# 数据库设计实验

16337113

劳马东

计算机科学与技术（超算方向）

## 实验目的

掌握数据库设计基本方法及数据库设计工具。

## 实验内容和要求

掌握数据库设计基本步骤，包括数据库概念结构设计、逻辑结构设计、物理结构设计和数据库模式SQL语句生成。能够使用数据库设计工具进行数据库设计。

## 实验环境

|  |  |
| --- | --- |
| 系统 | Windows 10 |
| SQL | MySQL 8.0 |
| 工具 | PowerDesigner |

## 实验过程

设计一个采购、销售和客户管理应用数据库。其中，一个供应商可以供应多种零件，一种零件也可以有多个供应商。一个客户订单可以订购多种供应商供应的零件。客户和供应商都分属不同的国家，而国家按世界五大洲八大洋划分地区。请利用PowerDesigner或者Erwin等数据库设计工具设计数据库。

###### 概念结构设计

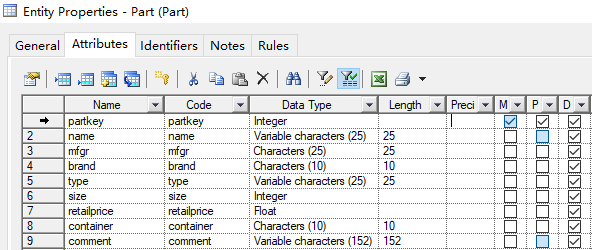
识别出零件Part、供应商Supplier、客户Customer、订单Order、订单项Lineitem、国家Nation、地区Region等7个实体集及其拥有的属性。

* 零件Part：零件编号partkey、零件名称name、零件制造商mfgr、品牌brand、类型type、大小size、零售价格retailprice、包装container、备注comment。主码：零件编号partkey。
* 供应商Supplier：供应商编号suppkey、供应商名称name、地址address、电话phone、备注comment（不需要nation属性）。主码：供应商编号suppkey。
* 客户Customer：客户编号custkey、客户名称name、地址address、电话phone、备注comment（不需要nation属性）。主码：客户编号custkey。
* 订单Order：订单编号orderkey、订单状态status、订单总价totalprice、订单日期orderdate、订单优先级orderpriority、记账员clerk、运送优先级shippriority、备注comment。主码：订单编号orderkey。
* 订单项Lineitem：订单项编号linenumber、零件数量quantity、零件总价extendedprice、折扣discount、税率tax、退货标记returnflag等（不需要partkey和suppkey属性）。主码：订单项编号linenumber。
* 国家Nation：国家编号nationkey、国家名称name、备注comment（不需要region属性）。主码：国家编号nationkey。
* 地区Region：地区编号regionkey、地区名称name、备注comment。主码：地区编号regionkey。

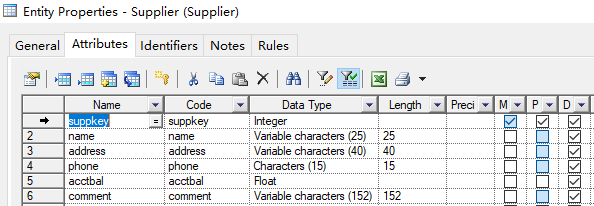
在File->New Model中新建一个Conceptual Data Model。双击图中的第二个图标，即可在Diagram中新建一个实体集，同理双击第三个图标可新建一个联系集。



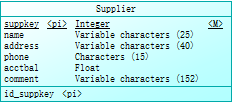
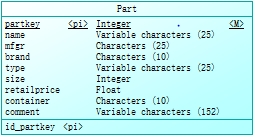
新建Part实体集，双击进入Attribute页添加实体集的属性，分别是partkey、name、mfgr、brand等。在最右侧，勾选P(rimary)以表示partkey是Part实体集的主码，M表示该属性的取值是否唯一（显然主码是Unique的），D(isplay)表示该属性是否显示在实体集上。



以同样的方法新建Supplier实体集，它具有suppkey、name、address等属性，以suppkey为主码。

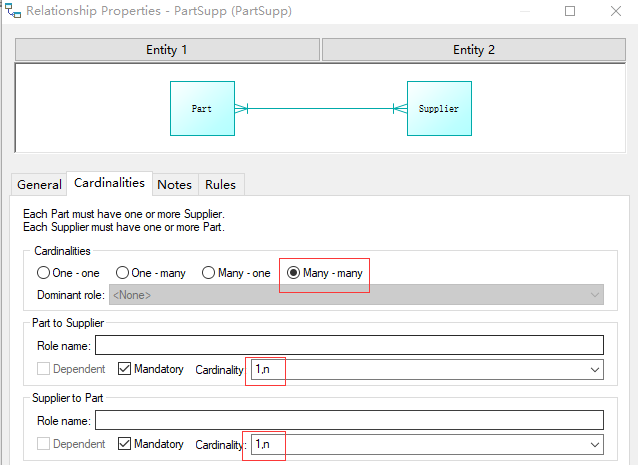


创建完Part实体集和Supplier实体集后，在Tools->Model Options中设置Notation为Entity/Relationship，就能在Diagram中看到如下两张图（默认Notation（Barker）下图形是圆角矩形，而且实体集名字和属性之间没有横线相隔，identifier也不显示）。其中主码用下划线标出，<pi>表示主码，<M>表示属性是否取值唯一，最下方是实体集上的标识符（identifier）。

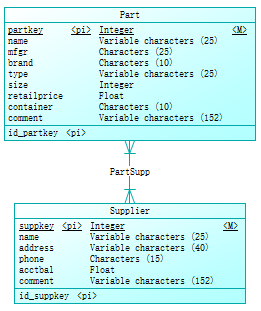


接下来，考虑Part实体集和Supplier实体集之间的联系。一个供应商可以供应多种零件，一种零件也可以有多个供应商，因此二者之间存在一个多对多联系，记为PartSupp。而由于不允许供应商不供应零件，不允许零件没有供应商，因此PartSupp是一个全参与的联系集。

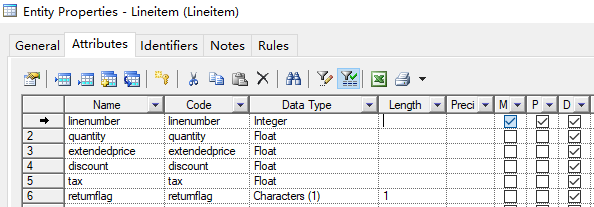
点击右侧的图标双击联系集图标，然后单击Part和Supplier两个实体集，就在二者之间建立了一个联系集（默认为多对1）。双击命名为PartSupp，在Cardinalities页选择多对多（Many-many），修改Cardinality的值为[1，n]（或者勾选Mandatory），以表示实体集Part和Supplier上的所有实体都参与到联系集PartSupp中。



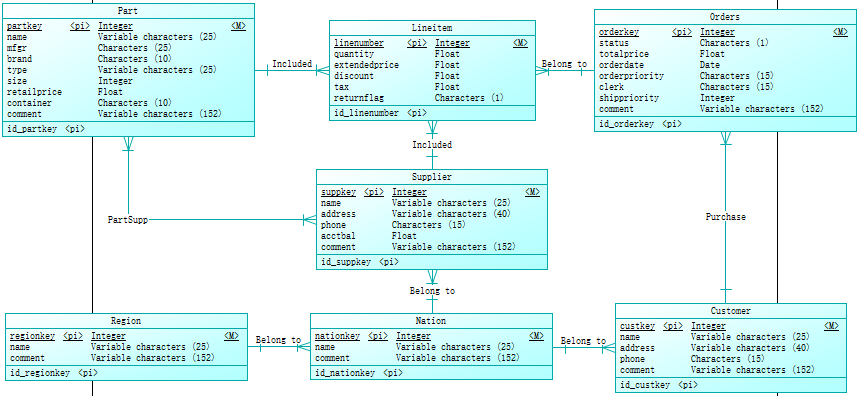
设置好PartSupp后，返回Diagram页面，就能看到如下图。Part和Supplier实体集之间的线表示联系集PartSupp，线两端的三角形表示多（many），横线/竖线表示全参与（部分参与用圆表示）。



以同样的方法创建其他实体集和联系集，需要注意的是**外键属性不需要添加到实体集上**。例如，教材中LineItem实体集具有partkey和suppkey两个外键属性，如果将他们添加到LineItem实体集上，一方面会由于重复使用名为partkey、suppkey这样的Data Item而报错，另一方面即使添加成功，它们也不会被当成外键来对待，之后生成逻辑模型时不会在实体集上以外键显示。



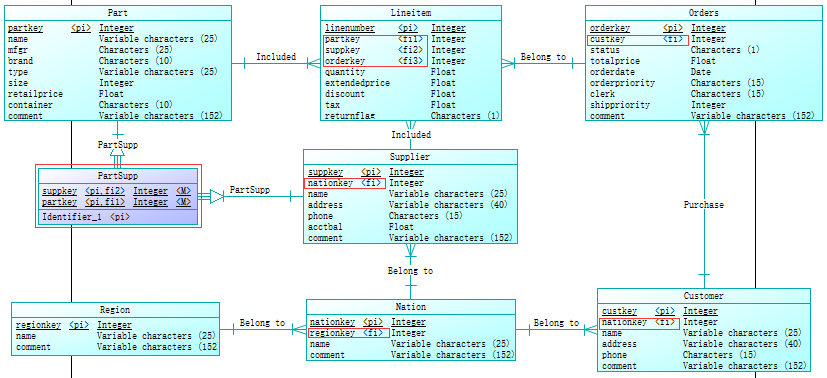
所有实体集和联系集创建完毕，在Symbol->Auto layout中调整图片的布局，能得到下图。



###### 逻辑结构设计

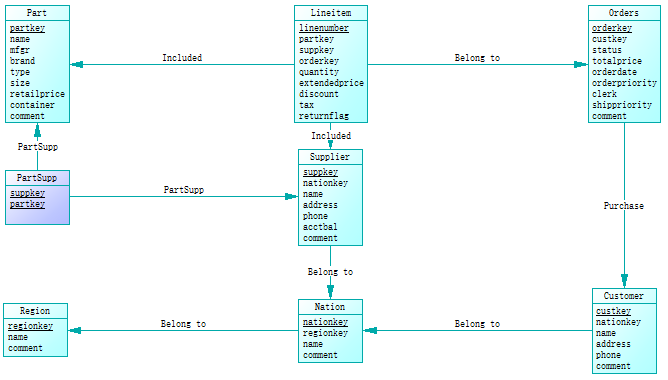
点击Tools->Generate Logical Data Model，设置模型名为Logical Data Model（保证该模型的code名与概念模型不同即可），生成了如下的逻辑模型图。其中，与在概念模型中不同的是，多出了一个PartSupp实体集，其他实体集上添加了一些外键属性（红线框出）。仔细研究，发现有这样的规律：**如果两个实体集之间存在多对一的联系，“一”对应的实体集会将它的主码添加到“多”对应的实体集上去；如果是多对多联系集，二者的主码就会“相遇”而形成一个新的实体集**。这也解释了前面设计概念数据模型时不添加外键属性，因为外键是自动识别并加上去的。

在逻辑结构图中，实体集的主码以下划线和<pi>标识，外键属性是<fi>（PartSupp中的两个属性既是主码又是外键）。



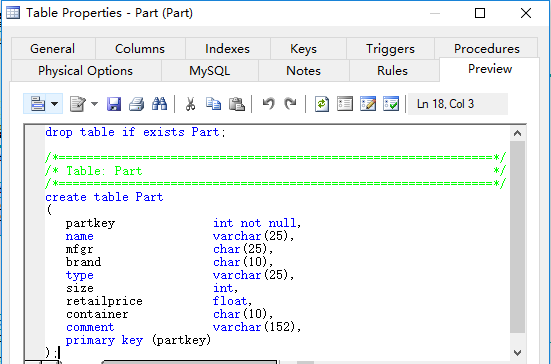
###### 物理结构设计

点击Tools->Generate Physical Data Model，设置模型名为Physical Data Model，选择数据库为MySQL 5.0，生成物理模型如下。主码仍以下划线标识，联系的箭头从“多”对应的关系指向“一”对应的关系。

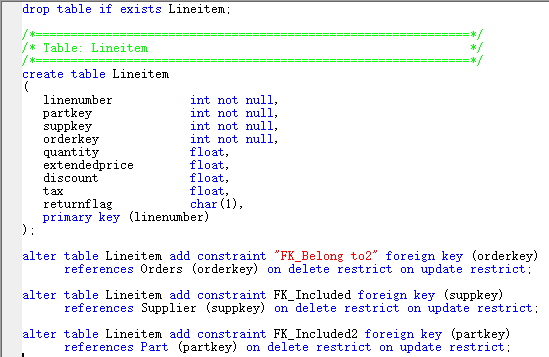


###### 数据库模式SQL语句生成

生成物理模型后，双击Part关系，切换到Preview页，就能看到Part的对应SQL代码，如下图。代码首先判断表格是否存在，如果已有就先删除。主码显式指定为not null，采用表级约束，这样的好处是便于模板化生成代码，不管主码有一个还是多个属性，都只在表级加上主码约束。



另一个比较有代表性的关系时Lineitem，它具有外键属性，创建代码如下图。从代码中可以看到，外键约束是在创建表之后添加的，而不是在列级（实际上对MySQL来说是无用的）或表级（猜测是MySQL 5.0无法添加表级外键约束，因为在MySQL 8.0中表级约束比表外约束的语法更加简单，也具有模板化特性）。



## 实验总结

这次实验与以往的实验有所不同。以往的实验都是已经有了一个设计良好的数据库，所有的数据表（关系）都已经被设计好了，我们实验者的任务是创建这些数据表、在这些数据表上执行查询更新等操作、设置数据表的访问权限等。而在这个实验，我们担负了设计数据表的任务，使用的方法是ER建模技术。

ER建模技术将事物分为实体和联系。实体是那些具体的人、物（如课程），也可以是发生的某件事（如开课、选课）；联系是实体之间的关系（如课程之间的先修关系、老师和学生之间的关系）。每个实体都会有一些属性来表征自己的性质，每种属性上都有对应的值。

配合使用PowerDesigner软件，我很快就构建了该实验的概念数据模型，因为ER建模技术并不难，而且有很强的可解释性。在正式开始实验之前，还发生了一些小插曲。一开始，我安装了PowerDesigner Viewer，这是一个只能查看模型而不能创建和修改模型的软件，这让我纠结了好一阵，明明网上的教程点击File菜单就能新建模型，而我的软件的新建按钮却是灰色的。后来，继续看其他教程才发现了Viewer版和正式版的区别，改装正式版，很快就完成了实验。

我认为，在这个实验中，最重要的技巧是识别出实体集、实体集属性以及联系集。考虑实体集创建少了的情况，例如在大学模式中没有课程这一实体集，而又开课这一实体集，那么一门新课只有在开了之后才能存入数据库中；实体集属性设计不好容易造成冗余，例如在课程和开课中都存储课程名和课程ID，在这里开课的课程名就是冗余的，实际上只需存储课程ID，就能在课程中查询到与这个唯一的ID对应的课程名；联系集的设计比较考验设计者的能力，一方面需要判断两个实体集之间有没有联系，另一方面要设计是何种联系（一对一、多对一、一对多、多对多）。

以上就是我在这次实验中的收获和感想。