

# 软工第三次上机

PowerDesigner 数据模型构建

# 实验目标

- ▶ 理解数据模型的概念，包括概念数据模型（CDM）、逻辑数据模型（LDM）、物理数据模型（PDM）。并了解三者的关系及区别。
- ▶ 使用 PowerDesigner 完成三种数据模型画图设计以及相互转换，并通过物理数据模型导出SQL脚本。

# 数据模型

- ▶ 数据模型是现实世界中数据特征的抽象。数据模型应该满足三个方面的要求：
  - 1) 能够比较真实地模拟现实世界
  - 2) 容易为人所理解
  - 3) 便于计算机实现
- ▶ 作用：方便数据库的构建

# 概念数据模型 Conceptual Data Model

- ▶ 概念数据模型也称信息模型，它以实体 - 联系 (Entity-RelationShip,简称E-R)理论为基础，并对这一理论进行了扩充。它从用户的观点出发对信息进行建模，主要用于数据库的概念级设计。在CDM中要体现实体、实体所包含的属性以及实体之间的关系。

# 逻辑数据模型 Logical Data Model

- ▶ 逻辑模型介于概念模型和物理模型之间。逻辑模型中一方面显示了实体、实体的属性和实体之间的关系，另一方面又将继承、实体关系中的引用等在实体的属性中进行展示，具有物理模型方面的特征。在概念模型中的多对多关系，在逻辑模型中将会以增加中间实体的一对多关系的方式来实现。
- ▶ 逻辑模型是CDM到PDM转化的中间桥梁。

# 物理数据模型 Physical Data Model

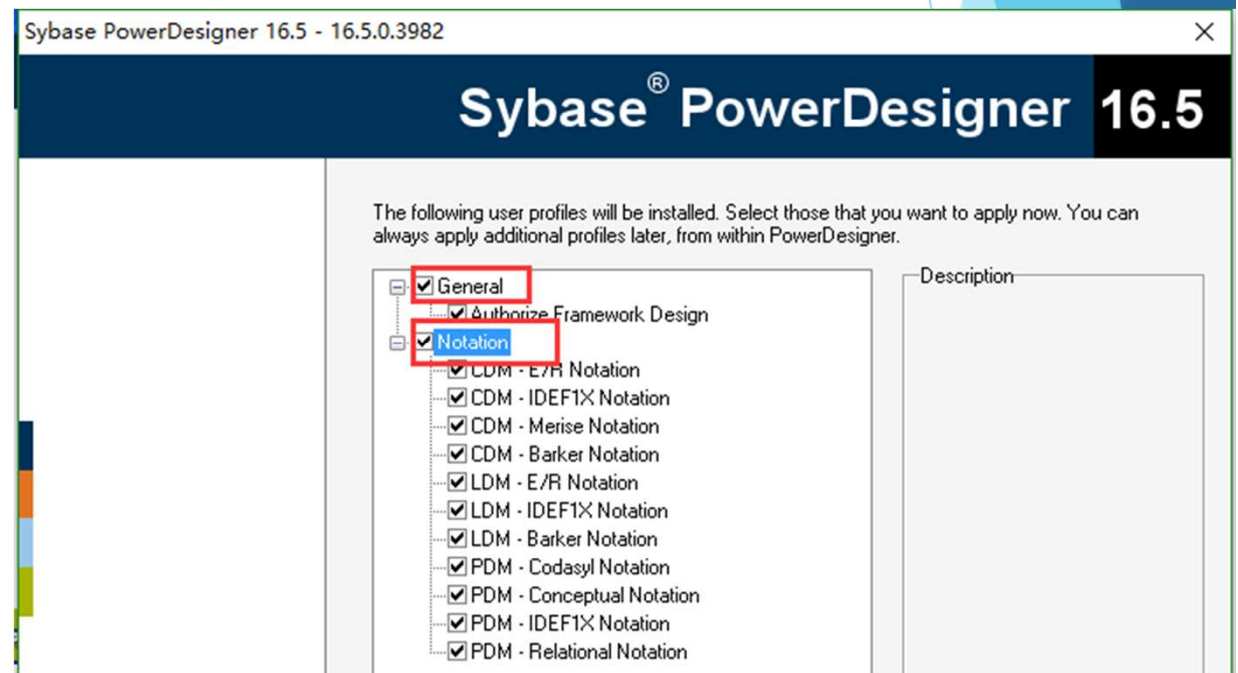
- ▶ 物理数据模型PDM用于描述数据在存储介质上的组织结构，与具体的DBMS（如SQL sever, MySQL）有关。用户需要在逻辑数据模型的基础上综合考虑具体的技术实现因素，如数据库存储过程、触发器、索引等。
- ▶ 使用PowerDesigner可以通过PDM直接导出多种数据库的对应构建脚本。

# 实验步骤

- ▶ 0. 安装 PowerDesigner
- ▶ 1. 新建模型
- ▶ 2. 构建CDM模型图
- ▶ 3. 使用PowerDesigner完成CDM到LDM的转换。
- ▶ 4. 使用PowerDesigner完成CDM到PDM的转换。
- ▶ 5. 使用PowerDesigner从PDM模型直接导出MySQL脚本。

# 安装软件

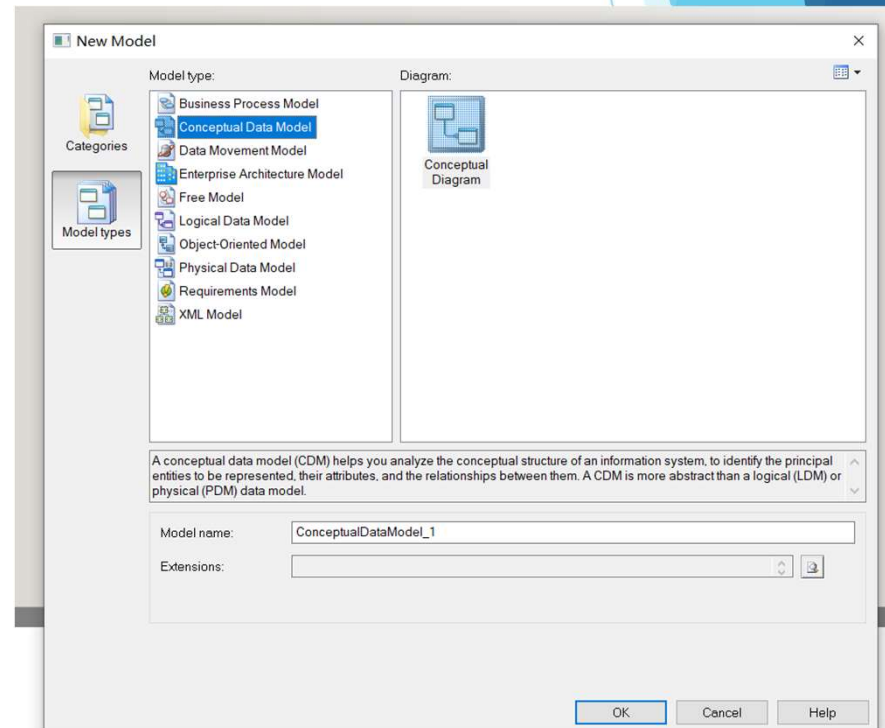
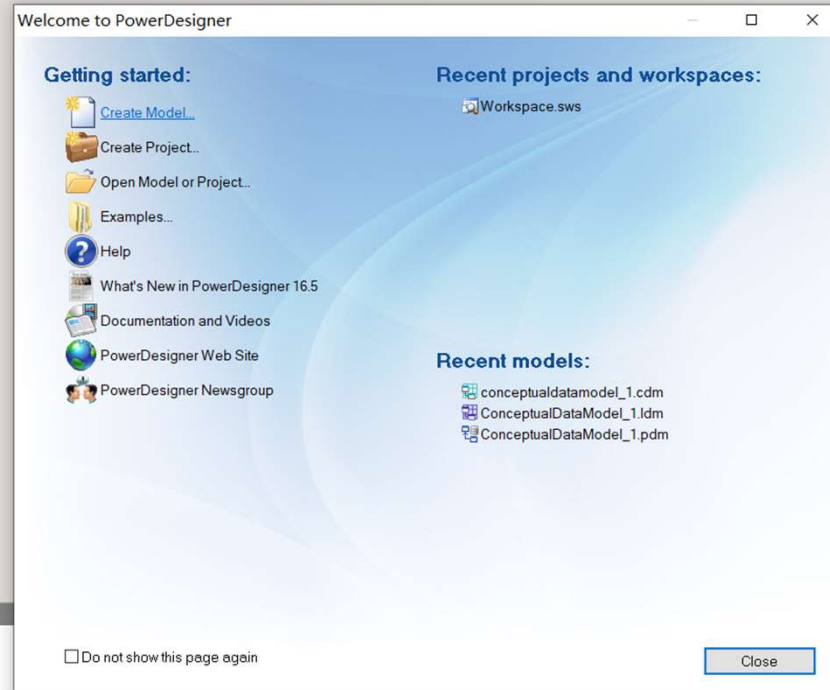
- ▶ 课程群里提供了软件的下载链接，自行从网上下载也可以
- ▶ 安装基本都用默认设置一路 next 就行
- ▶ 这个我也不懂是干啥的，反正都选上了↓





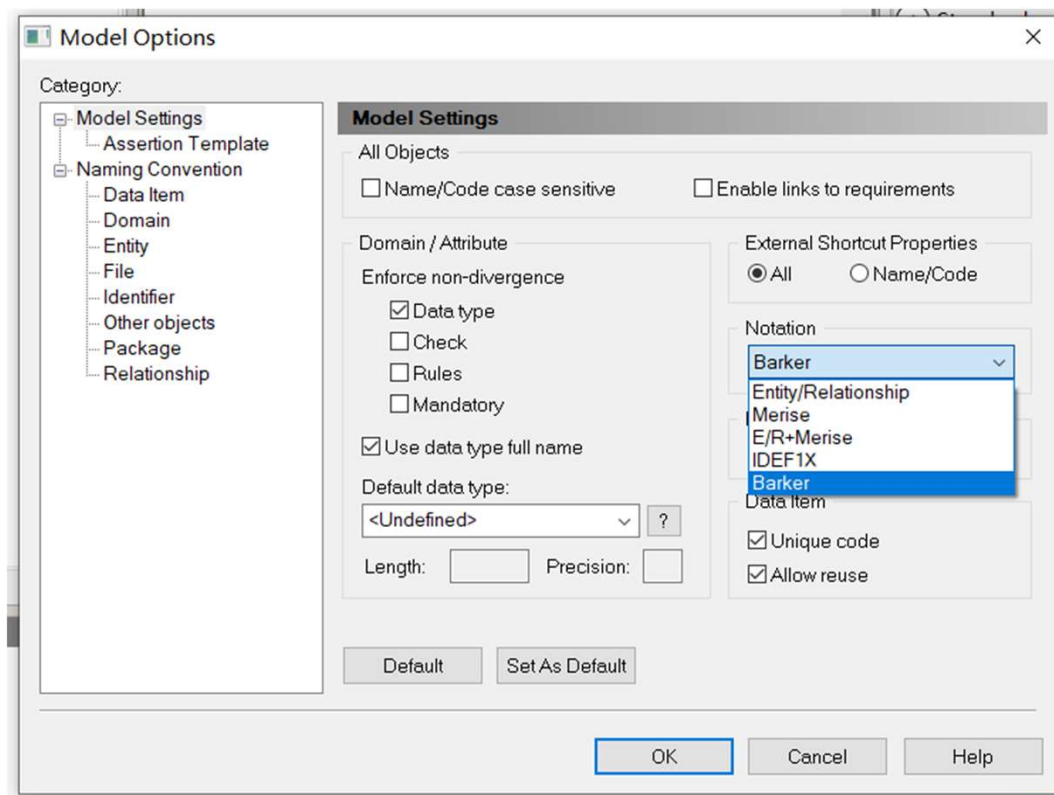
# 新建项目

- ▶ 在开始页面选择create model
- ▶ 选择model types – conceptual data model
- ▶ 编辑model name, 点OK完成创建



# 预先设置

- ▶ CDM模型默认采用的是Barker' s Notation，我们本次上机采用ER模型，若不做修改将无法使用继承关系。
- ▶ Tools-model options-Notation -选择Entity/Relationship

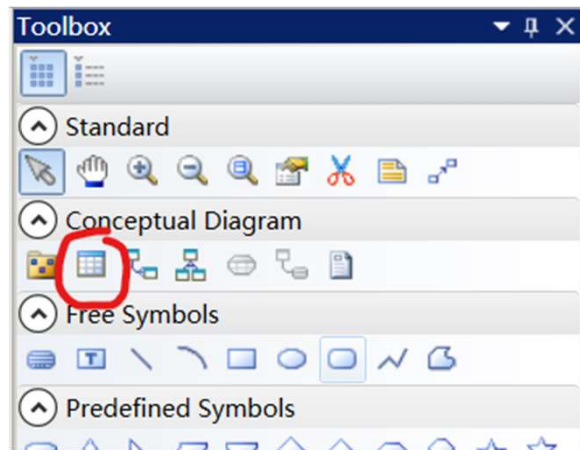


# 预先设置

- ▶ 注：Barker' s notation 是另外一种绘图标准，因由Barker提出而得名。
- ▶ Data model实际上有多种notation，感兴趣的同学可以了解一下：
- ▶ [https://en.wikipedia.org/wiki/Data\\_modeling](https://en.wikipedia.org/wiki/Data_modeling)

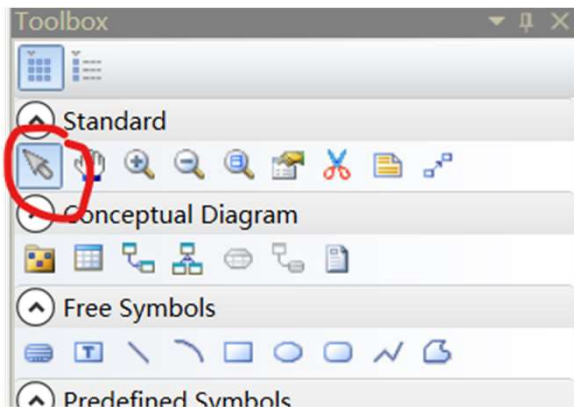
# 新建实体 (entity)

- ▶ 在工具栏 (tool box) 中选择实体，在绘图空白区域点击即可创建一个实体。



# 编辑实体

- ▶ 先将工具换回指针后双击刚刚建立的实体，在弹出的对话框中进行编辑。



The image shows the 'Entity Properties - Entity\_1 (Entity\_1)' dialog box. It has five tabs: General, Attributes, Identifiers, Notes, and Rules. The General tab is active. The dialog contains the following fields and controls:

- Name: 注册用户
- Code: 注册用户
- Comment: (Empty text area)
- Stereotype: (Empty dropdown menu)
- Number: (Empty text field) ☒ Generate
- Parent entity: <None>
- Keywords: (Empty text field)

At the bottom, there are four buttons: More >>, 确定 (Determine), 取消 (Cancel), 应用(A) (Apply), and 帮助 (Help).

# 添加属性 (attributes)

- ▶ 在刚才的对话框中选择attributes页即可添加属性。
- ▶ 注意右边三列字母的含义：
  - M-Mandatory, 选中表示该项不可为空。
  - P –Primary, 选中表示为主属性。
  - D –Display, 选中表示该属性在图中展示出来, 不选即隐藏。
- ▶ 一定要选好Data Type。



# 添加标识符 (identifier)

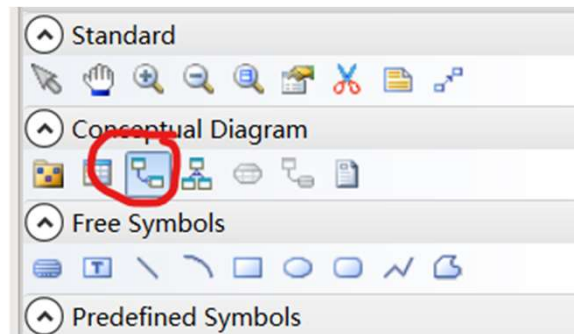
- ▶ 在刚才的对话框中选择identifiers页即可添加属性。
- ▶ 标识符分为主标识符和次标识符，类似数据库中的主键与候选键。
- ▶ 选中后面P选项的为主标识符。特定情况下可以没有标识符（如该实体有继承关系，由父类属性唯一标识）。
- ▶ 标识符没有对应属性的话导出逻辑模型会报错





# 添加关系 (relationship)

- ▶ 在toolbox中选择关系组件
- ▶ 点按实体拖至另一实体，在两个实体间建立关系。





# 编辑关系

- ▶ 双击建立的关系连线编辑关系名等信息。
- ▶ 在cardinalities页中编辑关系的多重性。

Relationship Properties - 作为买方 (作为买方)

Entity 1 Entity 2

注册用户 交易

General Cardinalities Notes Rules

Name: 作为买方

Code: 作为买方

Comment:

Stereotype:

Entity 1: 注册用户

Entity 2: 交易

☒ Generate

Keywords:

More >> 确定 取消 应用(A) 帮助

Relationship Properties - 作为买方 (作为买方)

Entity 1 Entity 2

注册用户 交易

General Cardinalities Notes Rules

Each 注册用户 may have one or more 交易.  
Each 交易 must have one and only one 注册用户.

Cardinalities

☐ One - one ☒ One - many ☐ Many - one ☐ Many - many

Dominant role: <None>

注册用户 to 交易

Role name:

☐ Dependent ☐ Mandatory Cardinality: 0..n

交易 to 注册用户

Role name:

☐ Dependent ☒ Mandatory Cardinality: 1..1

More >> 确定 取消 应用(A) 帮助

# 编辑关系-关系的多重性

- ▶ 与类图每一端的定义类似，如图也有用户到交易以及交易到用户的双向定义。
- ▶ Mandatory选中后表示强制关系，如一个交易必有一个注册用户。对应的一端会出现圆圈标记。

Relationship Properties - 作为买方 (作为买方)

Entity 1 Entity 2

注册用户 交易

General Cardinalities Notes Rules

Each 注册用户 may have one or more 交易.  
Each 交易 must have one and only one 注册用户.

Cardinalities

☐ One - one ☒ One - many ☐ Many - one ☐ Many - many

Dominant role: <None>

注册用户 to 交易

Role name:

☐ Dependent ☐ Mandatory Cardinality: 0..n

交易 to 注册用户

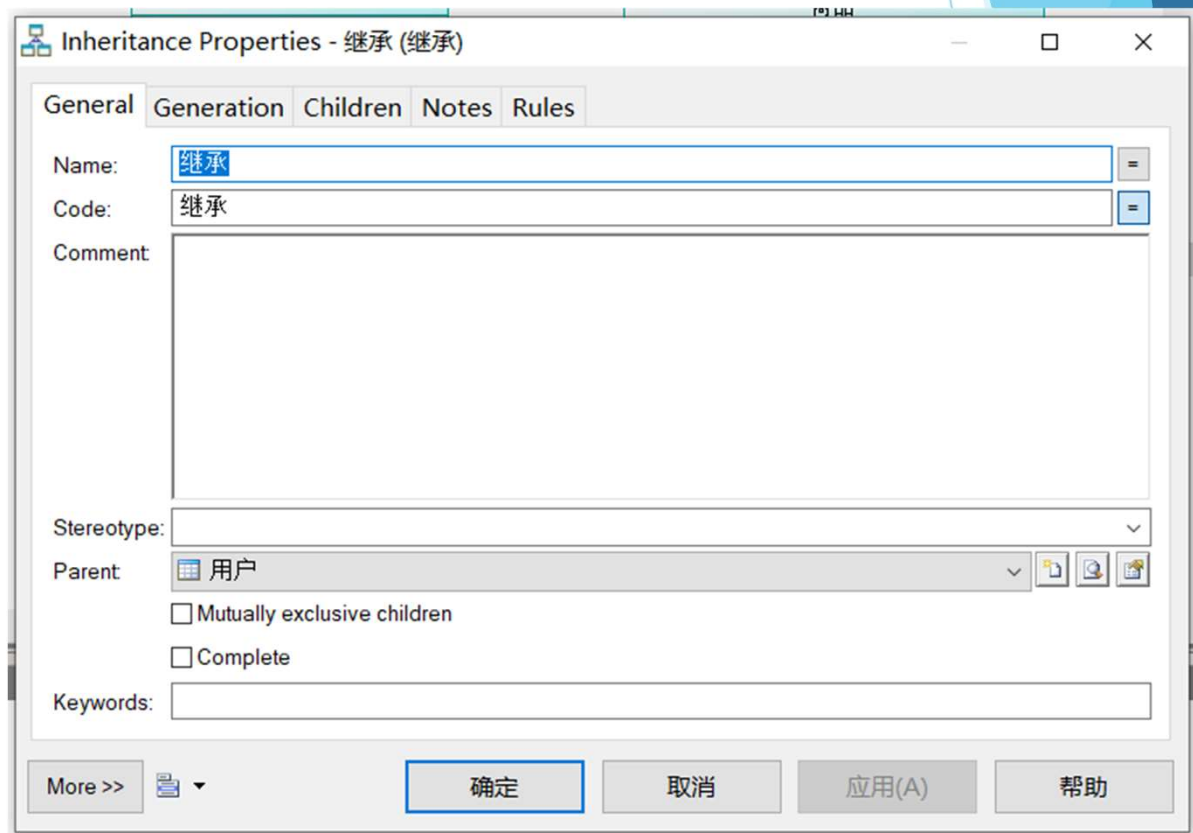
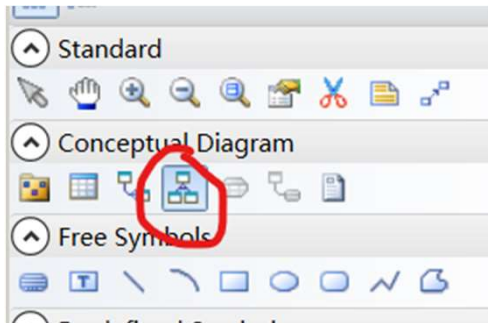
Role name:

☐ Dependent ☒ Mandatory Cardinality: 1..1

More >>

# 继承 (inheritance) 关系

- ▶ 和建立关系类似，只不过需要点击继承按钮。
- ▶ 之后双击中间半圆可以编辑继承属性规则等详细信息。



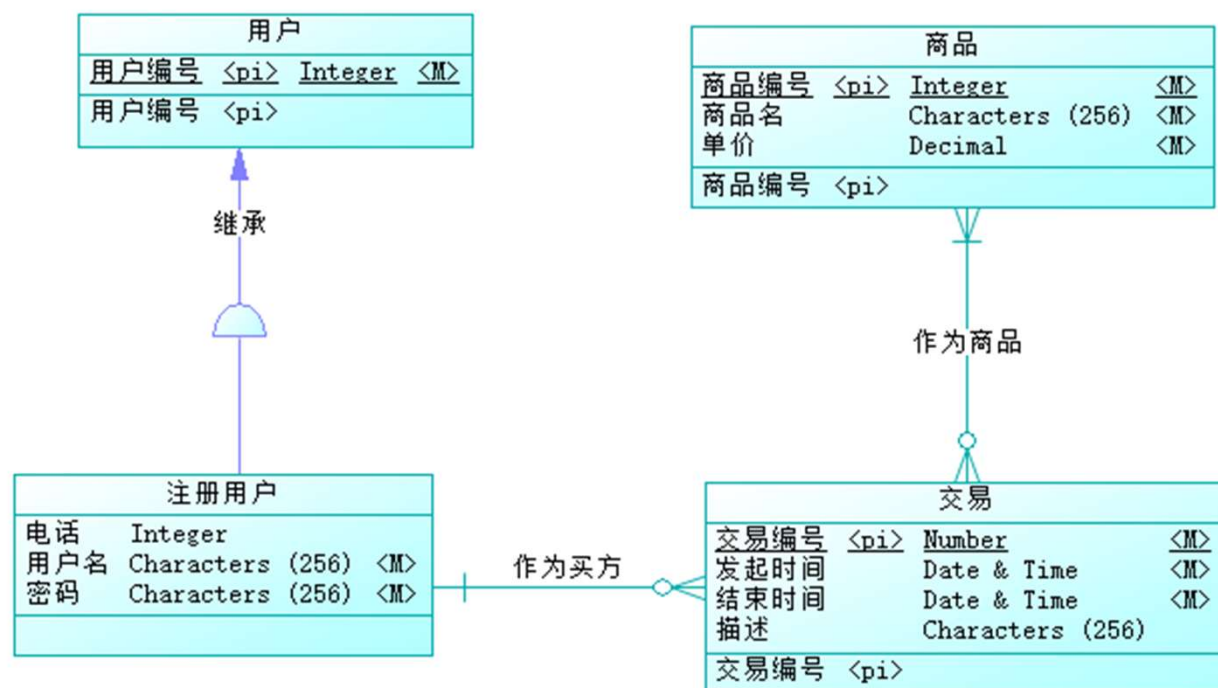
# 注意事项

- ▶ 1. 任意实体的任意主属性在整个模型中都是唯一的，不能存在与主属性同名的属性。
- ▶ 2. 注意属性数据类型的选择。
- ▶ 3. 单纯画CDM可能看不出问题，根据后面CDM到LDM、PDM转换时的报错信息中可以分析自己CDM的问题在哪并进行修改。

# 例题：简单交易系统

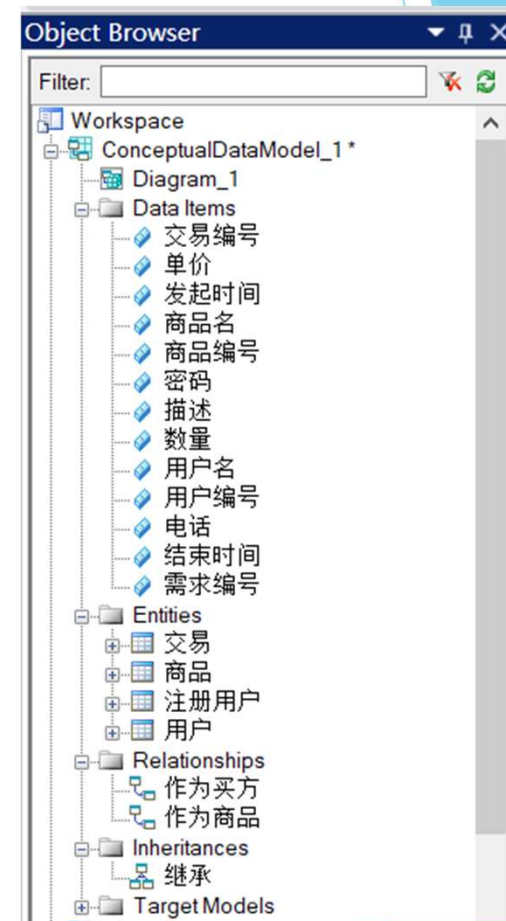
- ▶ 某网站的交易系统描述如下：游客通过注册登录后以注册用户身份进入网站，注册用户 can 参与交易，每次交易可购买网站内的不同商品，即每笔交易只涉及单个用户，可涉及多个商品。
- ▶ 请分析并画出CDM模型图

# 参考答案



# 参考答案

- ▶ 注意：
- ▶ 完成CDM的绘制后可以在左边 object browser看到CDM的所有数据。
- ▶ Data items对应所有实体中包含的 attributes



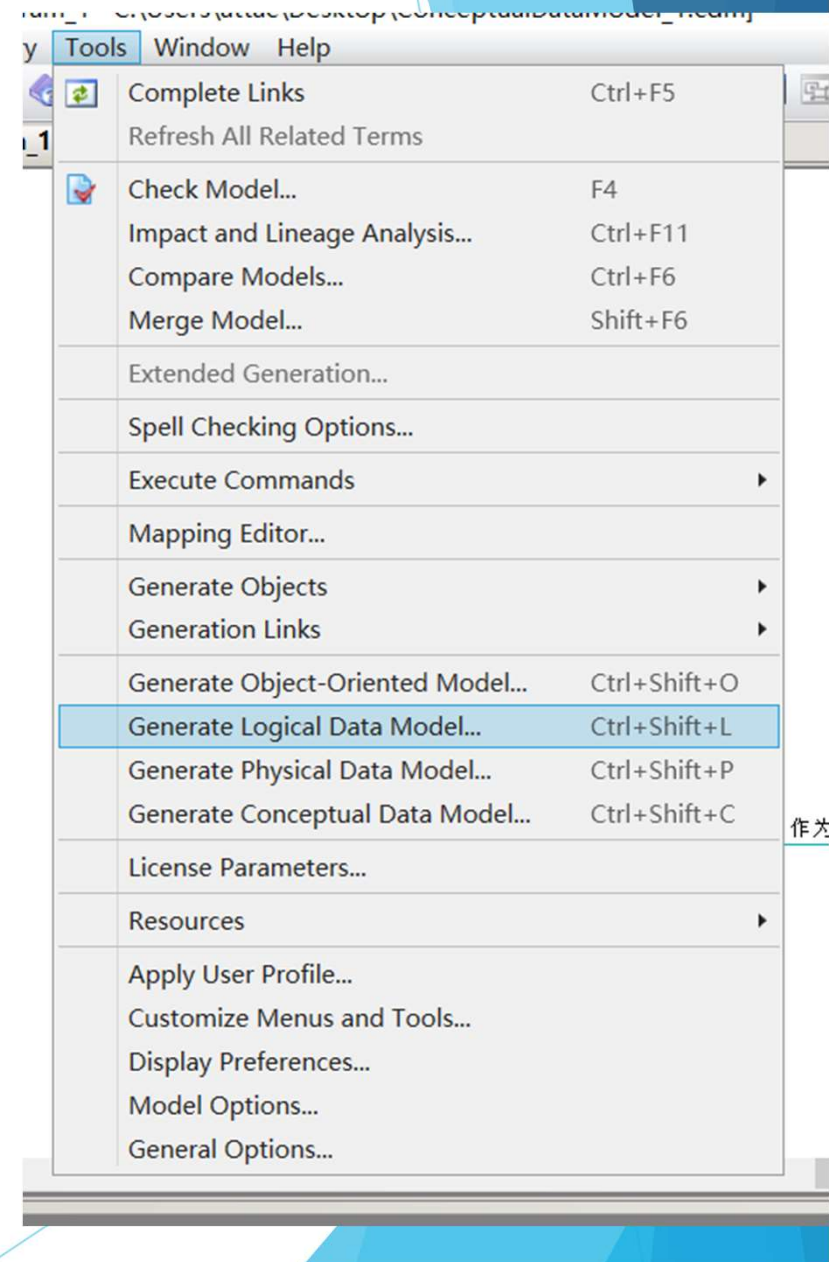
# CDM到LDM转换

- ▶ 使用PowerDesigner可以实现CDM到LDM的一键转换而不必重新绘图。
- ▶ 但转换要求CDM模型规范。可根据报错信息修正CDM模型图。
- ▶ 具体转化体现：
  1. 多对多关系会以增加一个中间实体的方式转化为两个一对多关系。



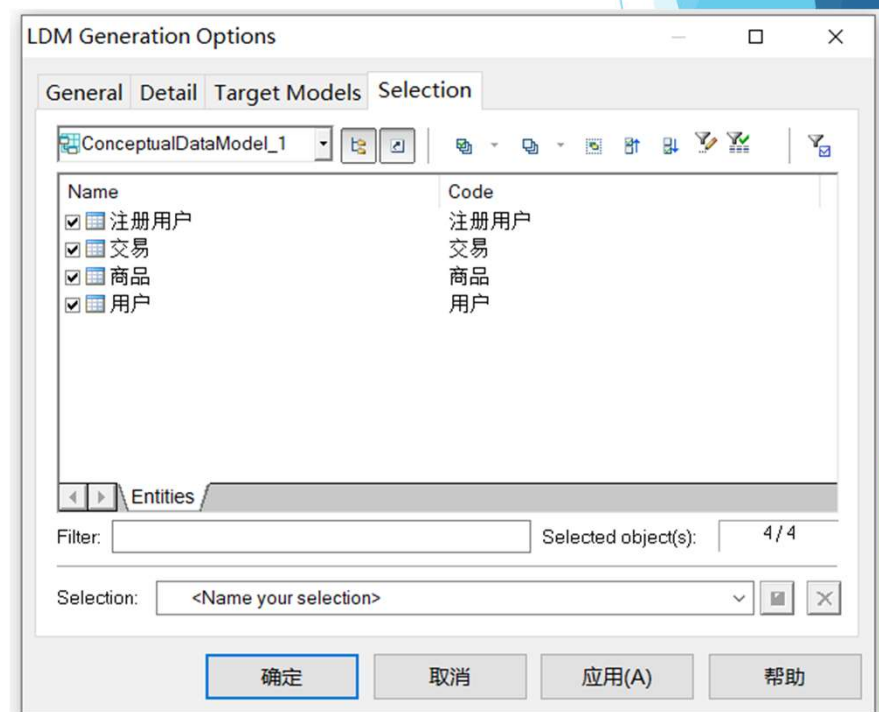
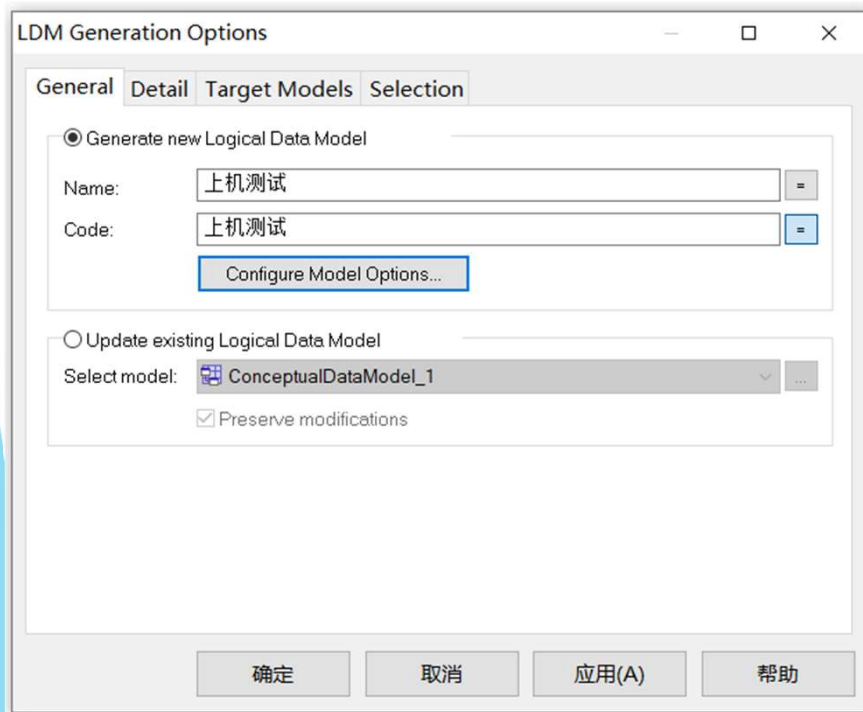
# 操作步骤

- ▶ 点击菜单栏tools – Generate Logical Data Model





# 操作步骤

- ▶ 在弹出的对话框中选择generate new logical data model并选择刚刚绘制的CDM图即可。
- ▶ Selection页中可以选择需要转化的实体有哪些。这里我们默认全选即可。



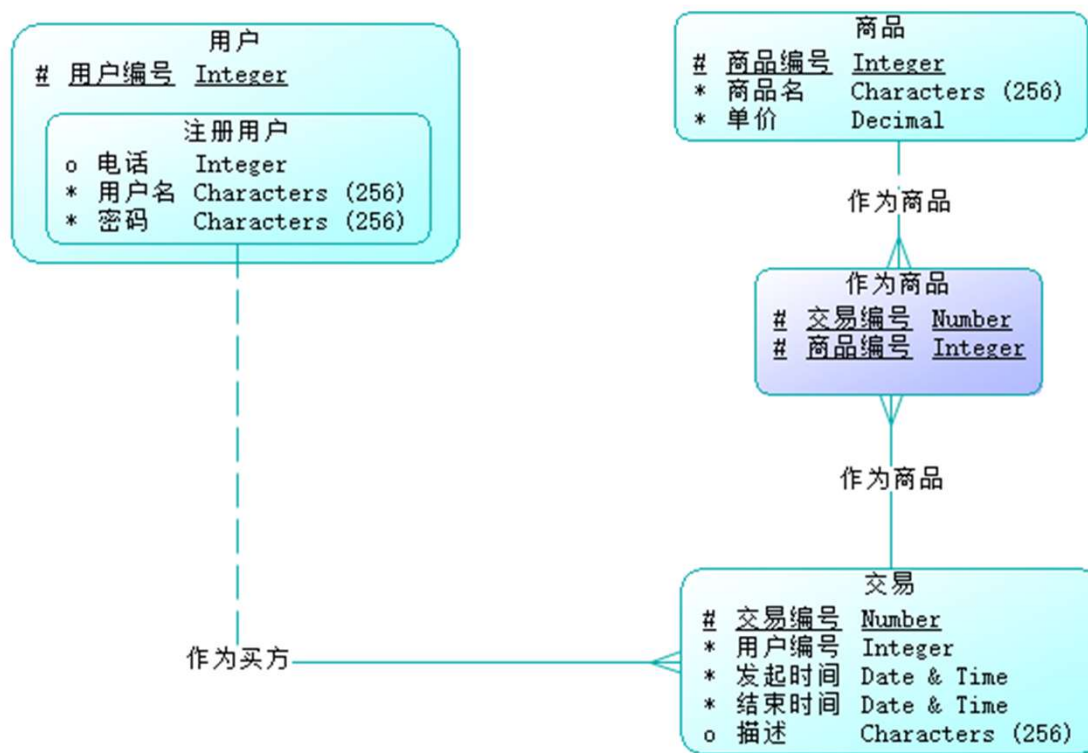
# 操作步骤

- ▶ 自动转换后会在result页面显示转换信息，如有报错则说明CDM绘图不规范，需要根据check部分提示修改CDM。
- ▶ 只有warning可以生成LDM模型，而如果存在error则不会生成LDM模型。

Diagram_1		Result List	
Category	Check	Object	Location
 Data Item	Undefined data type	Data Item "用户ID"	<Model>
 Entity Identifier	Existence of entity attribute	Identifier "交易 交易ID"	<Model>::交易

# 转换结果

- ▶ 上一例题转换结果如图，注意多对多关系以及继承关系在图中的变化。

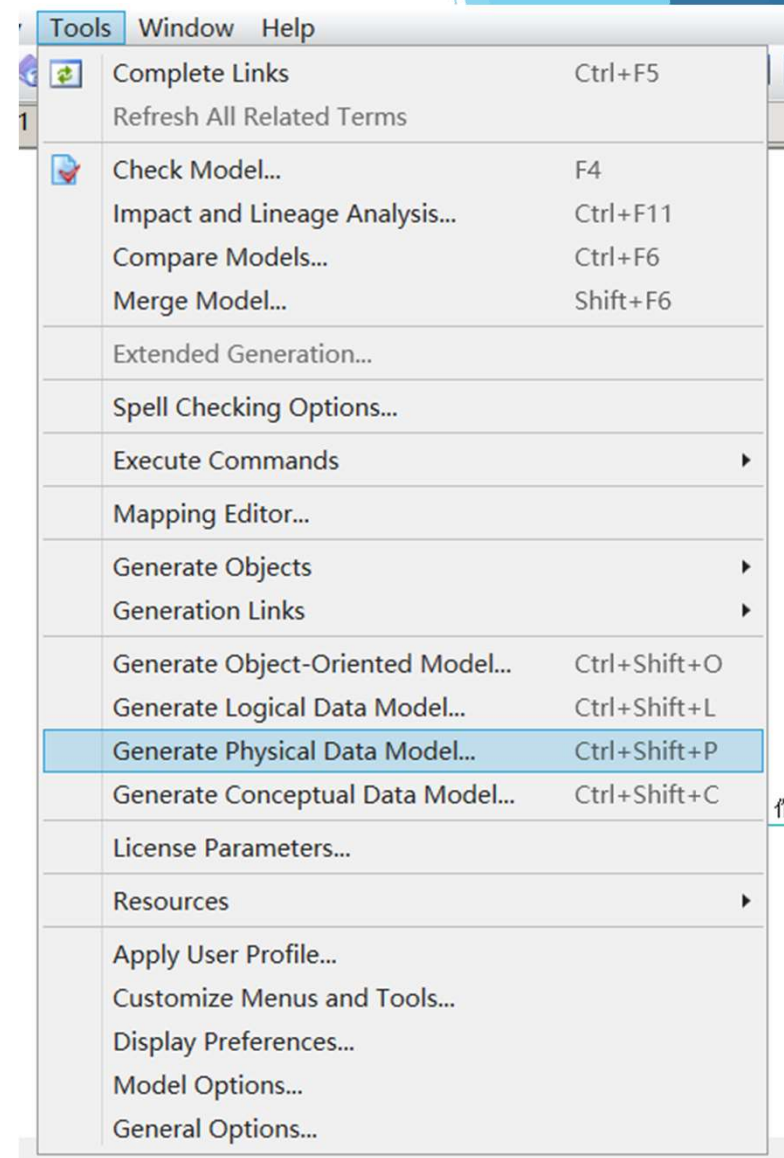


# CDM到LPM转换

- ▶ 我们也可以实现CDM到LPM的一键转换。或者通过刚才的LDM模型生成LPM模型。这里我们采用前者。
- ▶ 与转换为LDM不同的是，LPM的转换需要依赖对应的DBMS。转换之前需要用户选择。

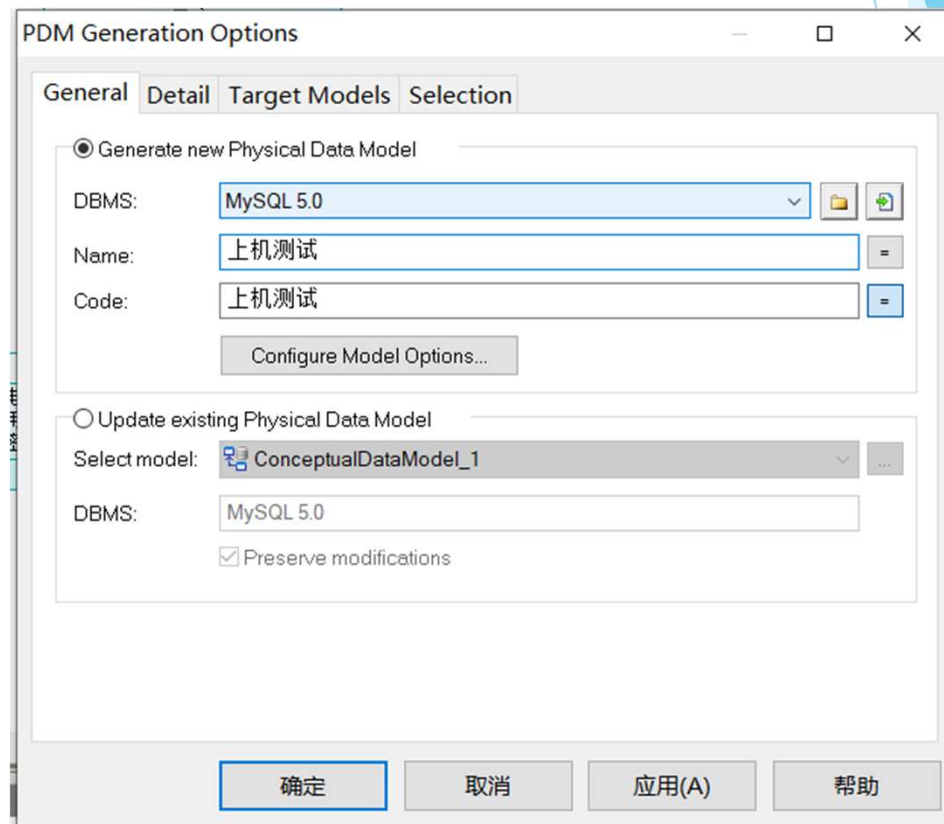
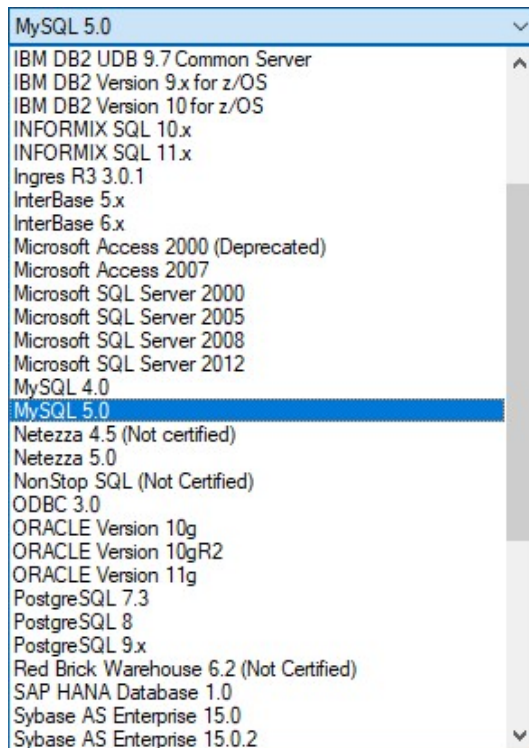
# 操作步骤

- ▶ 与LDM转换类似，在CDM模型图页面点击菜单栏tools – Generate Physical Data Model



# 操作步骤

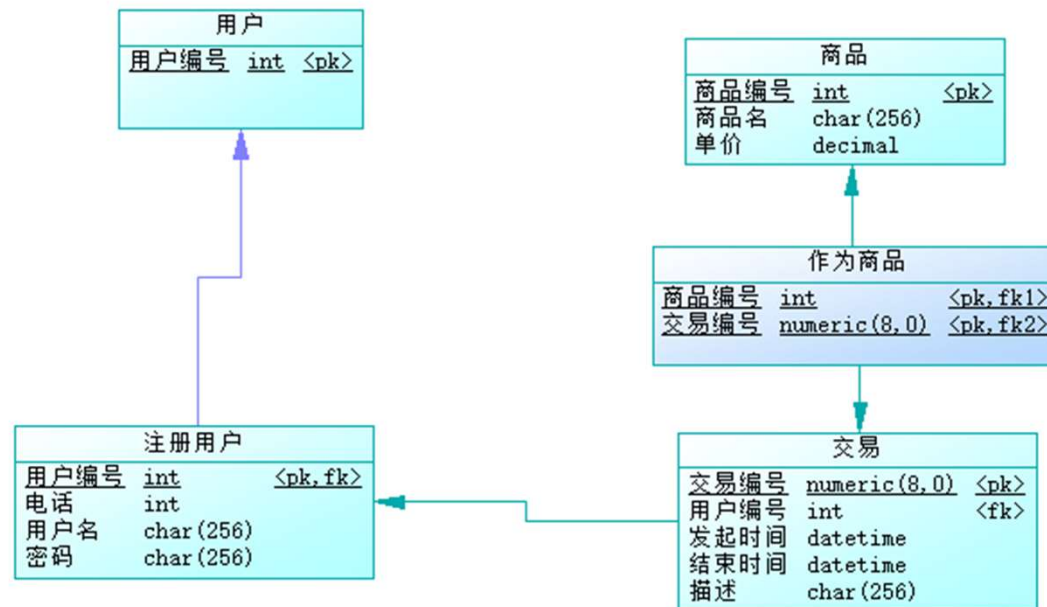
- ▶ 这时我们可以看到生成模型需要指定DBMS。这里我们选择MySQL5.0
- ▶ 点击确定即可转换，与LDM转换相同，会有报错信息提供参考。





# 转换结果

- ▶ 上一例题的PDM转换结果如图，注意多对多关系以及继承关系在图中的变化。





# 转换结果

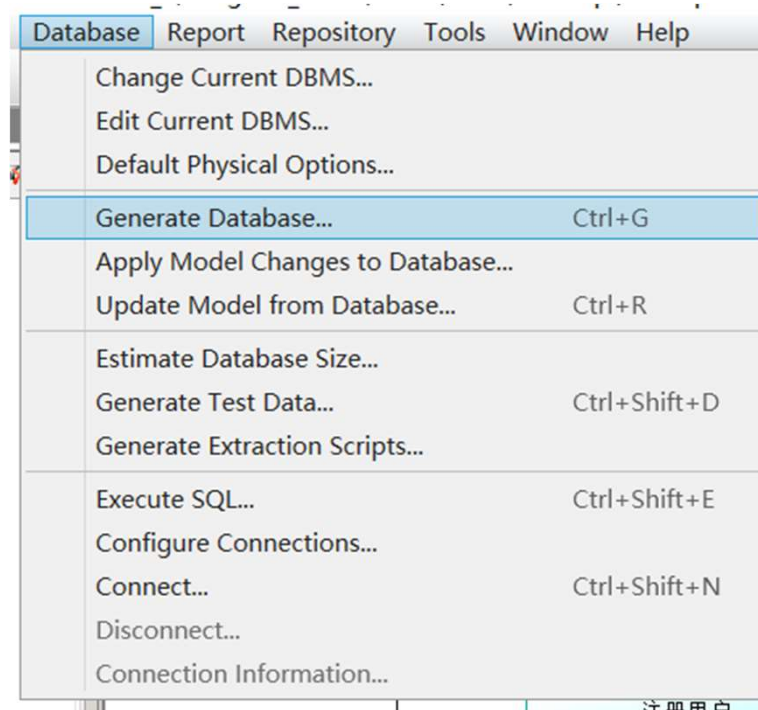
- ▶ 可以看到PDM模型转化后有如下两个明显变化：
- ▶ 1. 属性后有了主键<pk>和外键<fk>的标识。
- ▶ 2. 之前CDM中注册用户没有用户编号，因为这是继承自用户实体的属性。在PDM中注册用户中也出现了父类的属性。
- ▶ 总而言之，PDM更符合数据库的构建标准。
- ▶ 通过CDM到PDM的转换可以高效实现数据库设计。

# PDM生成数据库脚本

- ▶ 转换得到的PDM已经符合数据库建库标准。因此PowerDesigner也可以根据PDM生成对应数据库构建脚本代码。

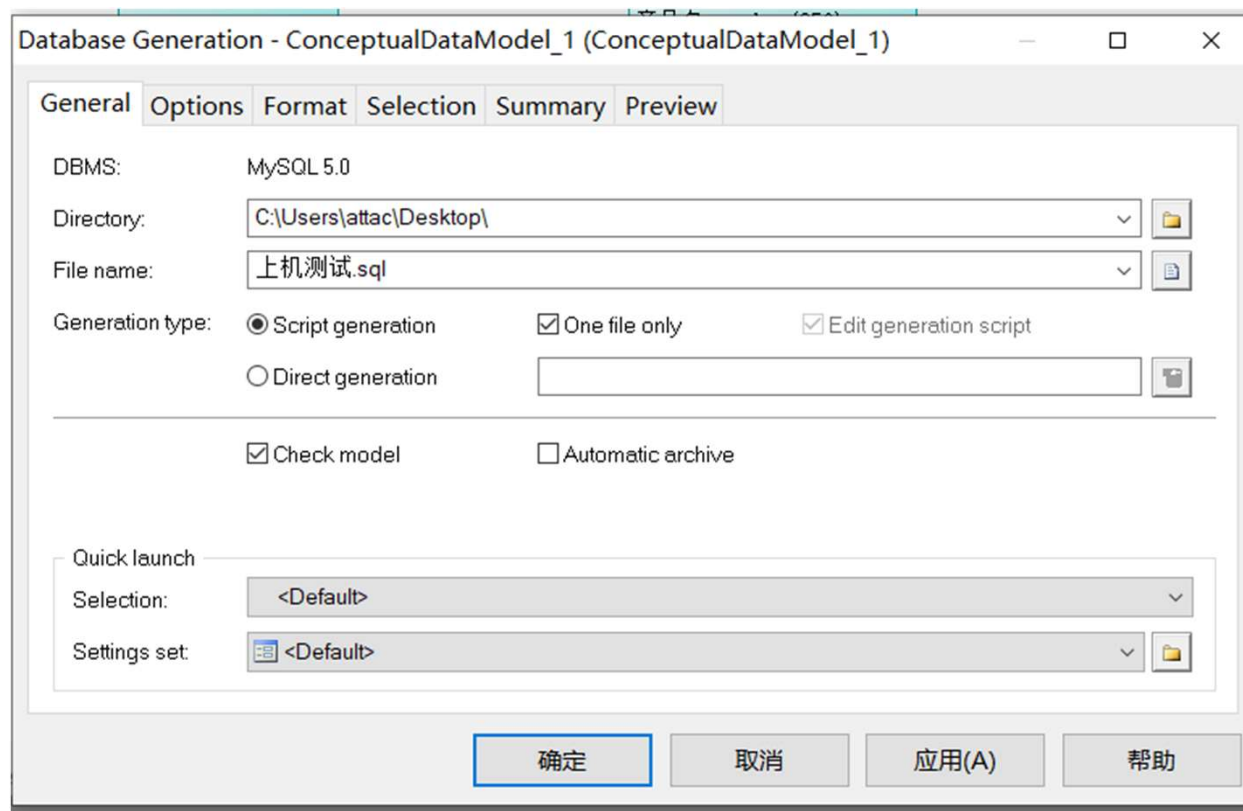
# 操作步骤

- ▶ 在PDM图页面选择菜单栏  
database – generate database



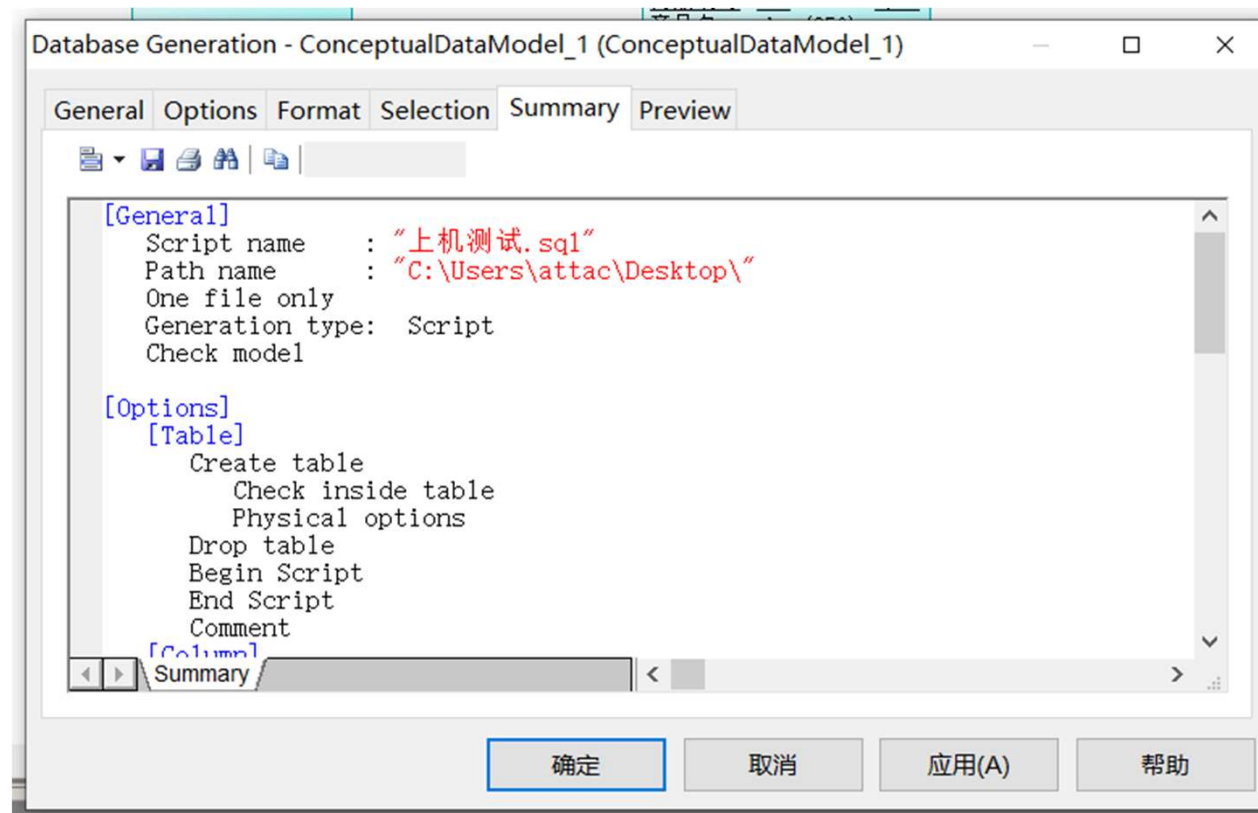
# 操作步骤

- ▶ DBMS是我们不能更改的，与生成PDM时选定的DBMS一致。
- ▶ 注意设置路径与脚本文件名。



# 操作步骤

- ▶ Summary页可以看到生成的sql语句预览。



# 操作步骤

- ▶ 生成后同样会在result list页生成报错信息。
- ▶ 成功会有generated files页面弹出。

