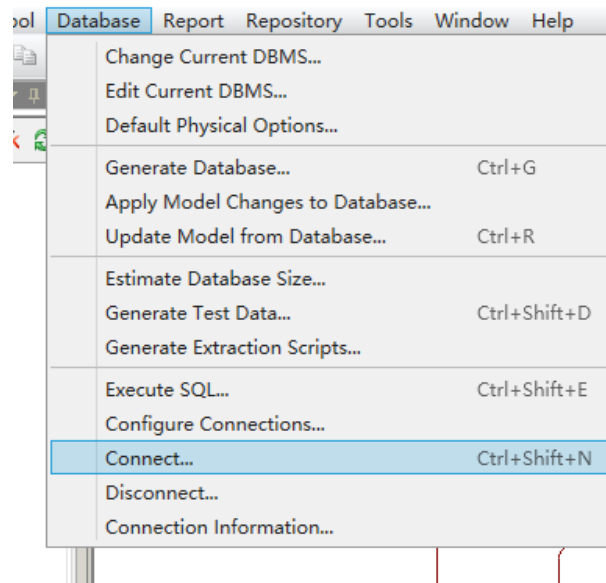
步骤 1：在 PowerDesigner 中，配置数据库连接：



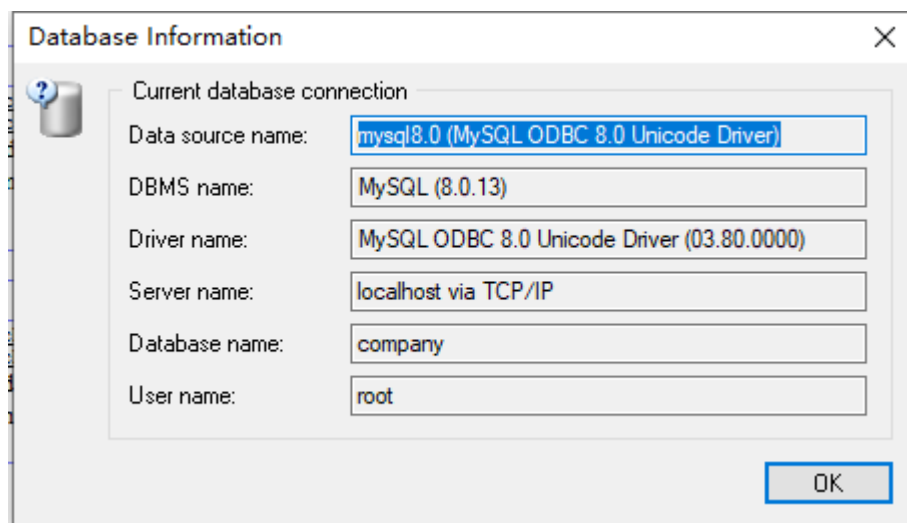
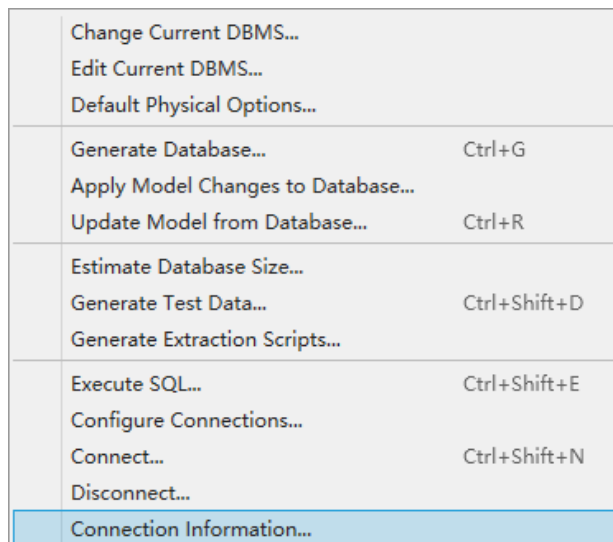
这里用 odbc 方式，先配置好了 odbc 数据源（步骤请自行百度）。



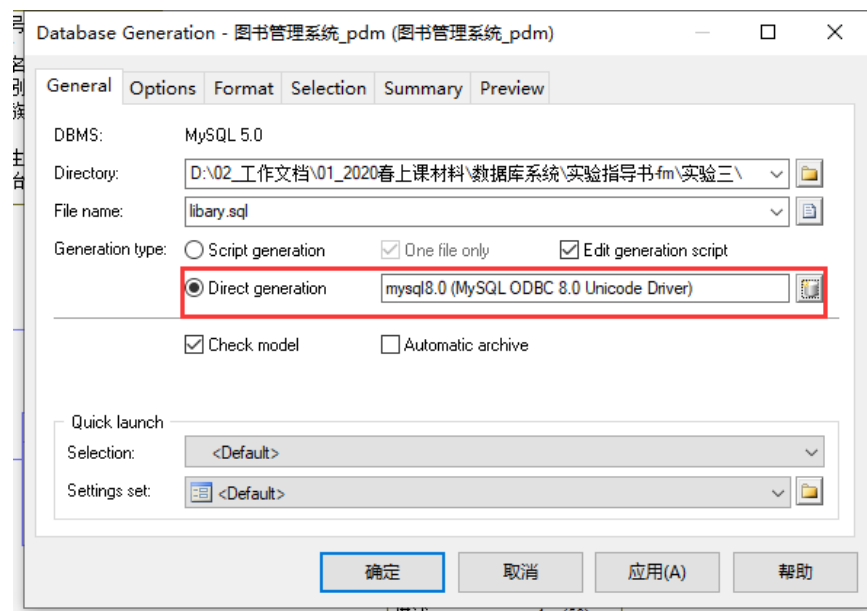
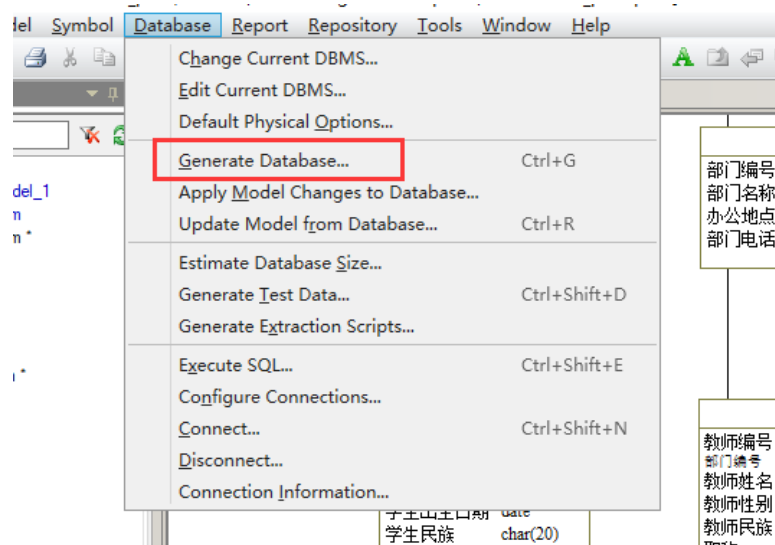
测试一下连接：



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：

```
/*=====*/
/* DBMS name:      MySQL 5.0 */
/* Created on:     2020/4/3 13:46:44 */
/*=====*/
create database library;
use library;

/*=====*/
/* Table: book */
/*=====*/
create table book
(
  b_no      char(20) not null,
  b_type    char(6),
  p_no      char(6) not null,
  b_name    char(50),
  b_sum     numeric(3,0),
  b_publishingTime date,
  b_price   numeric(8,2),
  primary key (b_no)
);

/*=====*/
/* Table: bookType */
/*=====*/
create table bookType
(
  b_type    char(6) not null,
  b_typeName char(20),
  b_notes   varchar(50),
  primary key (b_type)
);

/*=====*/
/* Table: borrowBooks */
/*=====*/
create table borrowBooks
```

运行完毕可以看到执行结果：

```
/*=====*/
/* DBMS name:      MySQL 5.0 */
/* Created on:     2020/4/3 13:46:44 */
/*=====*/
create database library;
use library;

/*=====*/
/* Table: book */
/*=====*/
create table book
(
  b_no      char(20) not null,
  b_type    char(6),
  p_no      char(6) not null,
  b_name    char(50),
  b_sum     numeric(3,0),
  b_publishingTime date,
  b_price   numeric(8,2),
  primary key (b_no)
);

/*=====*/
/* Table: bookType */
/*=====*/
create table bookType
(
  b_type    char(6) not null,
  b_typeName char(20),
  b_notes   varchar(50),
  primary key (b_type)
);

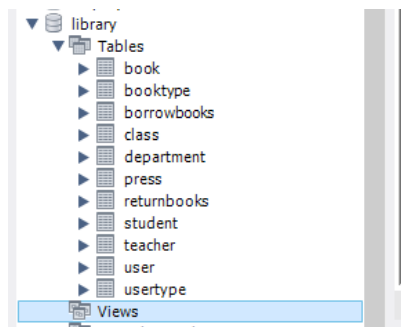
/*=====*/
/* Table: borrowBooks */
/*=====*/
create table borrowBooks
```

SQL statement successfully executed.

Executing SQL statement 6:
create table class
(
 c_no char(10) not null,
 c_name varchar(20),
 c_count int,
 primary key (c_no)
);
SQL statement successfully executed

Executing SQL statement 7:
create table department
(
 dep_no char(5) not null,
 dep_name char(20),
 dep_address char(10),
 dep_telephone char(8),
 primary key (dep_no)
);

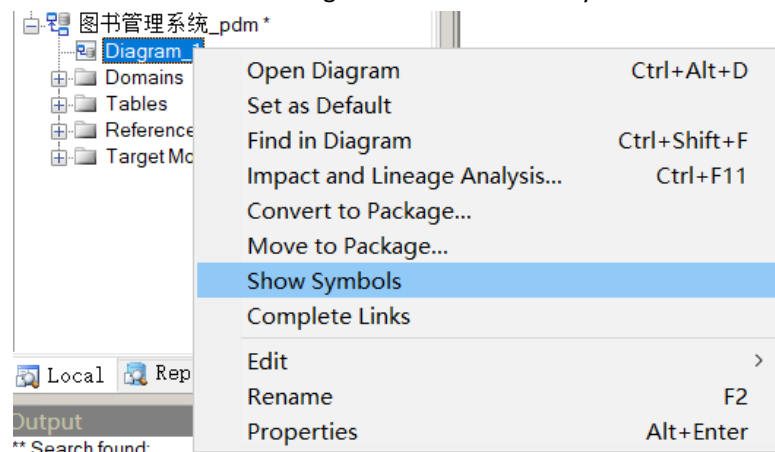
步骤 3：在 MySQL Workbench 中刷新可以看到新建的库、表信息：



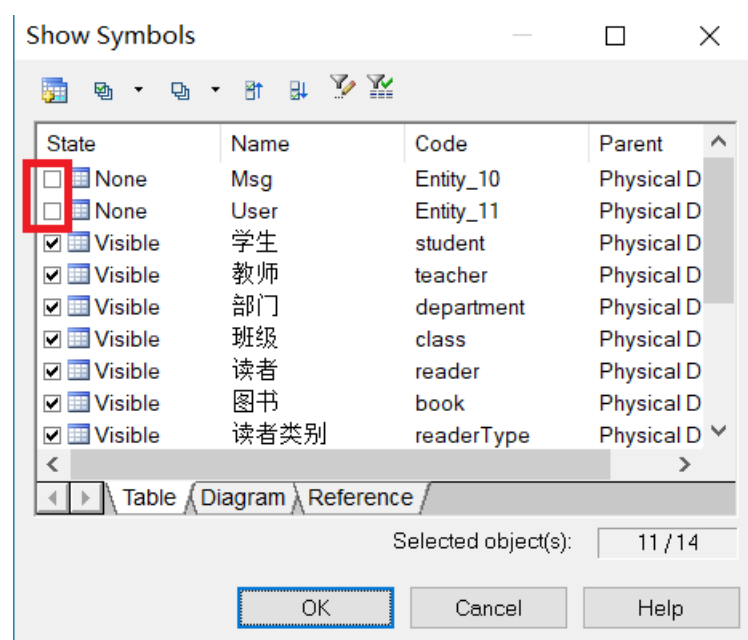
5 常见问题

5.1 PDM 存在表但不显示

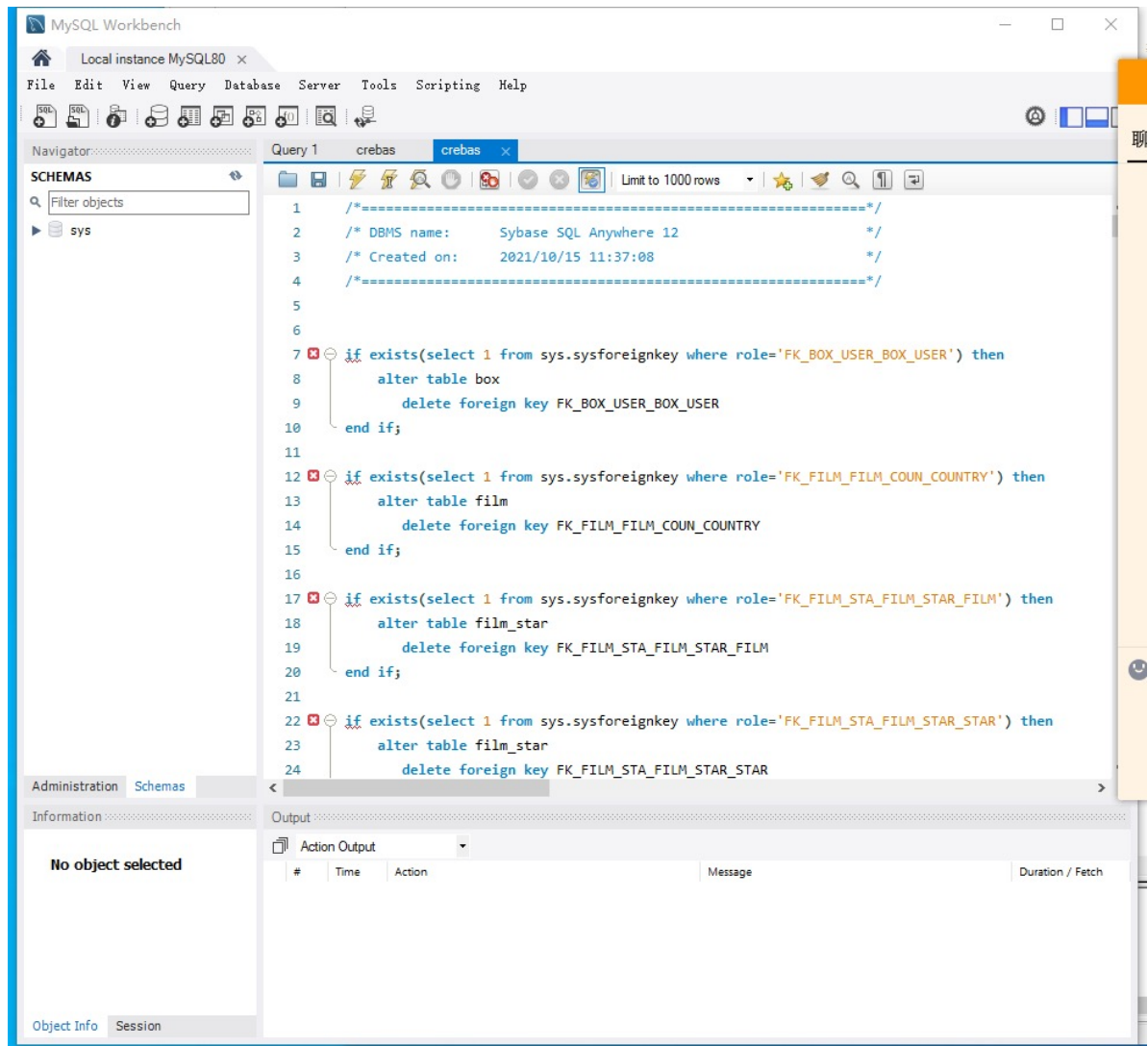
解决方法：在 PDM 的 Diagram 上右键→Show Symbols



在要显示的表前面打勾，OK

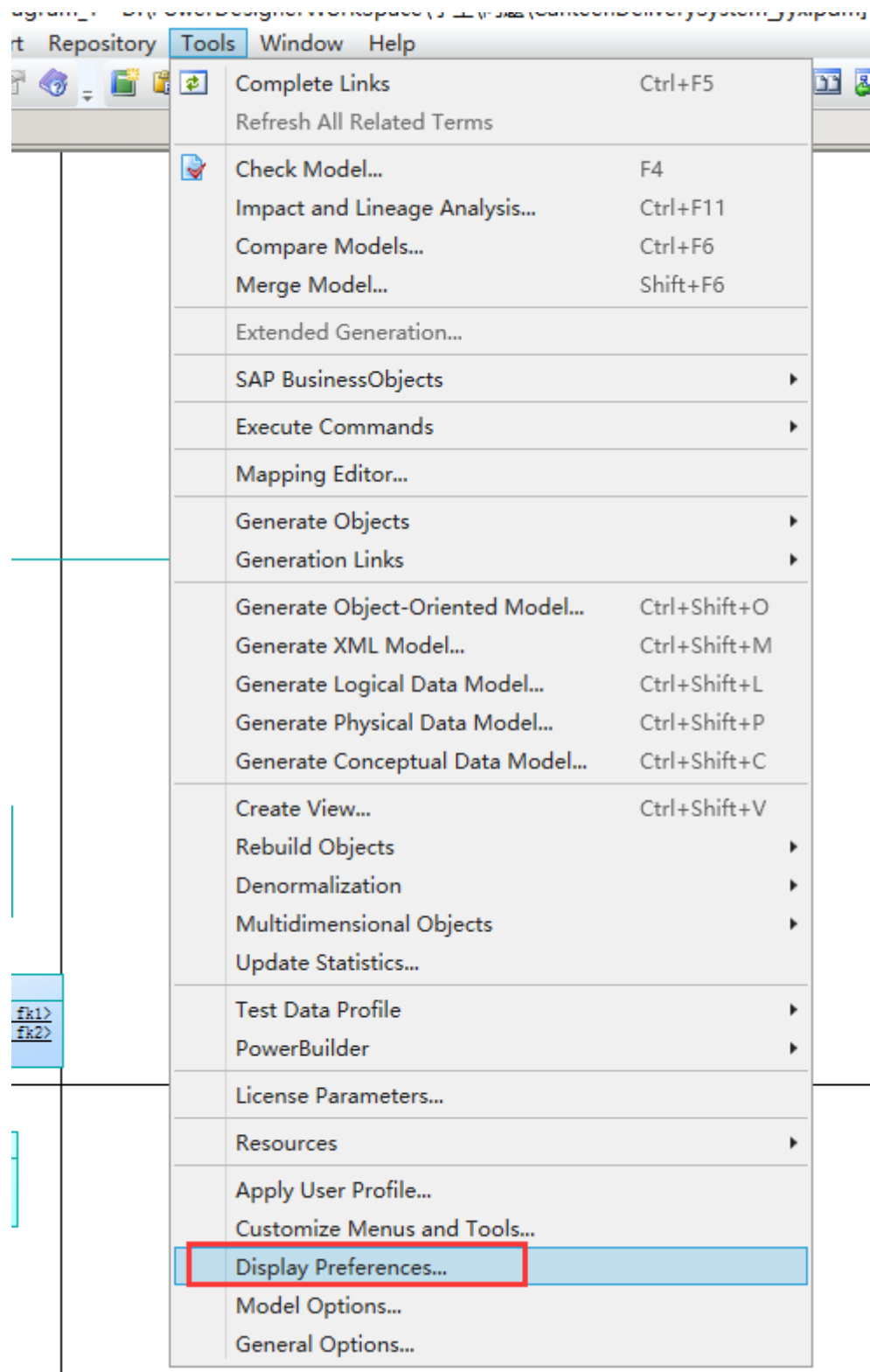


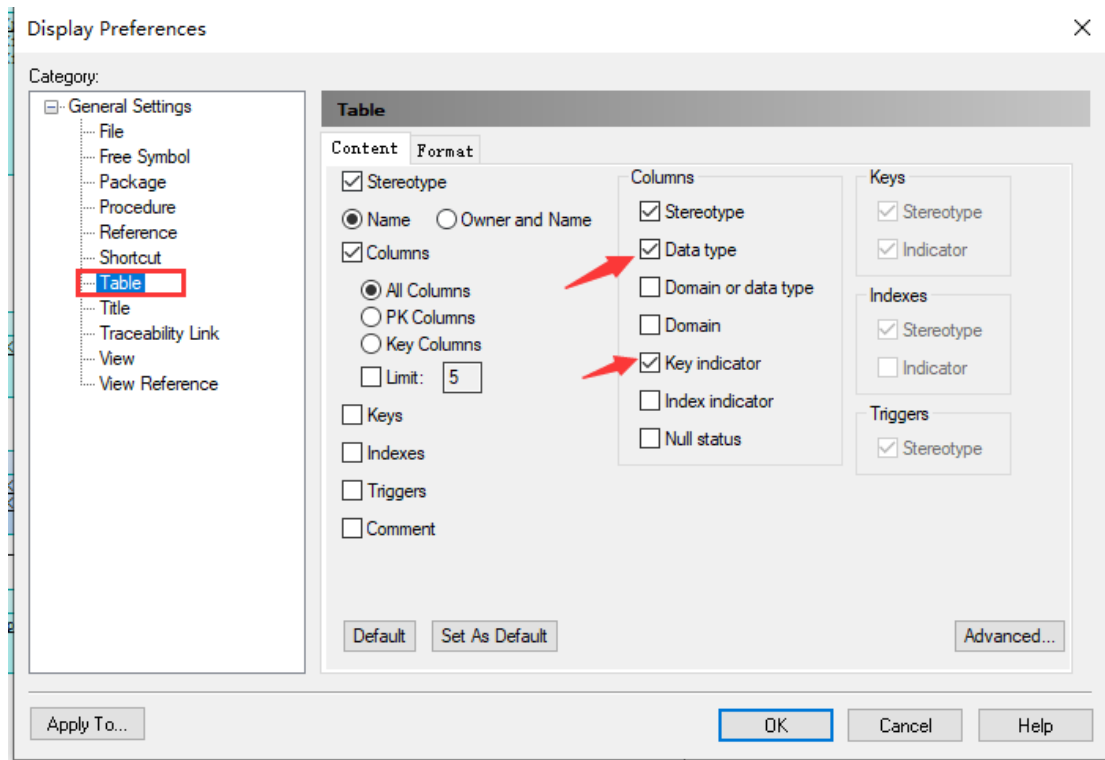
5.2 生成的 SQL 脚本在 MySQL 中执行报错



大家注意看上面的图第 2 行是 DBMS name: Sybase SQL Anywhere 12，表示没有选择 DBMS 是 MySQL，解决方法详见 4.3 节。

5.3 PDM 中 table 只显示字段的 name, 想要显示 type 和其他信息





5.4 生成的 SQL 执行报错：有重复的外键约束

问题可能出在：ER 图时 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。
解决办法：把其中一边的线条改成不重复的名字即可。

6 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）