# Food Mart的商务智能

## 背景介绍

Food Mart是一家跨国食品超市，每天有成千上万条来自世界各地的销售、库存等记录在其Food Mart数据库中诞生。基于大数据的背景，该公司的管理层决定为Food Mart建立数据仓库，期望能从庞大的数据中挖掘出有商业价值的信息，来进一步帮助管理层进行决策。

由于数据仓库的建立是个漫长的过程，需要分期进行。

在项目初期，管理层决定：

先用销售数据建立数据仓库；

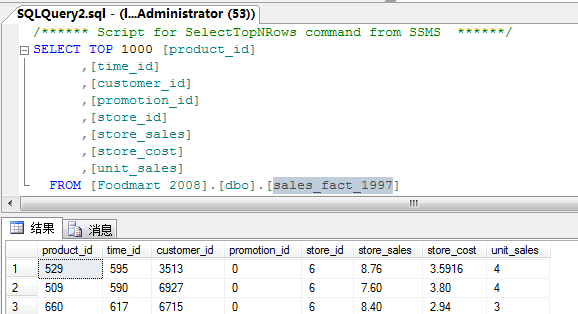
然后，在新建的数据仓库上进行OLAP（联机分析处理）；

接着，基于数据仓库建立一个数据挖掘模型，挖掘（研究）办理不同会员卡的老客户有怎样的特征，进而根据新客户的信息，为新客户推荐不同的会员卡，给公司带来利润；

最后，用Food Mart库存数据扩展上述数据仓库的内容，并完成上述相似的工作。

## 需求分析

现在，需要先用销售数据建立数据仓库，Food Mart数据库中记录销售数据的主要是sales\_fact\_1997表，如下图所示：



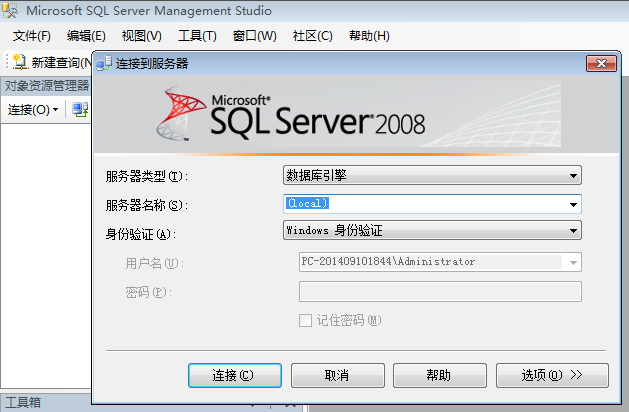
从图中可以看出，sales\_fact\_1997表有五个外键：product\_id，time\_id，customer\_id，promotion\_id，store\_id，所以还需要五张表：product，time\_by\_day，customer，promotion和store。

进而，初步分析得出，目标数据仓库有一张事实表：sales\_fact\_1997，并以product，time\_by\_day，customer，promotion和store这五张表作为维表，而sales\_fact\_1997表中的store\_sales，store\_cost和unit\_sales属性作为事实表的度量。

## 建立数据库

由于Food Mart公司已经有数据库，并且为我们提供了1997年和1998年的数据库备份foodmart.bak，所以我们只需要把数据库备份文件导入到我们的数据库中就可以了。

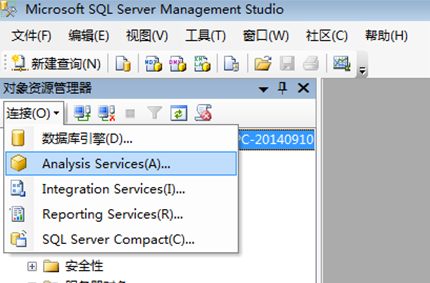
打开SQL Server Management Studio ，弹出如下界面：



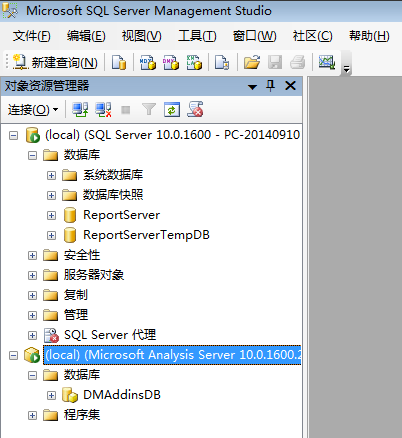
按上图选择和输入相关信息，点击“连接按钮”，进入数据库引擎界面：



接着按下图方式打开Analysis Service（此步骤方便对Analysis Service更好理解）：

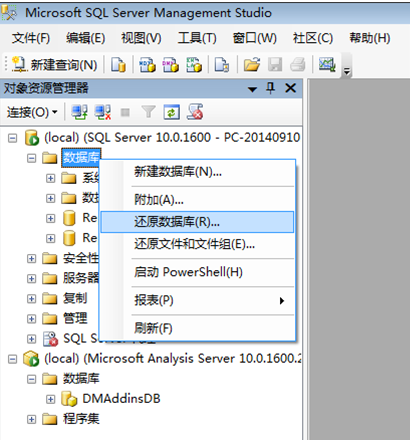


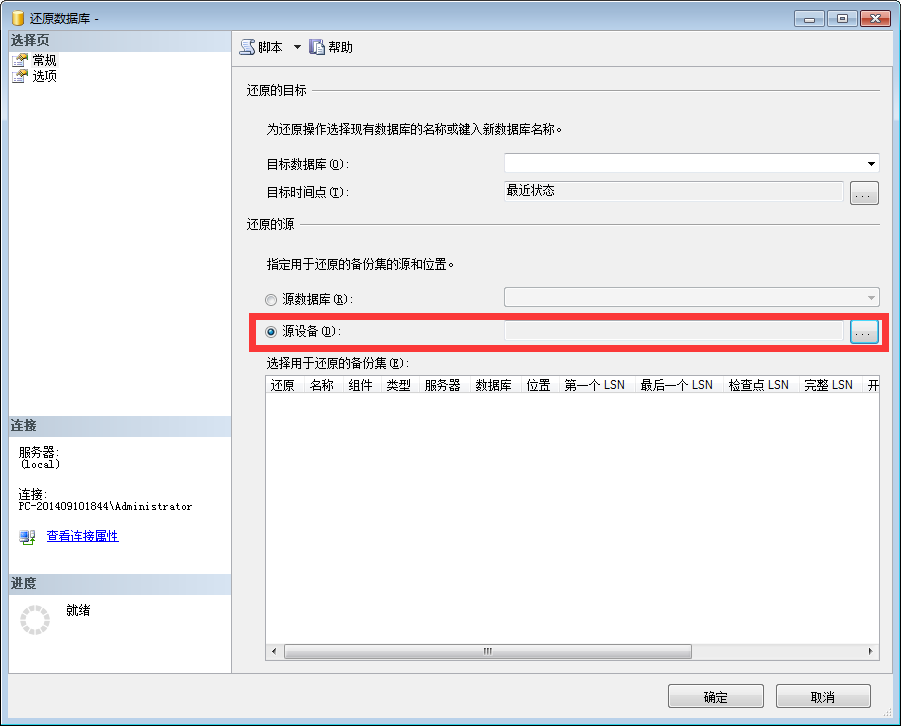




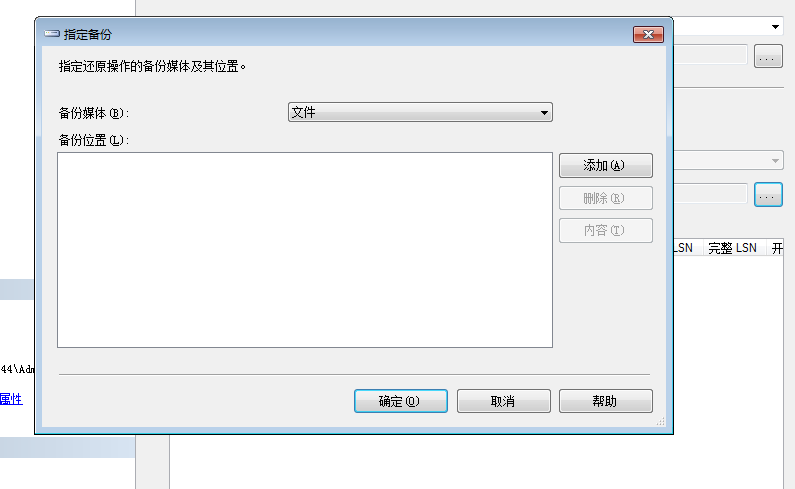
现在，进行数据库备份的导入：

右键点击“数据库”，选择“还原数据库”

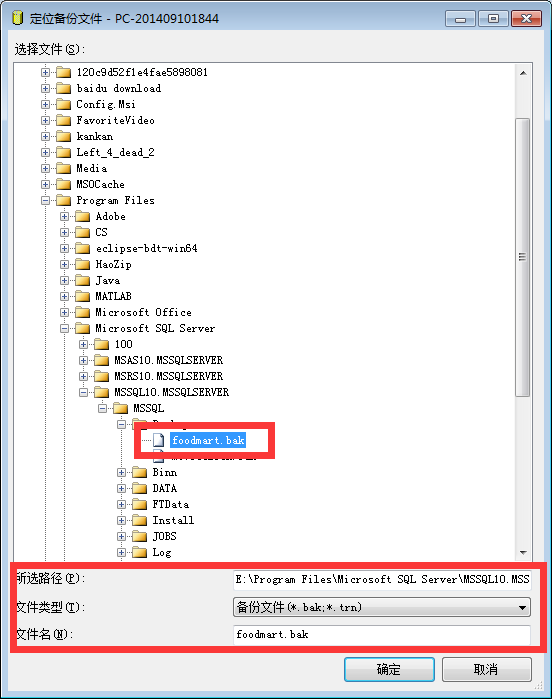




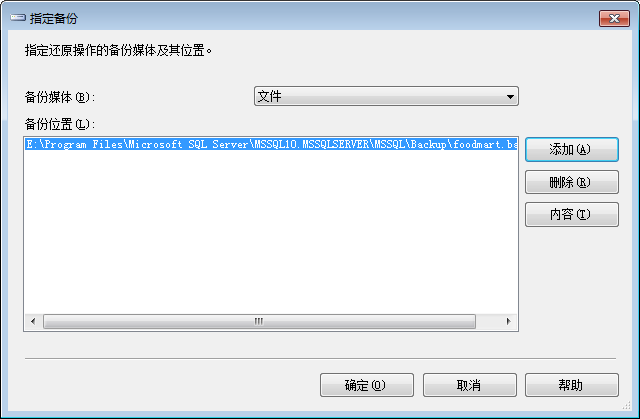
点击上图的“….”按钮得到如下弹出：

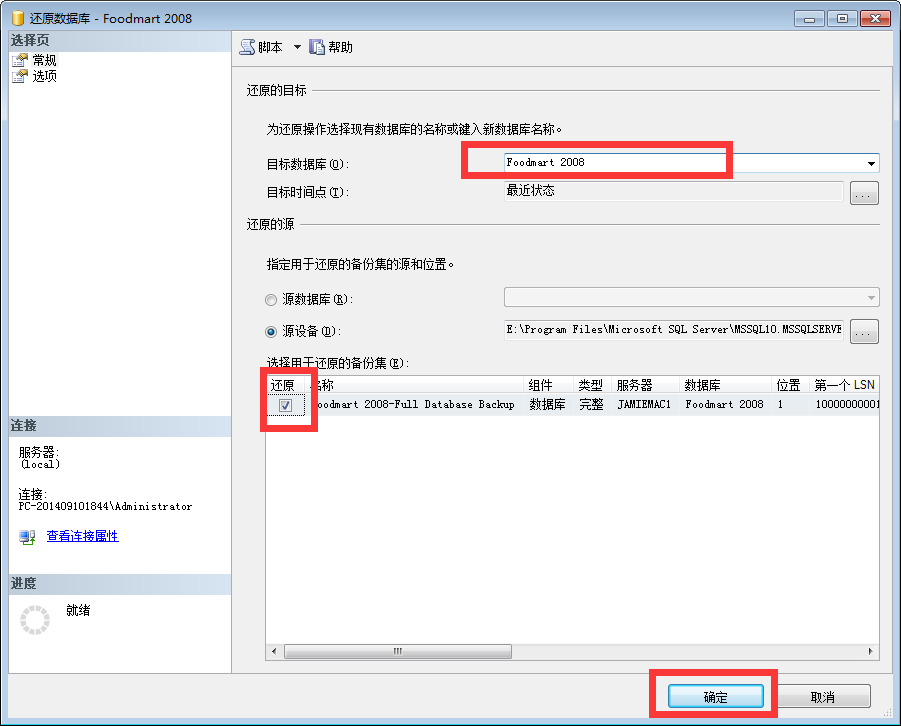


点击“添加”按钮，把数据库备份文件foodmart.bak添加进去：



点击“确定”按钮：





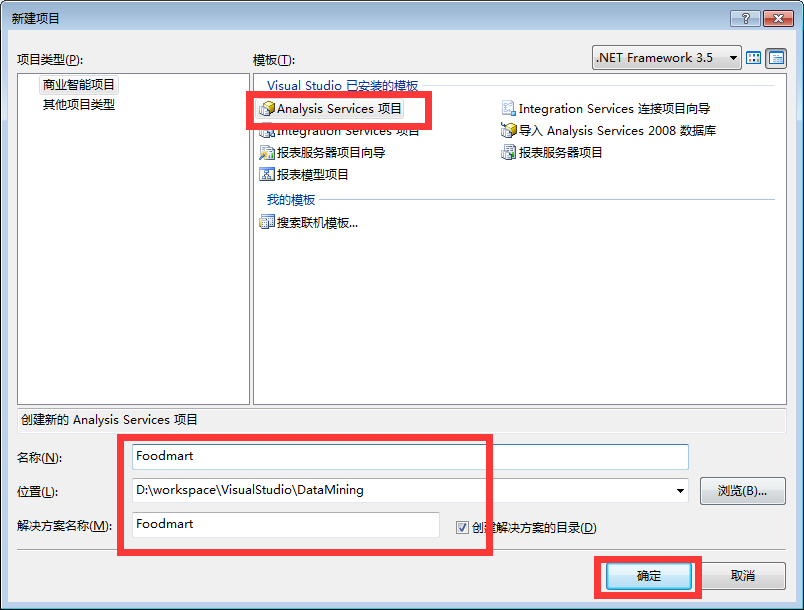
这时候，Foodmart 2008数据库便导入到了我们的数据库中。



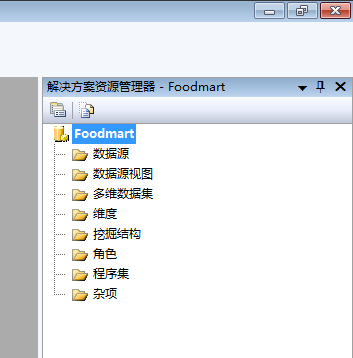
## 新建项目

打开Microsoft Visual Studio 2008，新建一个项目：





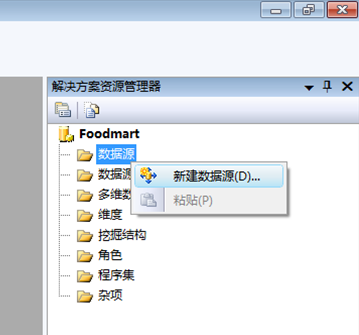
点击“确定”按钮，得到了如下空项目

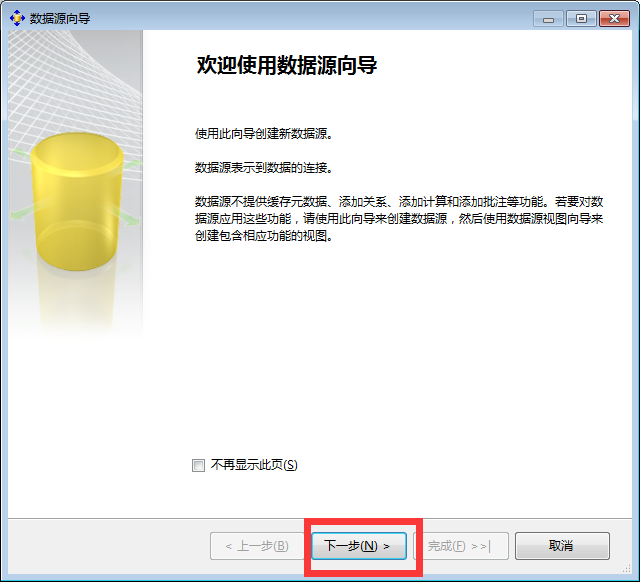


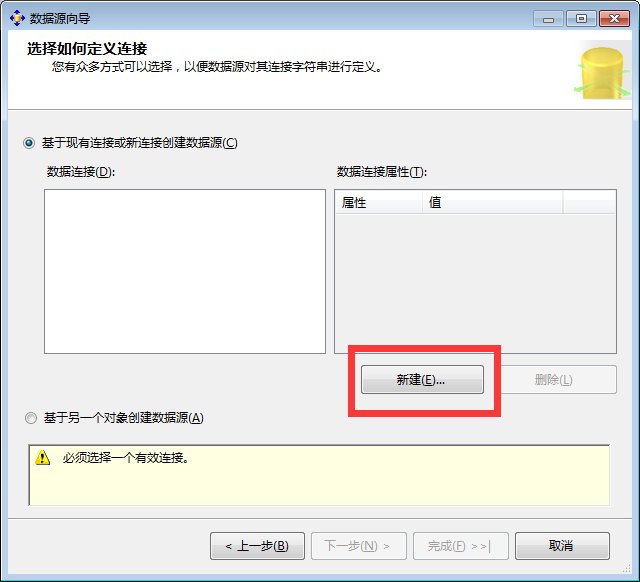
从上图可以看到，我们需要完成“数据源”，“数据源视图”，“维度”（注意顺序，这里先定义维度表），“多维数据集”（数据仓库），“数据挖掘”的一系列流程。下面，我们一起一一展开进行。

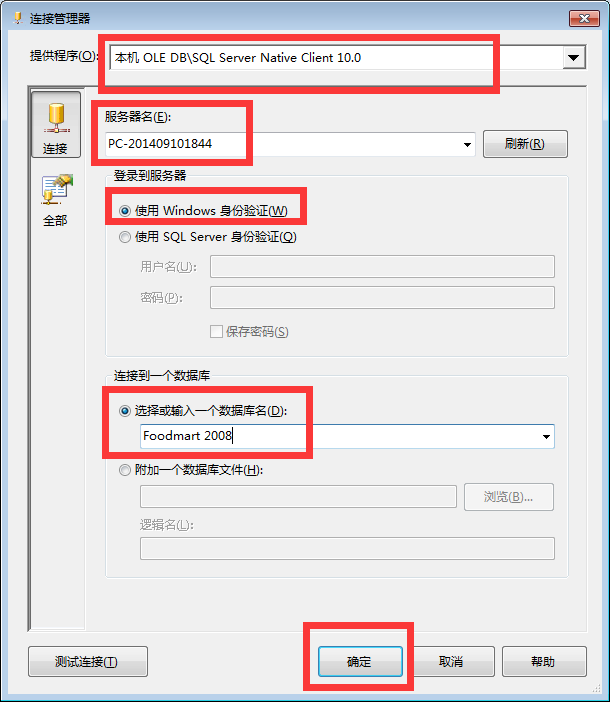
## 导入数据源

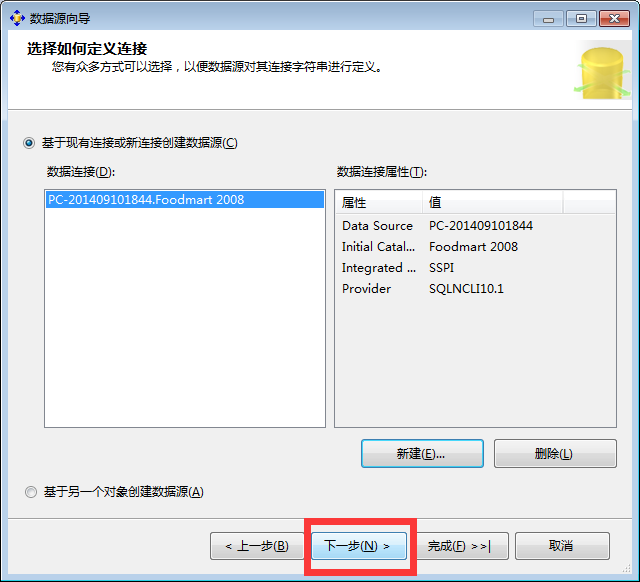
右键点击“数据源”来“新建数据源”：

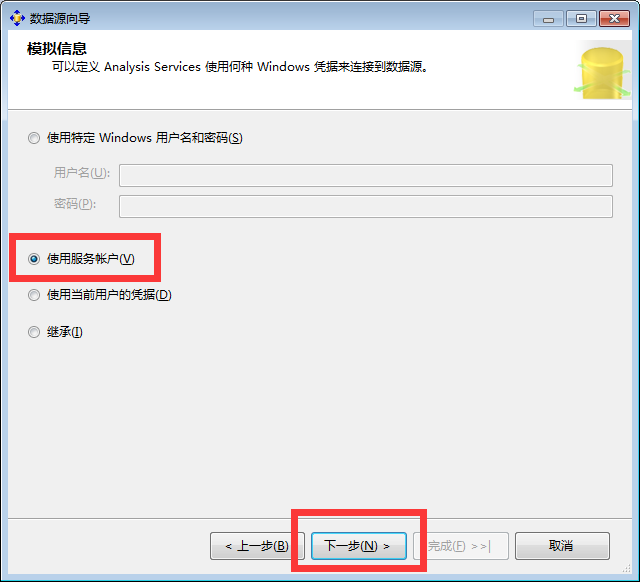








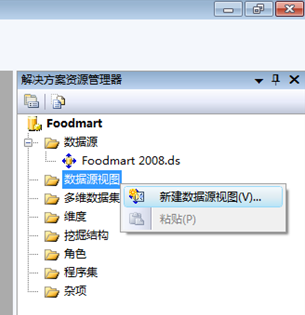


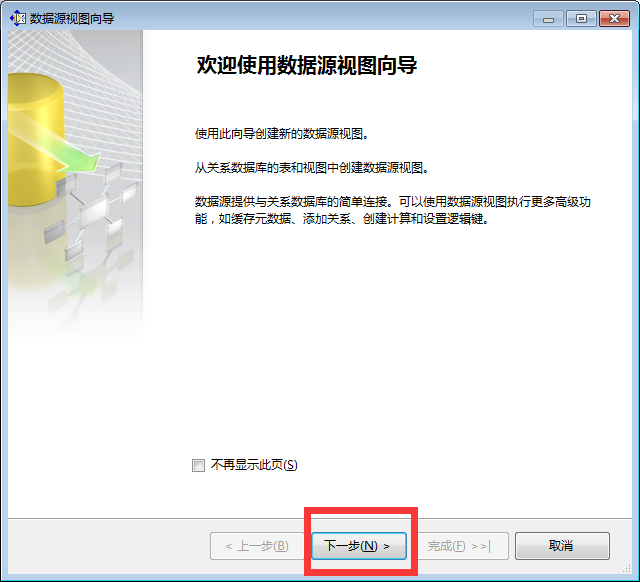


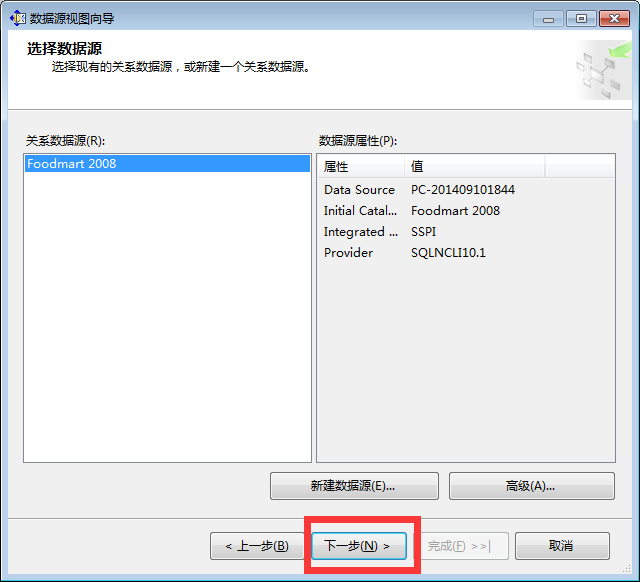
## 建立数据源视图

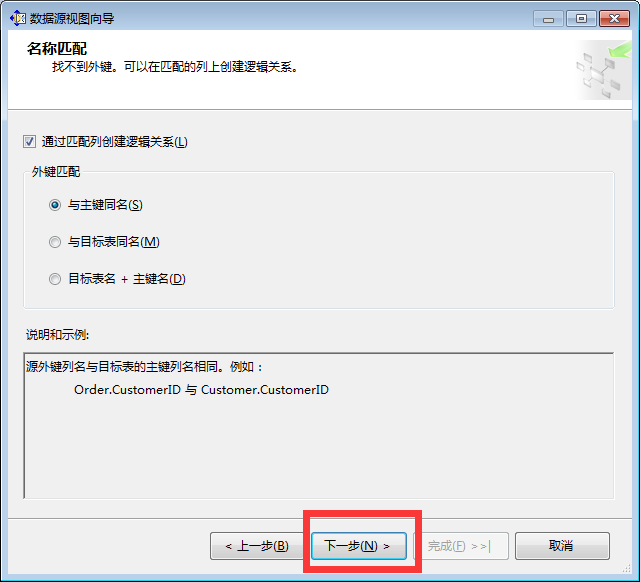
为项目建立项目与数据库之间的映射表，这里不需要把所有数据库中的表引入到数据视图中，需要什么表，就引用什么表。从需求分析里，我们知道需要一张事实表sales\_fact\_1997表，和product，time\_by\_day，customer，promotion，store五张维表。（注意：这里还没有事实表和维表，只是单纯的六张数据表，事实表和维表需要后续设计产生）

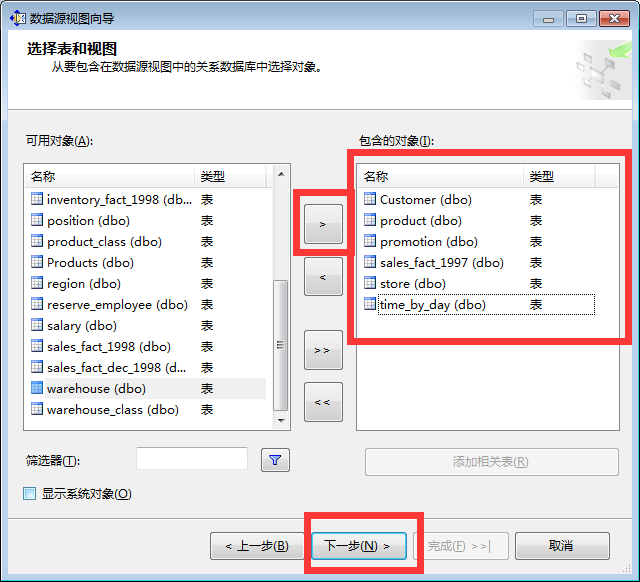
右键点击“数据源视图”：

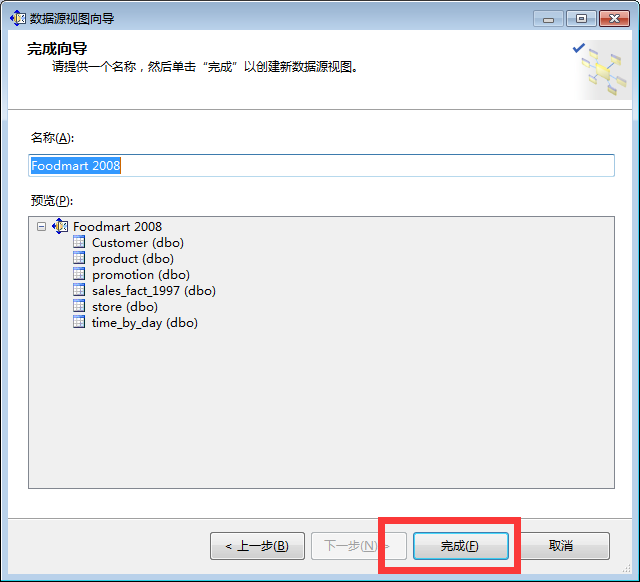


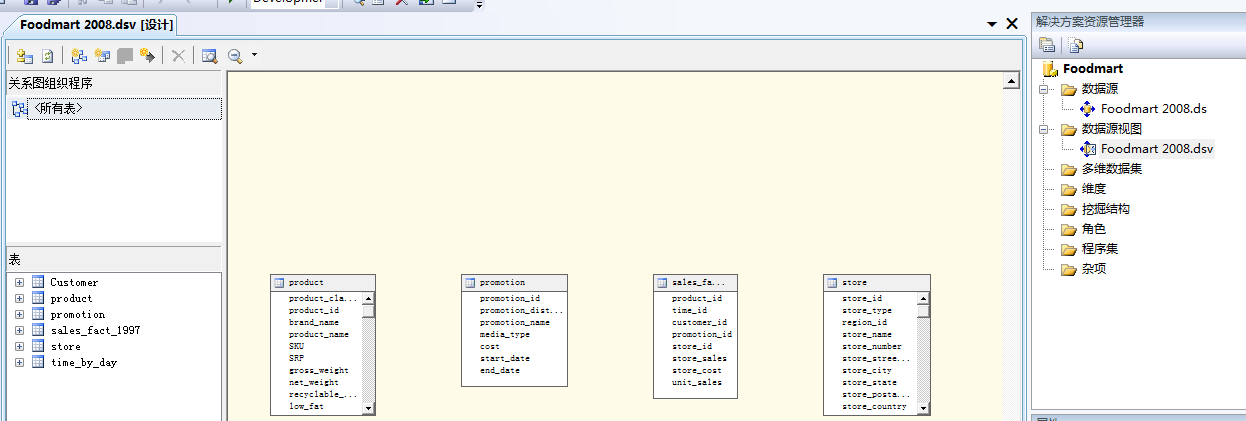






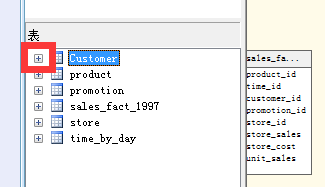


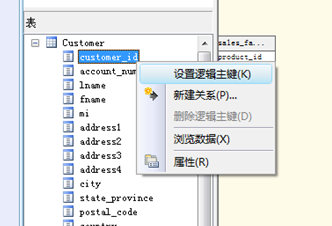


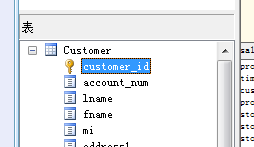


这里，我们就初步建立好数据源视图。但是，可以发现这些表都是独立的，互相之间没有关联关系，这是因为数据库中的数据表并没有设置主键和外键。因而，接下来的工作是给这些表建立主键和表与表之间的联系。

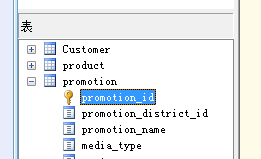
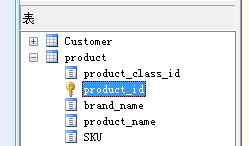
首先，为这些表设置“逻辑主键”：

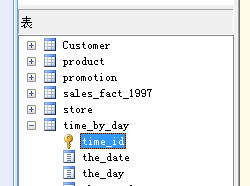
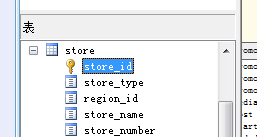




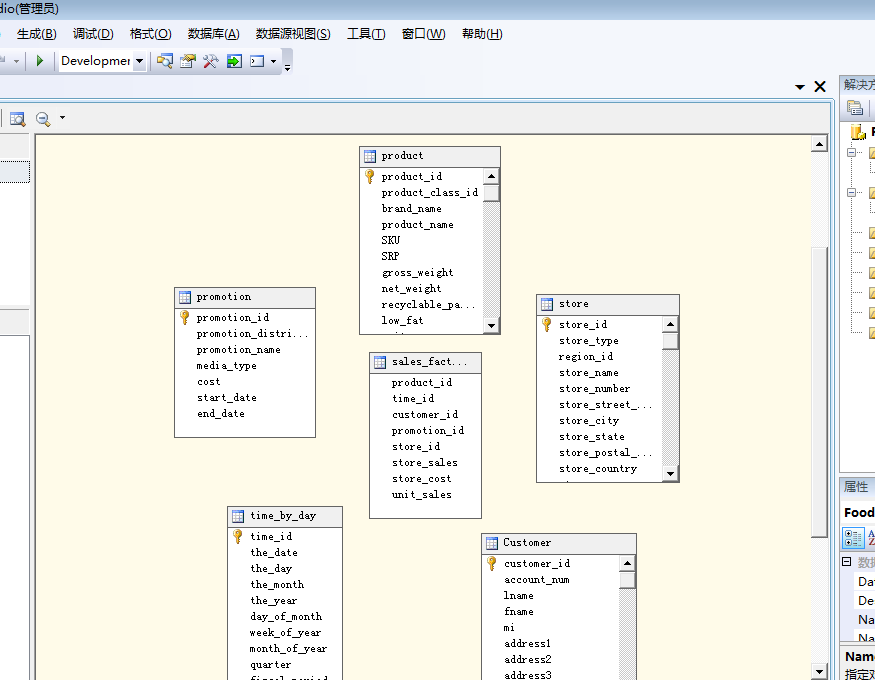


与上述操作类似，设置其它表的“逻辑主键”，得下图：

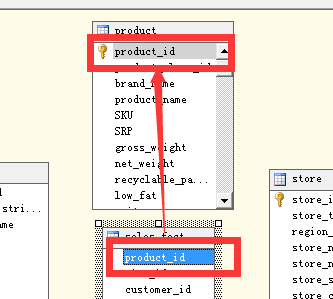
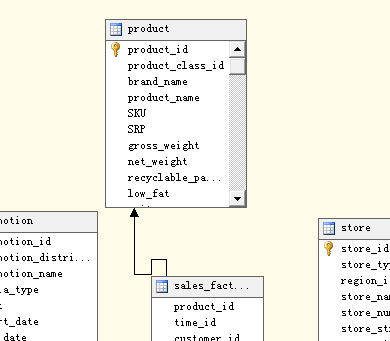




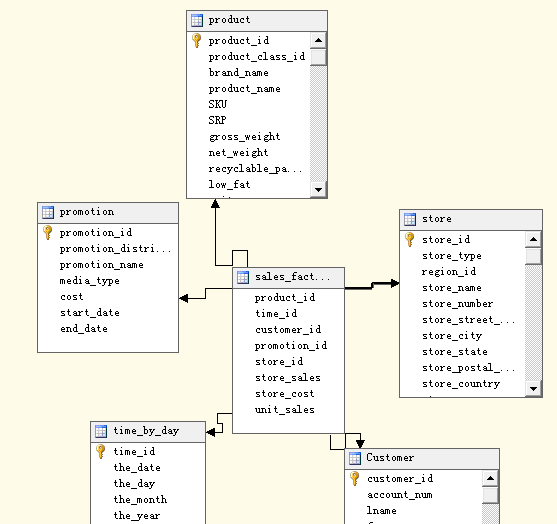
因为sales\_fact\_1997表将要作为事实表，而product，time\_by\_day，customer，promotion，store五张表将要作为维表。所以将sales\_fact\_1997表放在中间，product，time\_by\_day，customer，promotion，store五张表围绕着sales\_fact\_1997表放置：



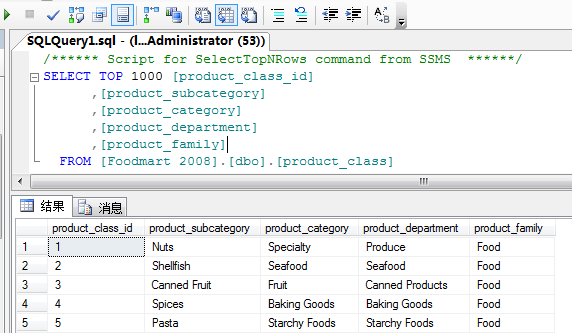
接着，建立表与表之间的联系，把sales\_fact\_1997表的“product\_id”属性拖曳到product表的“product\_id”属性上：

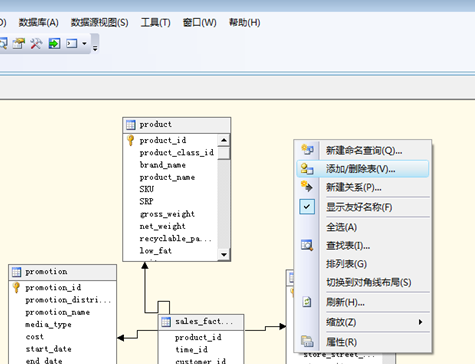
然后把sales\_fact\_1997表的time\_id，customer\_id，promotion\_id和store\_id属性分别与time\_by\_day，customer，promotion和store的相应属性连接：

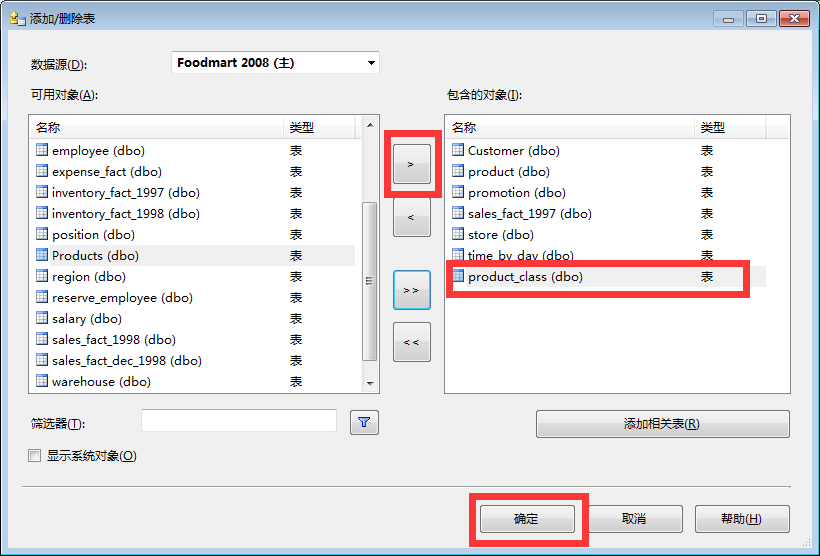


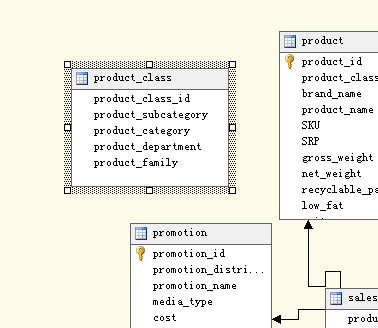
这里发现，product表有个“product\_class\_id”属性，它是与product\_class表的“product\_class\_id”主键相连，而product\_class表里存放了产品不同层级的分类信息，这些信息对于维表的层次结构建立非常有利，所以需要引入。



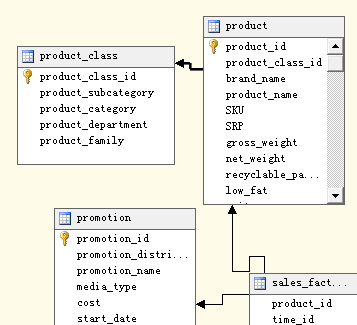
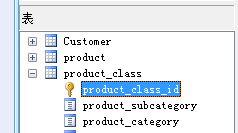
右键选择“添加/删除表”，把product\_class表引进来：



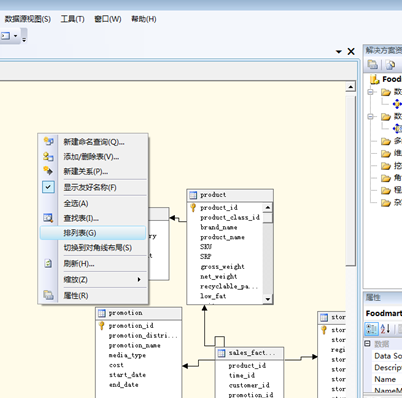


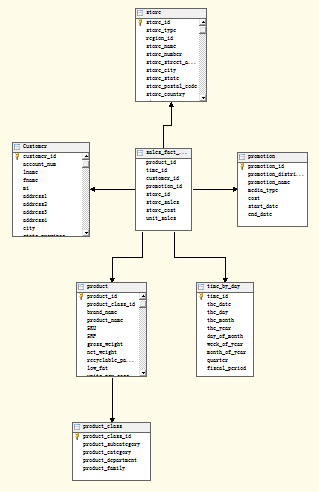


设置product\_class表的“逻辑主键”、product\_class表与product表的连接：



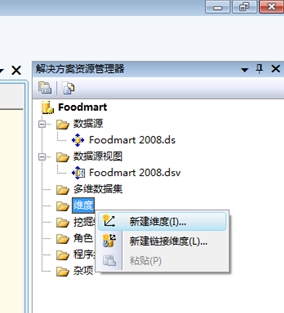
右键选择“排列表”，来整理一下视图：



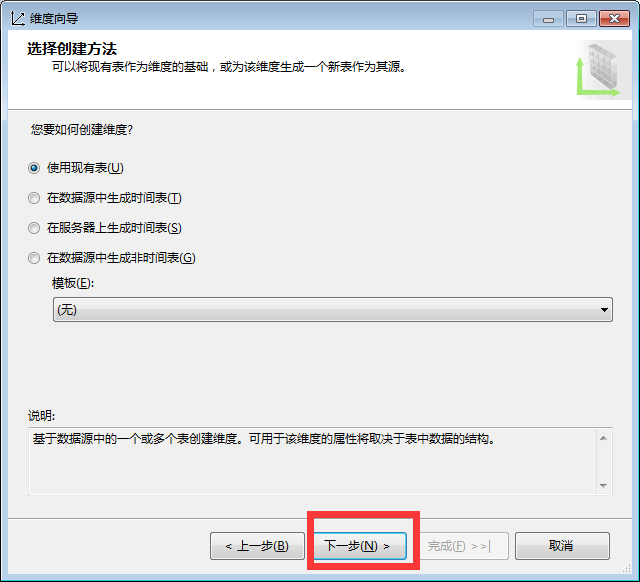


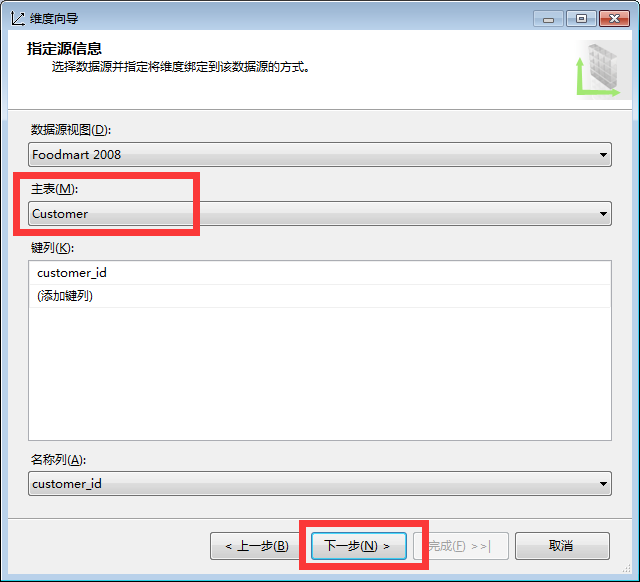
## 建立维表

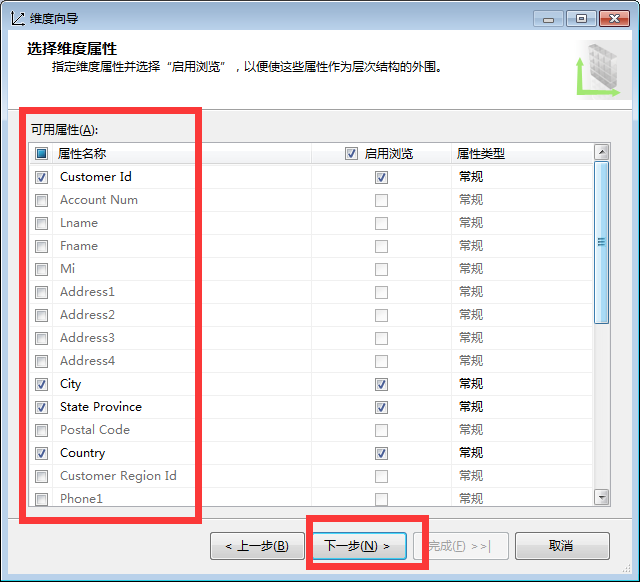
右键点击“维度”来“新建维度”：

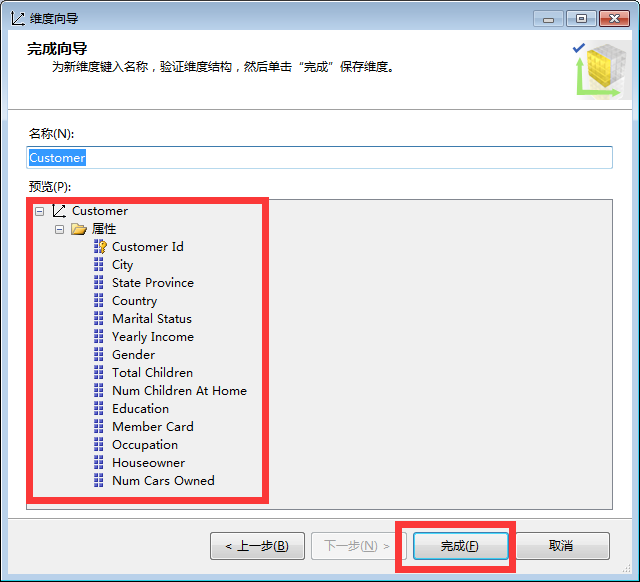






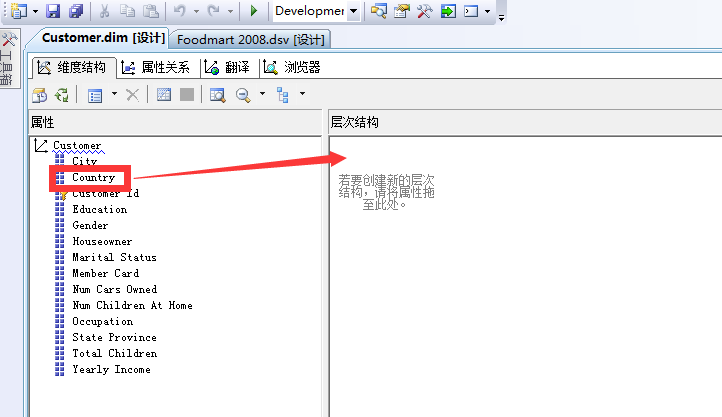


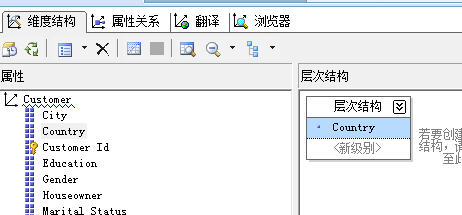




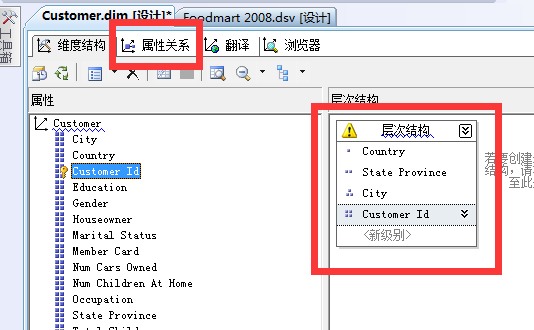


现在为Customer维表建立层次结构，将“Country”（国家）作为顶层，拖曳到层次结构窗口中：



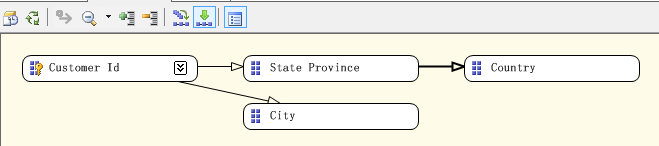


接着，分别把“State Province”（省）、“City”（城市）、“Customer Id”（具体客户）以同样方式拖曳到层次结构中：

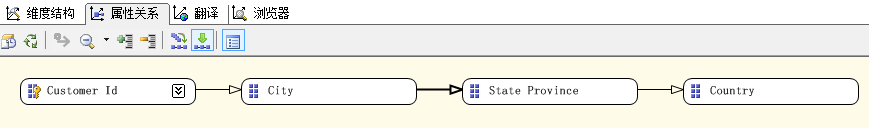


点击“属性关系”标签，将“State Province”拖曳到“Country”上：

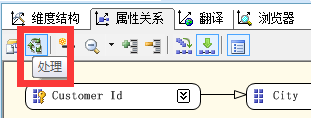


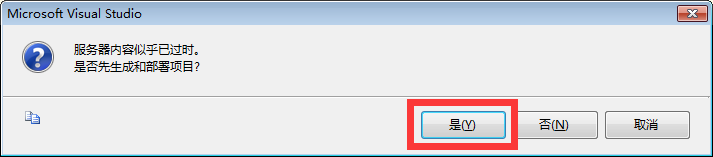


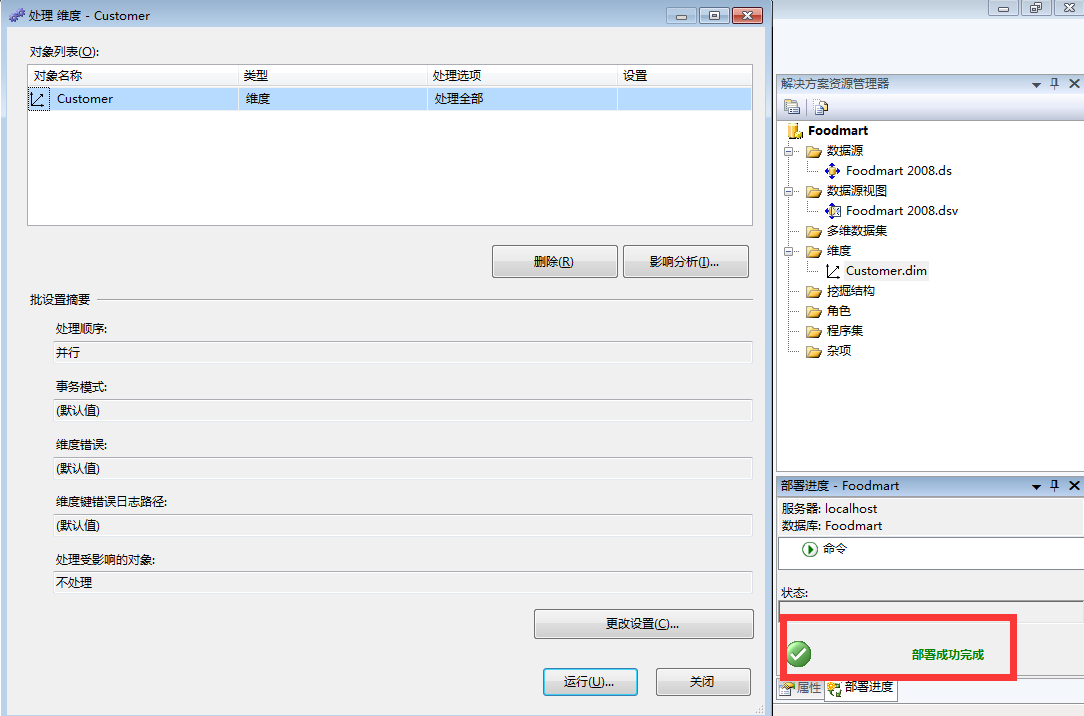
再将“City”拖曳到“State Province”上：



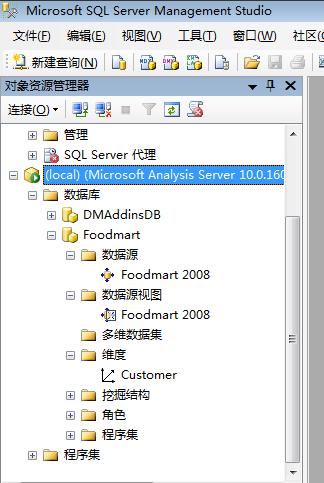
至此，customer维度的维度结构便设计完成了，现在点击“处理”按钮：



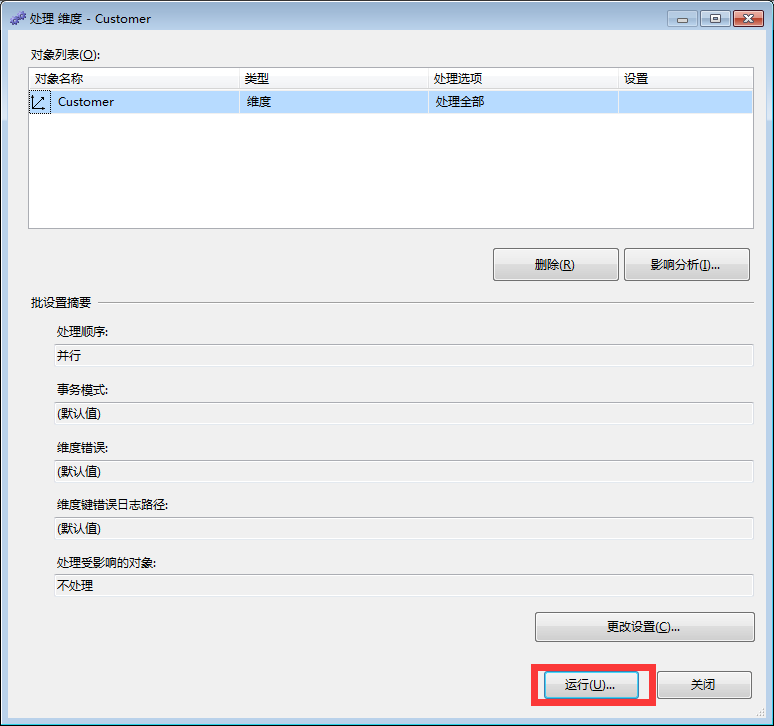


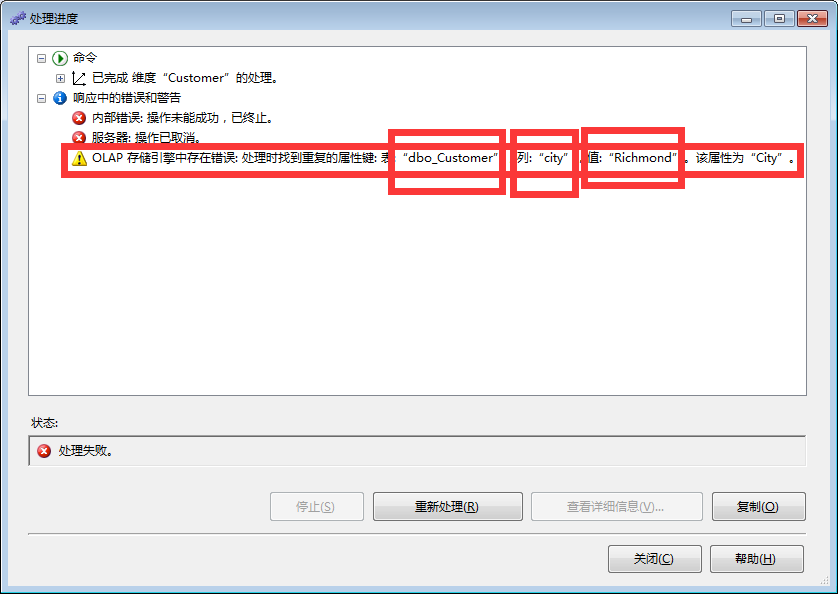


从上图右下角可以看到，该customer维表便已经部署到Microsoft Analysis Server服务器上了。打开SQL Server Management Studio ，连接上Microsoft Analysis Server（注意要刷新一下），便可以找到新部署的项目：

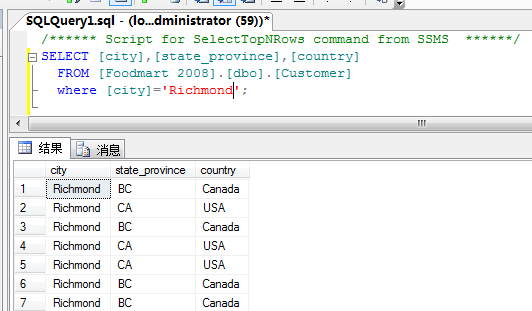


回到Microsoft Visual Studio 2008中，点击下图的“运行”按钮：

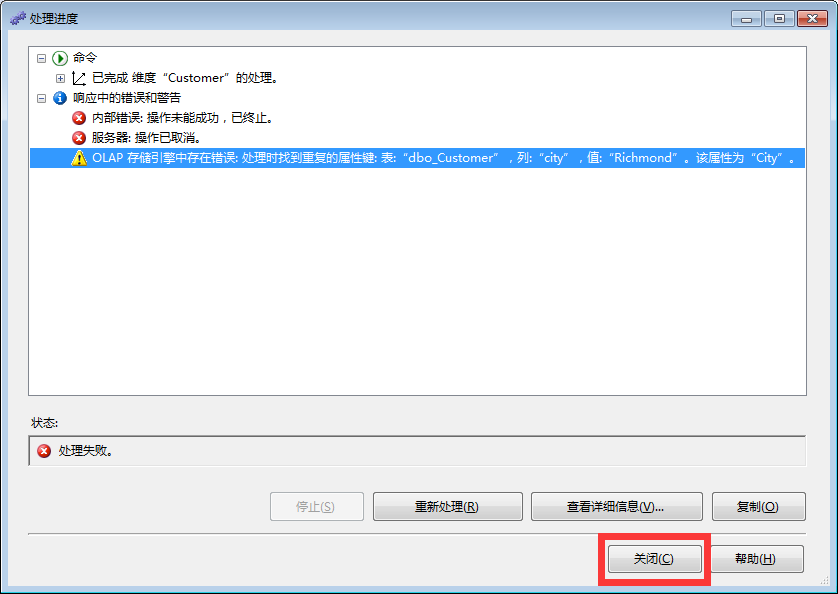


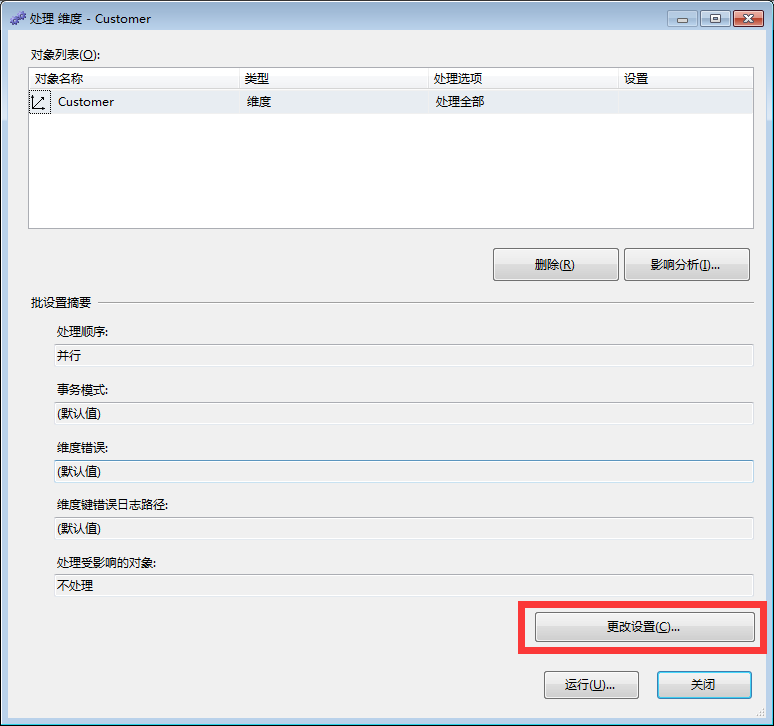


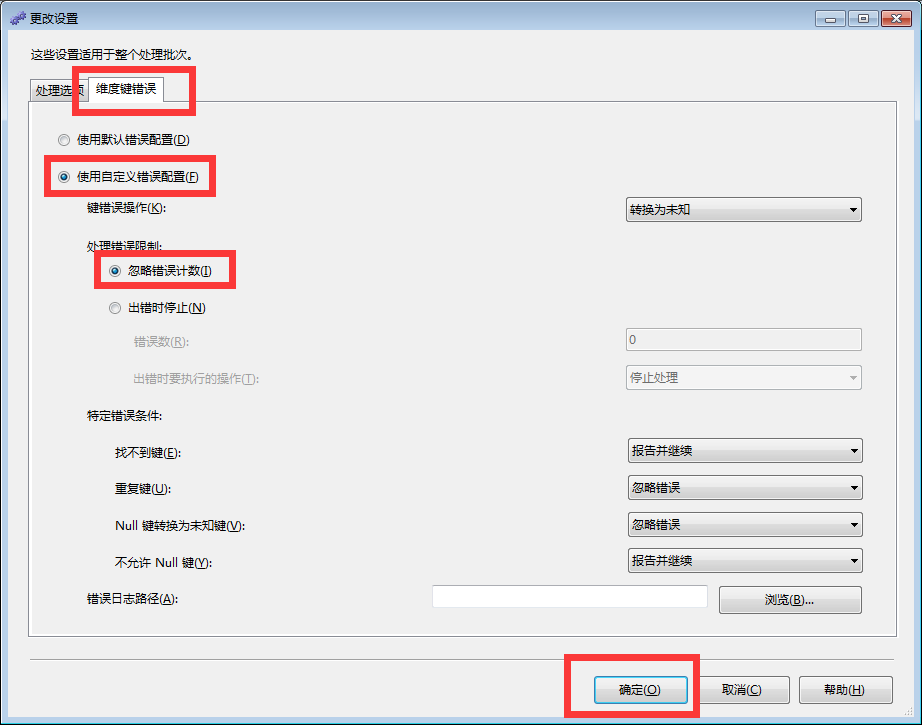
这时，出现了如上图所示的错误。为了进一步理解错误产生的原因，我们根据错误信息，在SQL Server Management Studio 中查找customer表属性为city，而属性值为Richmond的记录：

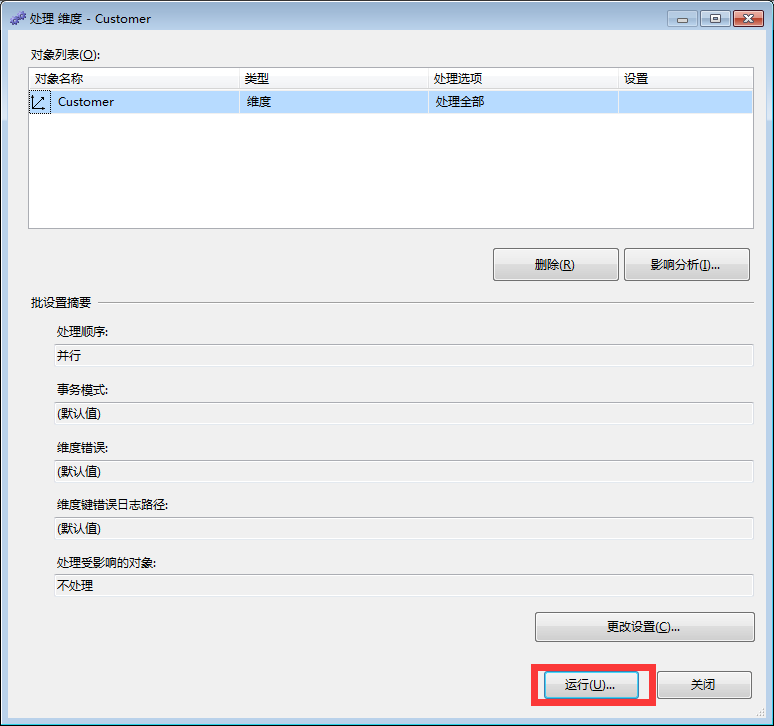


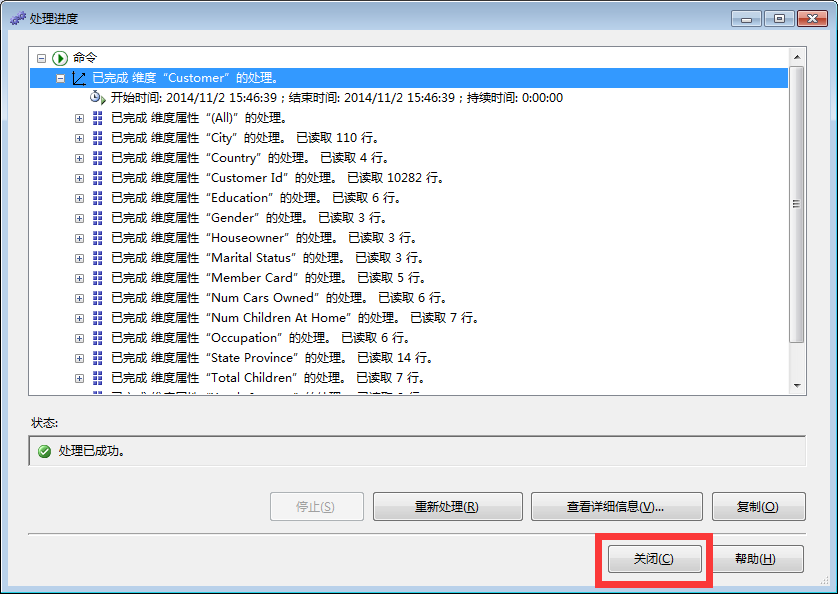
原来，“Canada”国家“BC”州有一个叫“Richmond”的城市，而“USA” 国家“CA”州也有一个叫“Richmond”的城市。这样，层次结构：country->state\_province->city有存在一个矛盾，“Richmond”到底是属于“Canada”国家“BC”州呢，还是属于“USA” 国家“CA”州呢？这里，我们仅仅做简单处理，就是把错误忽略掉：

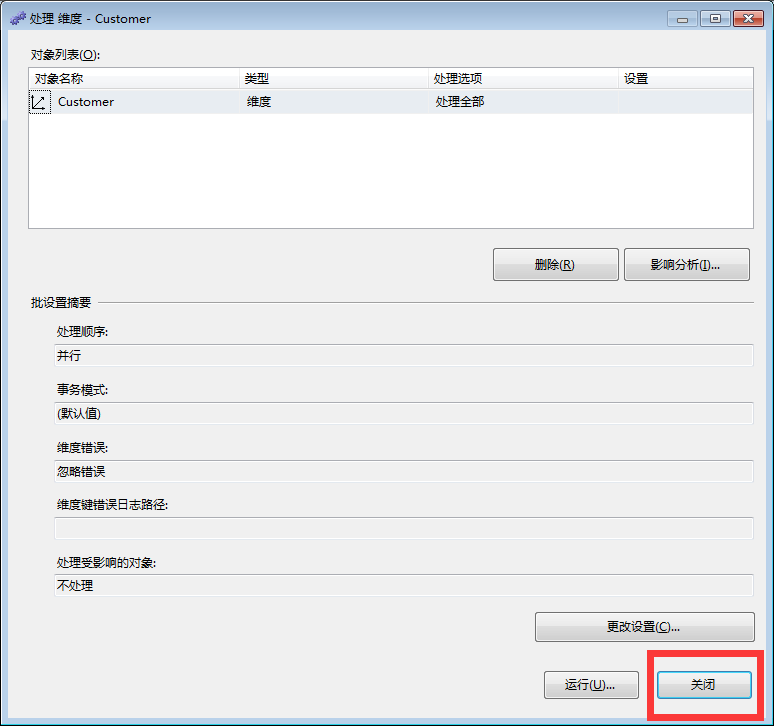


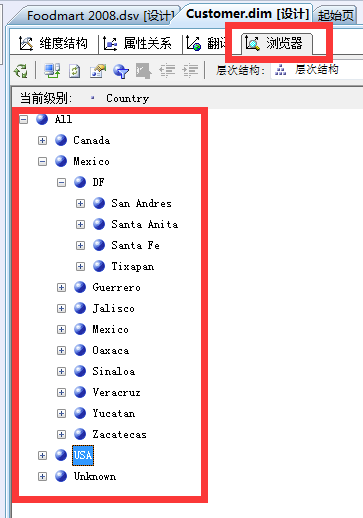




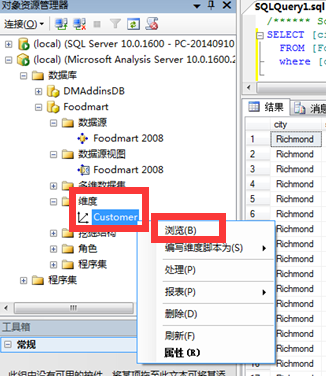




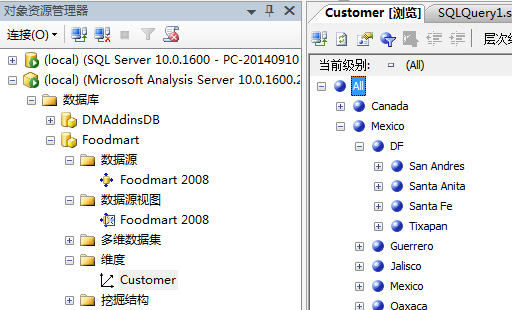




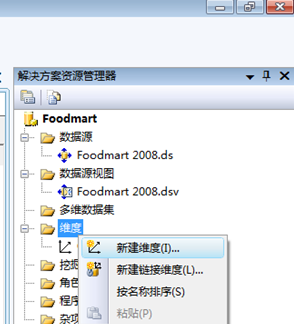
在SQL Server Management Studio 中刷新一下，右键点击“Customer”的“浏览”：

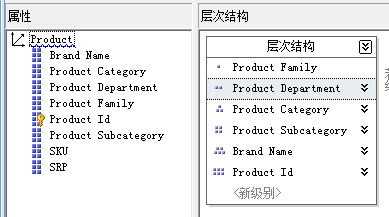


同样可以查看customer维表的分层数据：

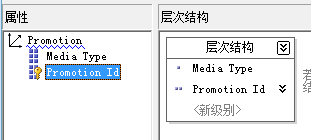


回到回到Microsoft Visual Studio 2008中，以上述方式继续新建其它（product，time\_by\_day，promotion和store）维表：





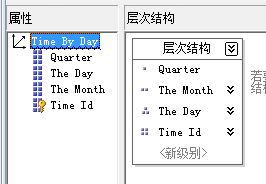
product维表：Product Family->Product Department->Product Category->Product Subcategory->Brand Name->Product Id



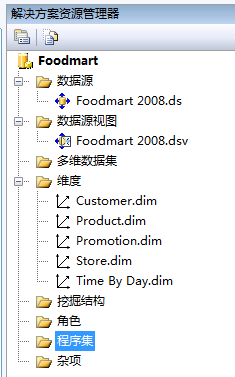
promotion维表：Media Type->Promotion Id



store维表：Store Country->Store State->Store City->Store Id



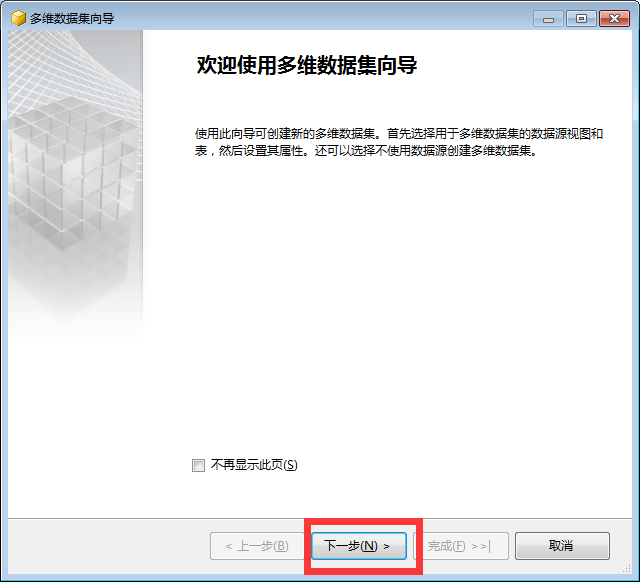
Time\_By\_Day维表：Quarter->The Month->The Day->Time Id

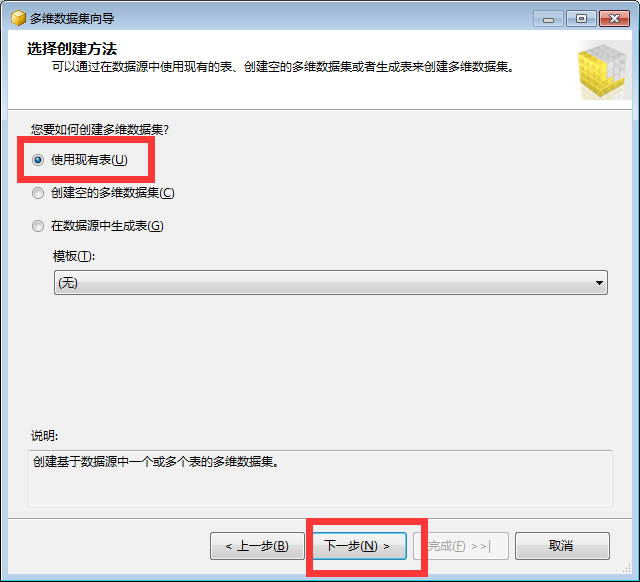


## 建立数据仓库

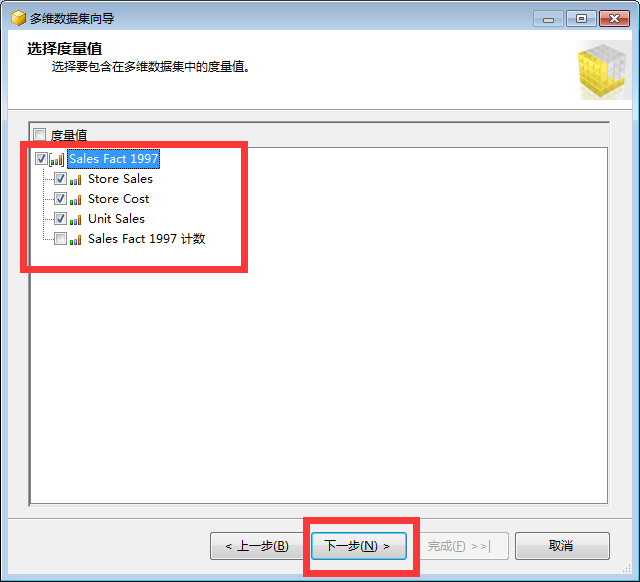
根据上述已经建好的维表，我们可以建立数据仓库。右键点击“多维数据集”：

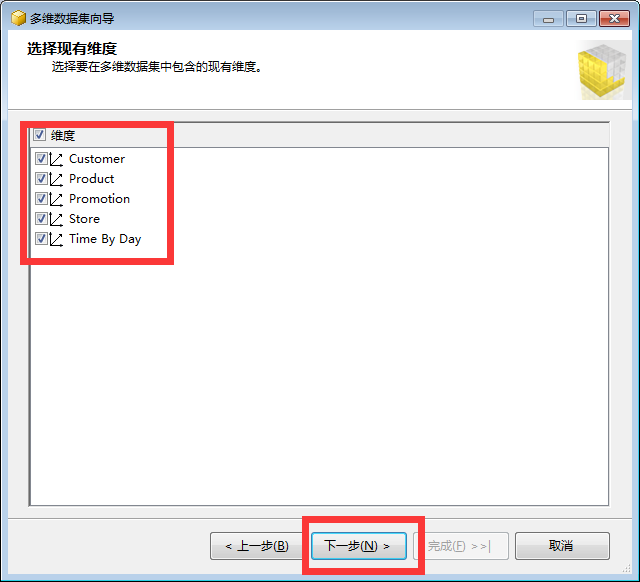


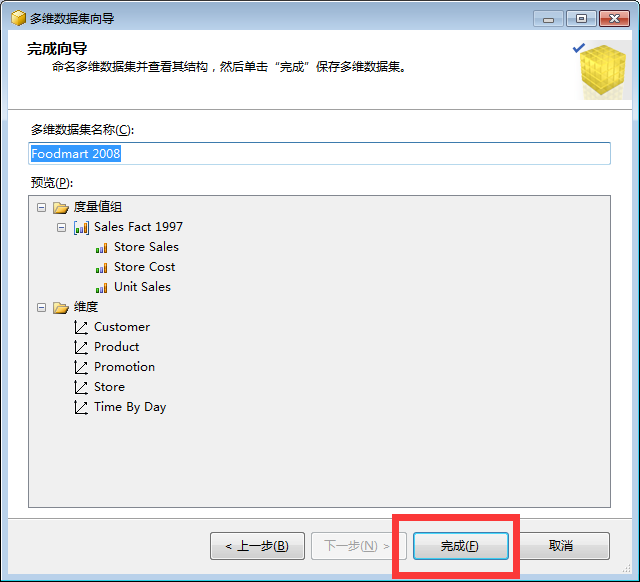




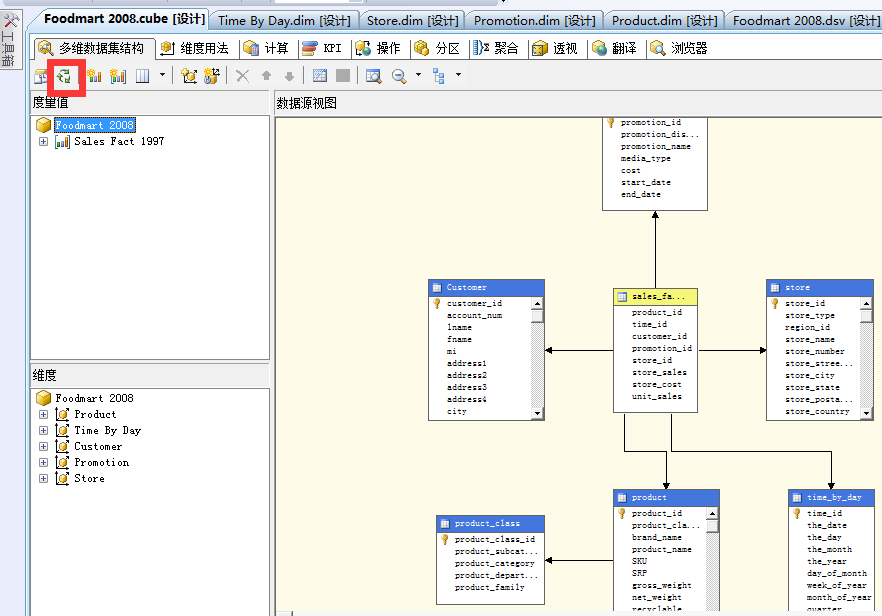


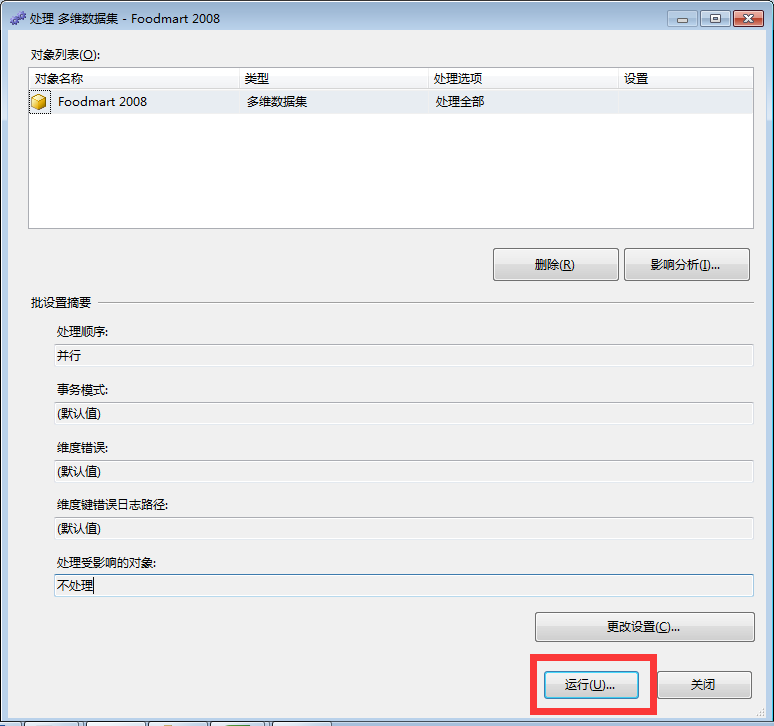


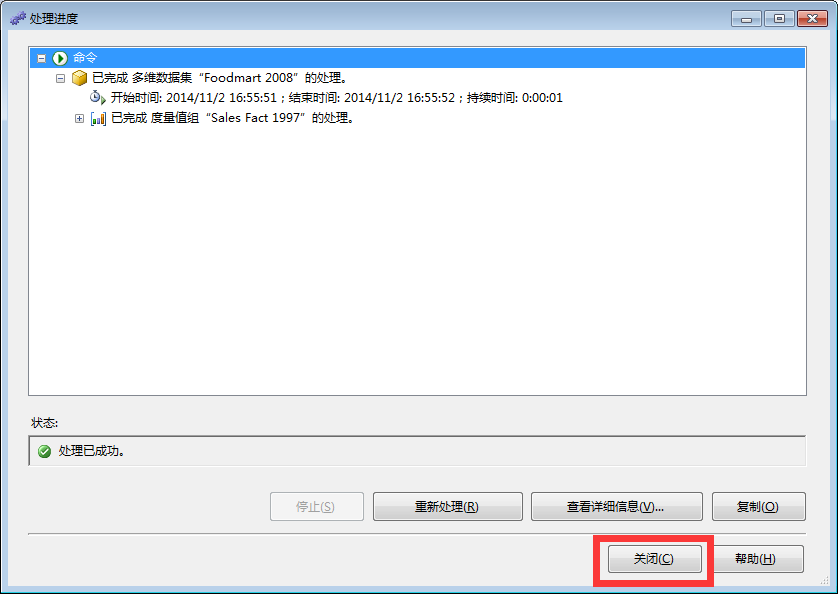


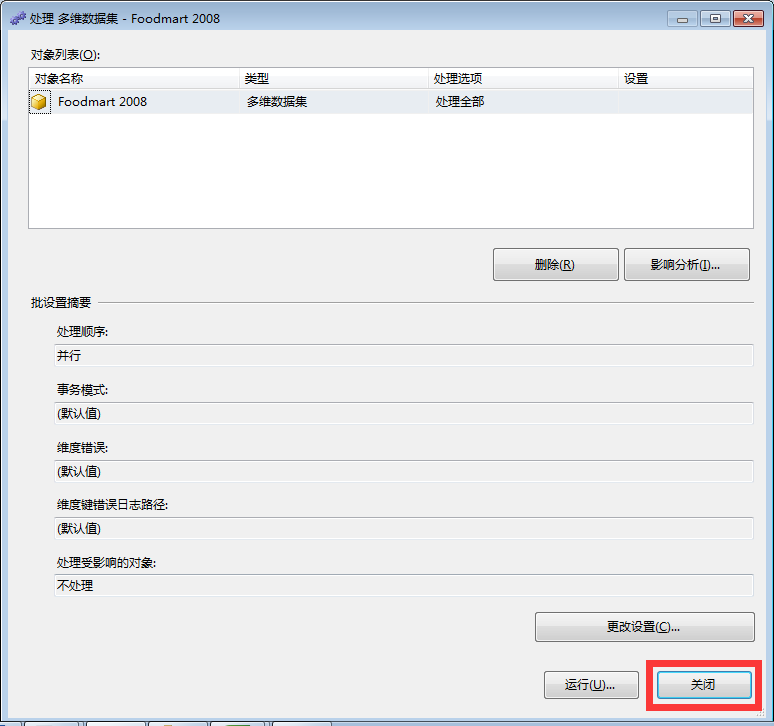


点击下方“处理”按钮，将多维数据集部署到Microsoft Analysis Server服务器上：

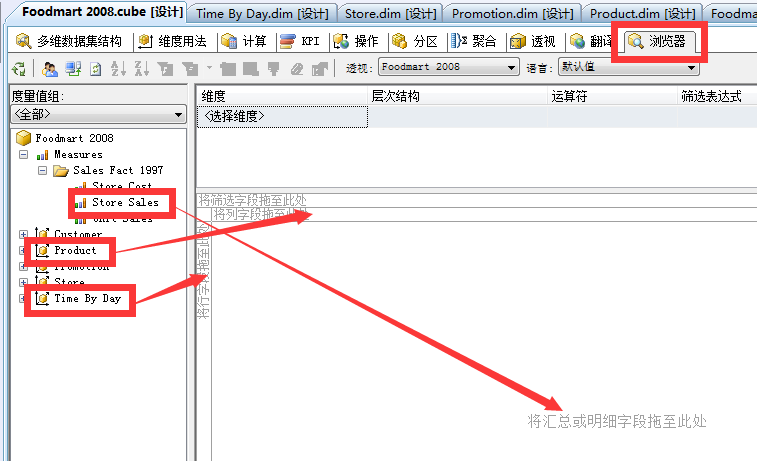


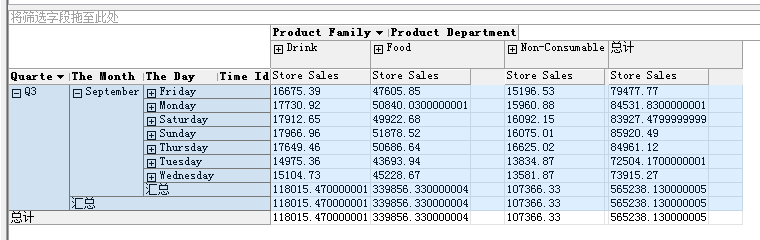






点击“浏览器”标签，并把“Store Sales”，“Product”和“Time By Day”拖曳到如下所示位置：

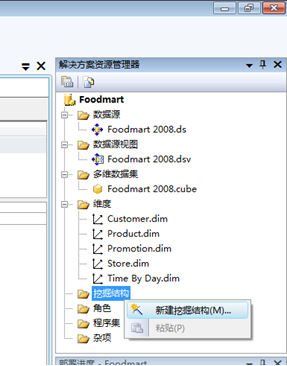


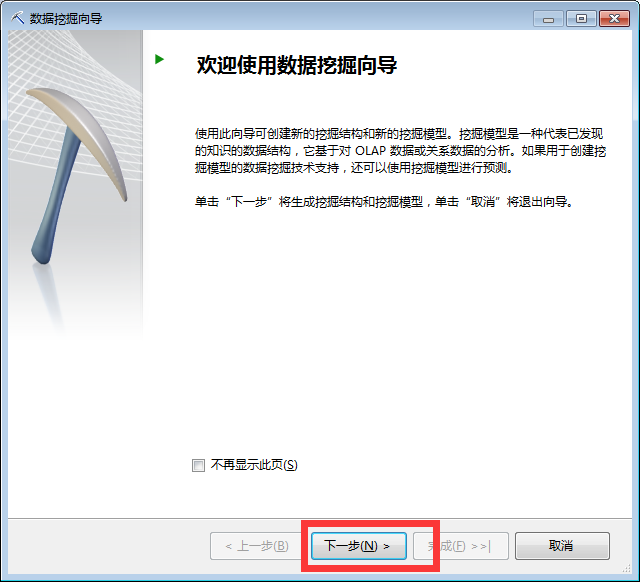


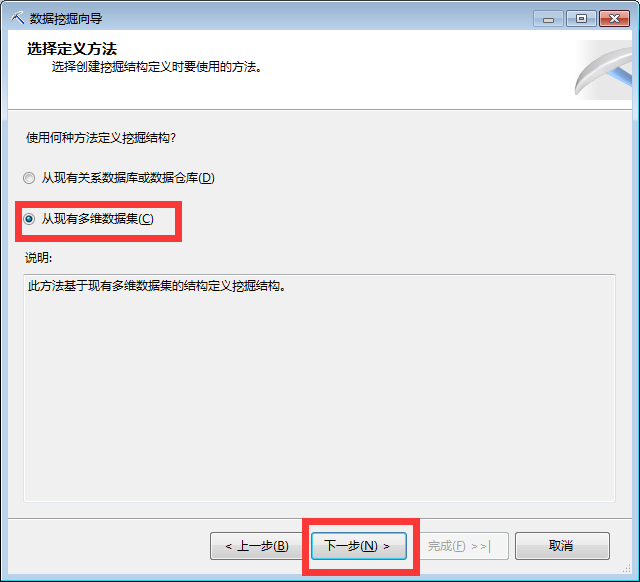
## 对数据仓库进行数据挖掘

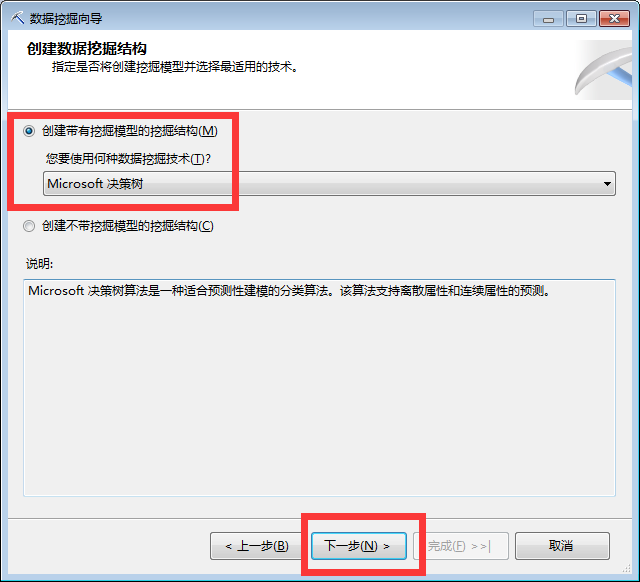
目前为止，数据仓库已经初步建立，我们可以尝试在数据仓库上进行数据挖掘。根据背景介绍，我们要挖掘（研究）办理不同会员卡的老客户有怎样的特征，进而根据新客户的信息，为新客户推荐不同的会员卡，给公司带来利润。

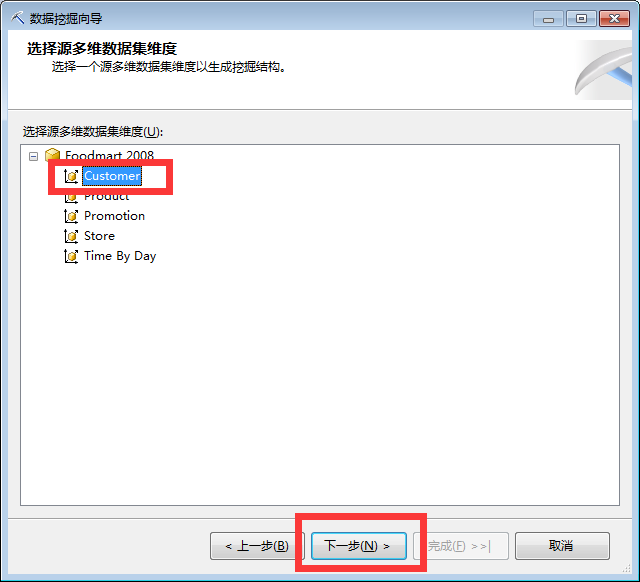
下面开始建立相应的数据挖掘模型。右键点击“挖掘结构”：



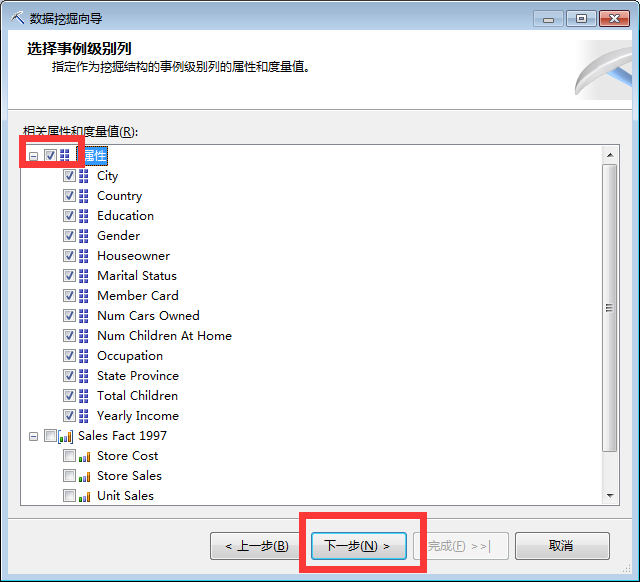




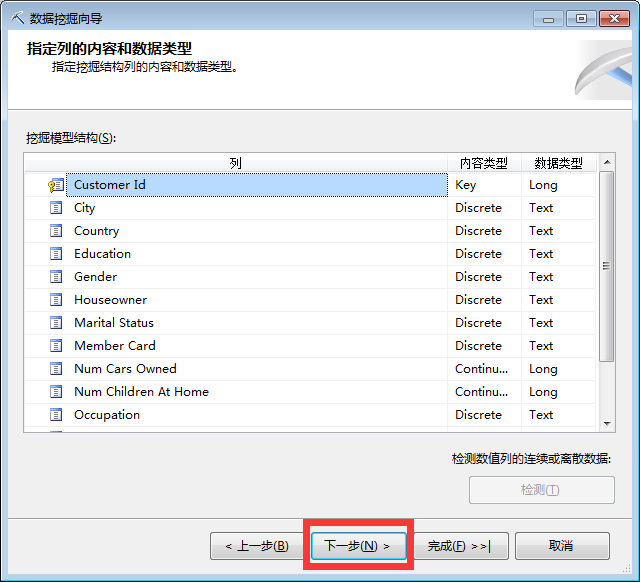


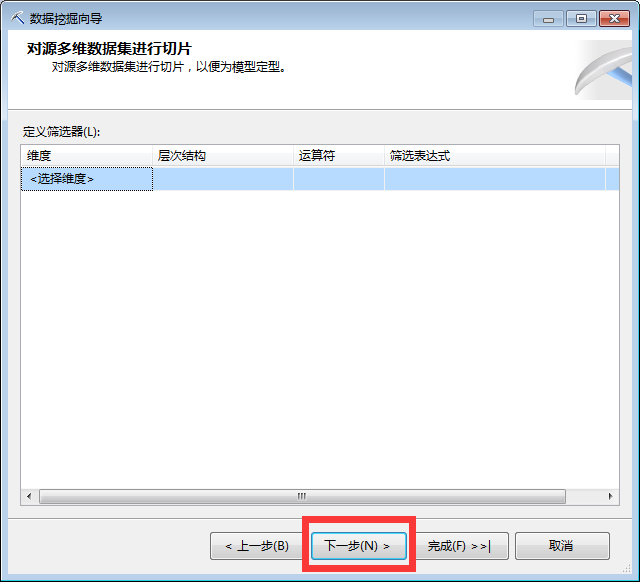


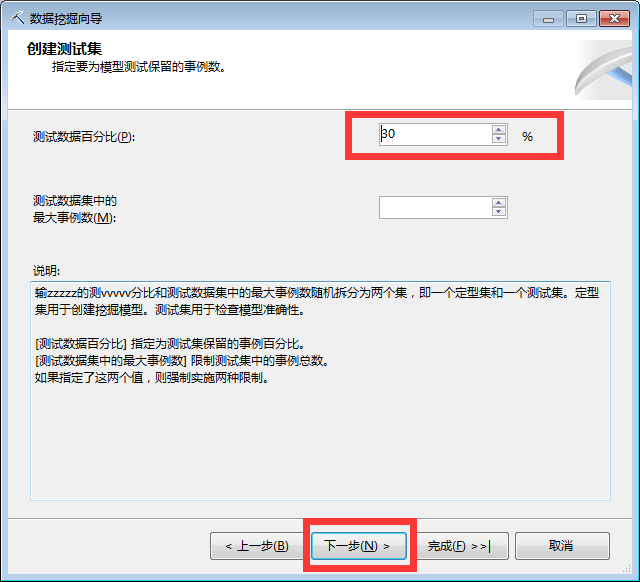


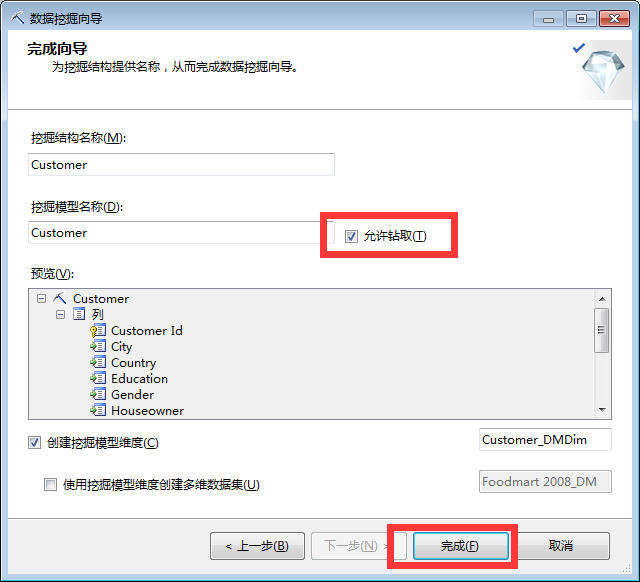


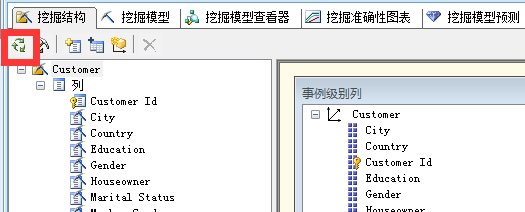


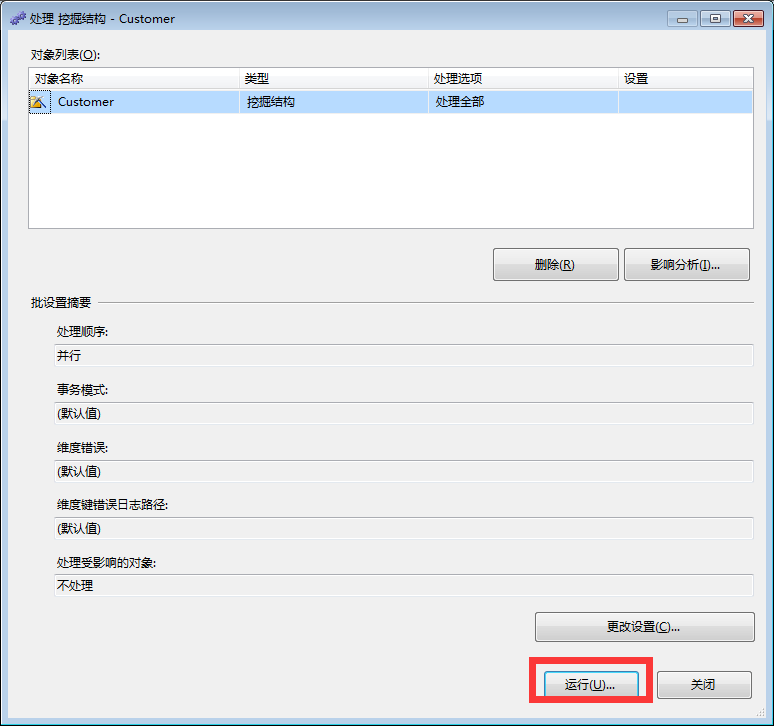


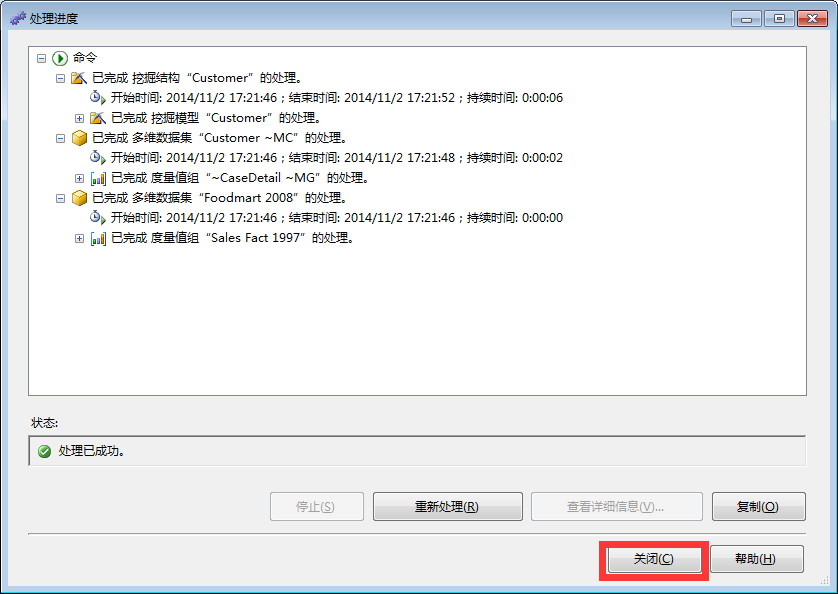


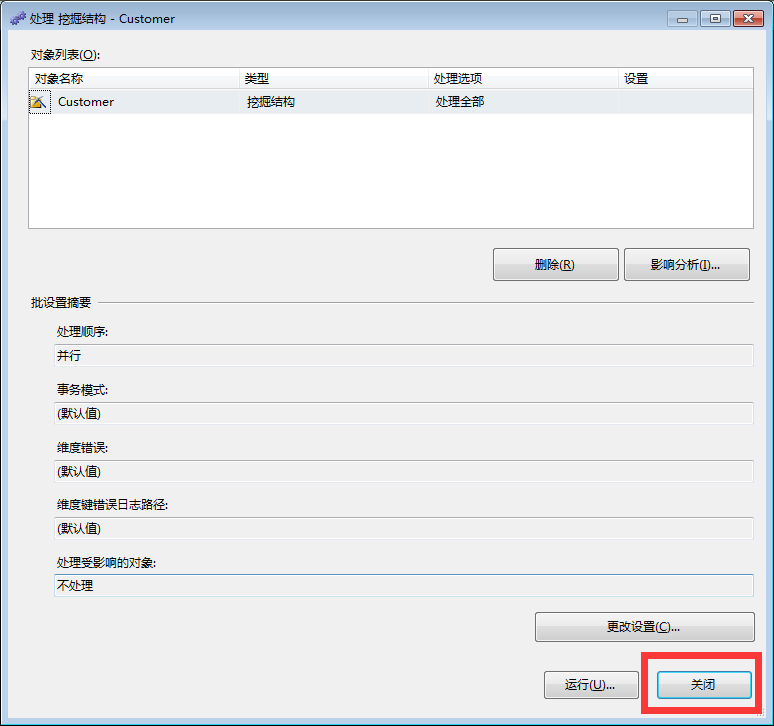




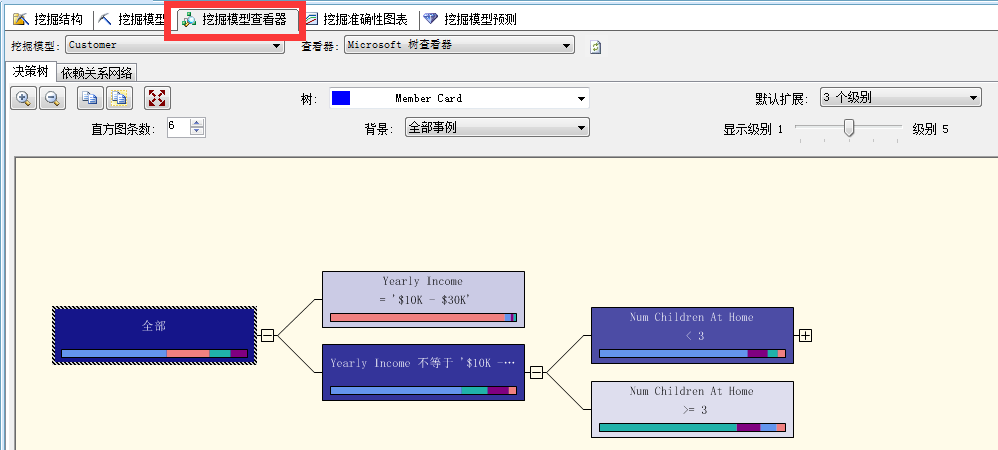




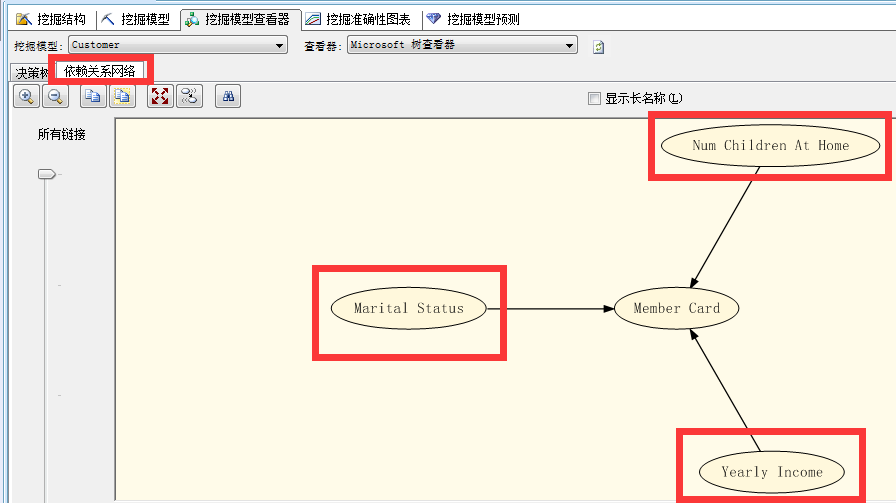




点击“挖掘模型查看器”查看数据挖掘模型的结果：

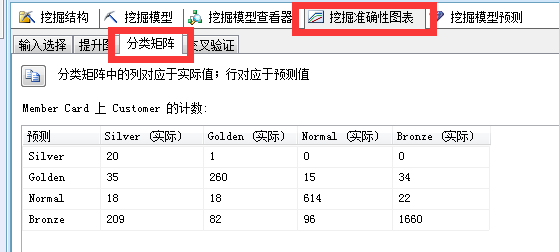


从上图可以发现，办理不同会员卡的老客户拥有的不同特征。

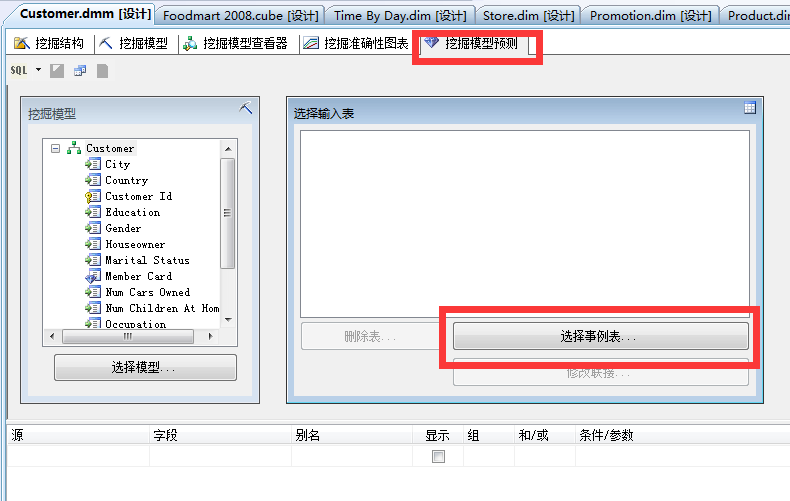


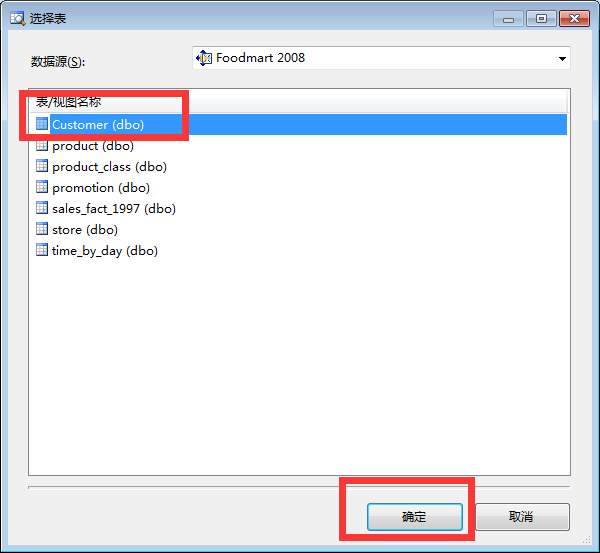
从上图“依赖关系网络”可以看到，与“Member Card”最相关的是“Num Children At Home”，“Marital Status”和“Yearly Income”属性。所以，如果想为新客户推荐适合的卡，应该最先考虑新客户的孩子个数（Num Children At Home），婚姻状况（Marital Status）和年薪（Yearly Income）。

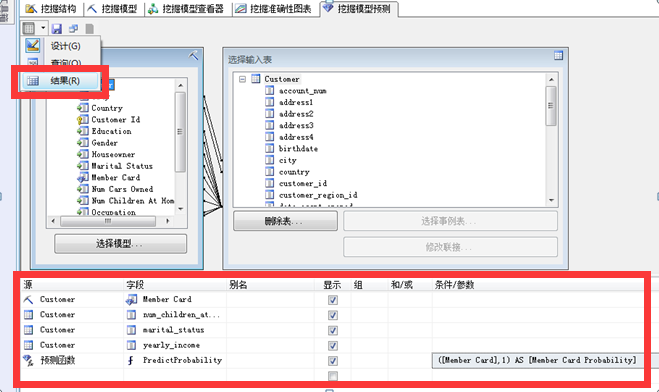
点击“挖掘准确性图表”查看数据挖掘模型的准确性：



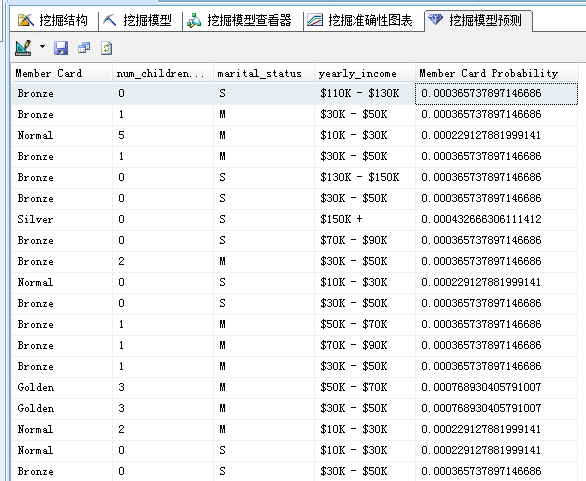
现在，我们可以尝试通过新客户的“Num Children At Home”，“Marital Status”和“Yearly Income”属性值来预测新客户需要哪种会员卡（Member Card）。点击“挖掘模型预测”标签：







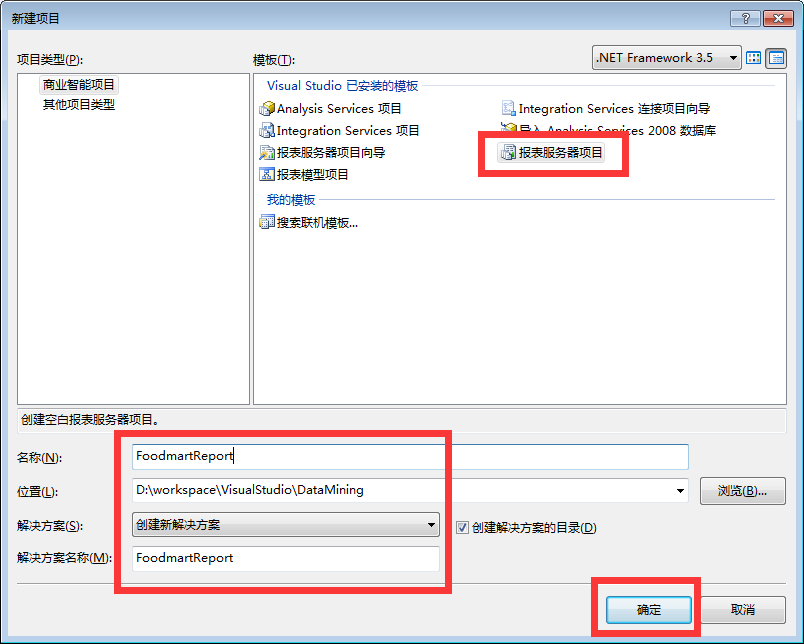
([Member Card],1) AS [Member Card Probability]

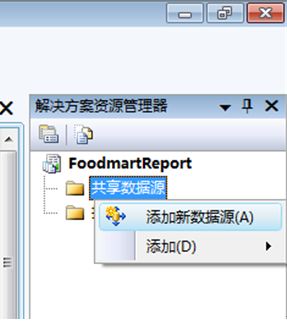


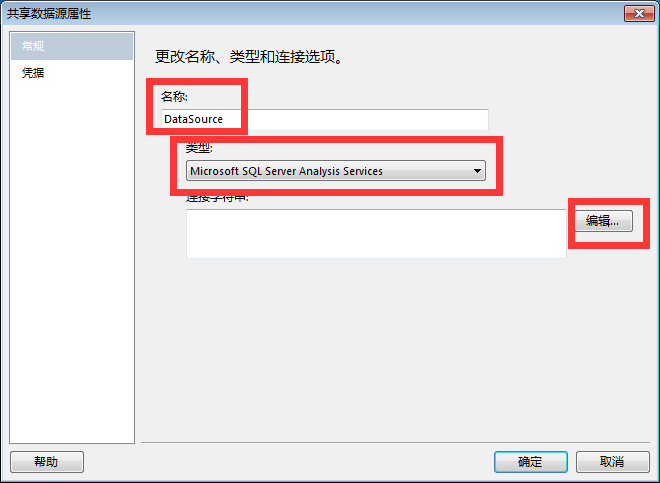
## 形成报表

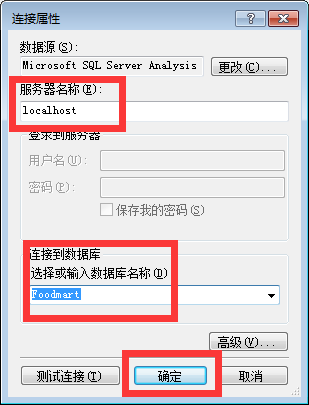
在Microsoft Visual Studio 2008新建一个项目：





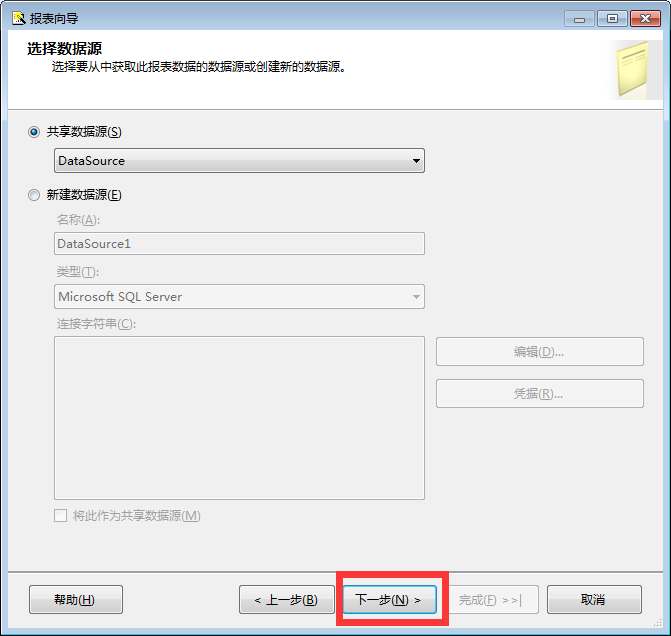


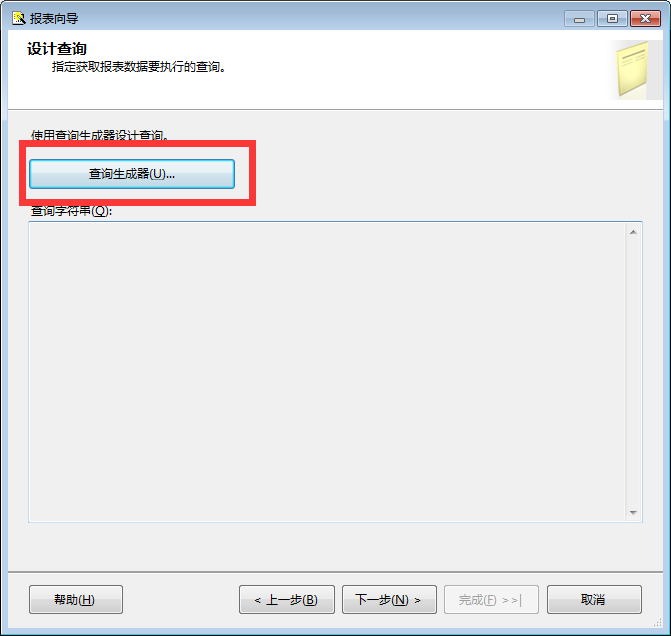


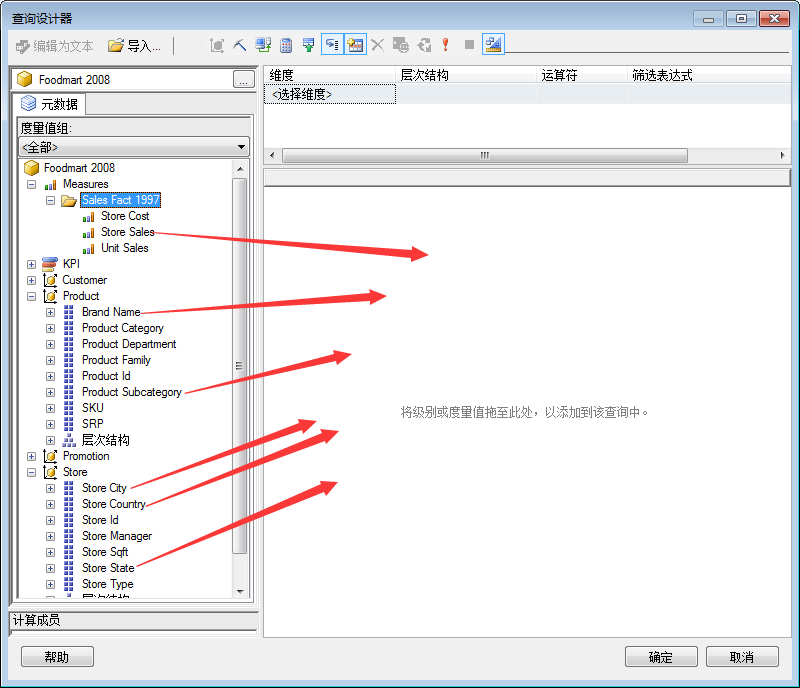


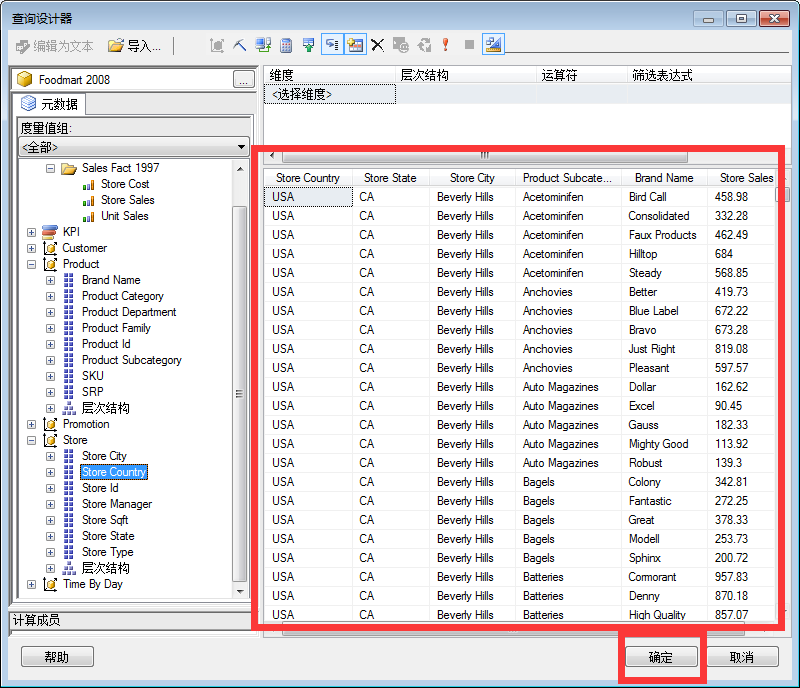


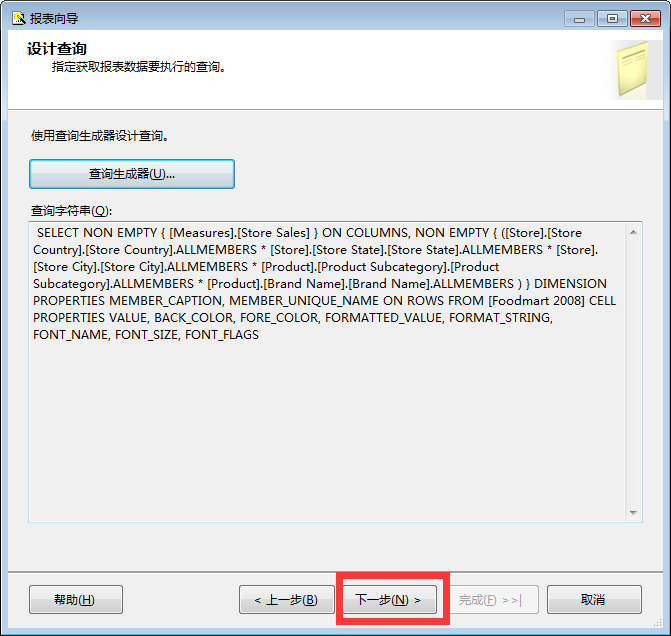




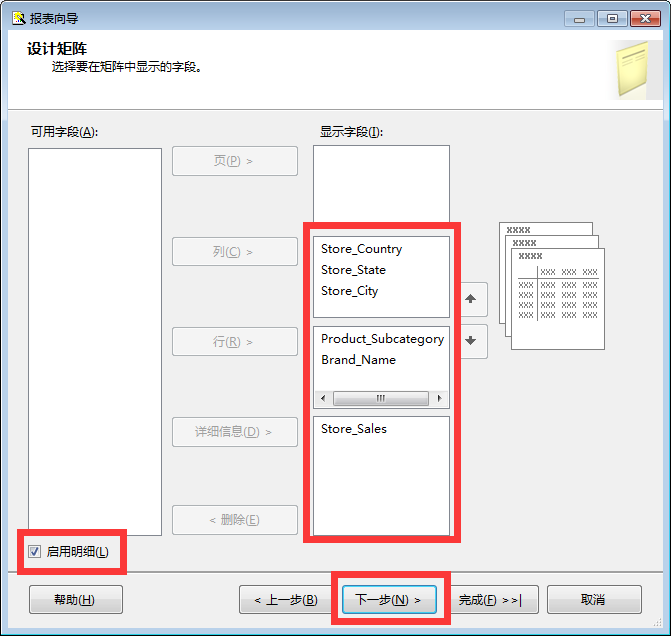


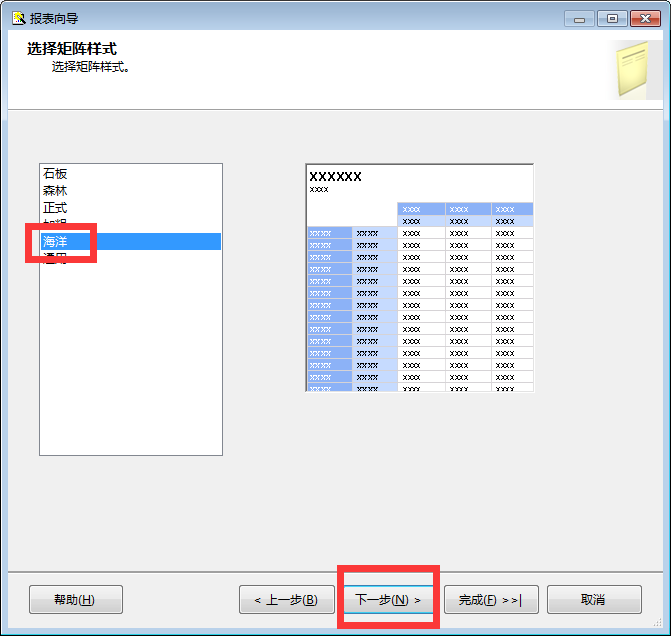


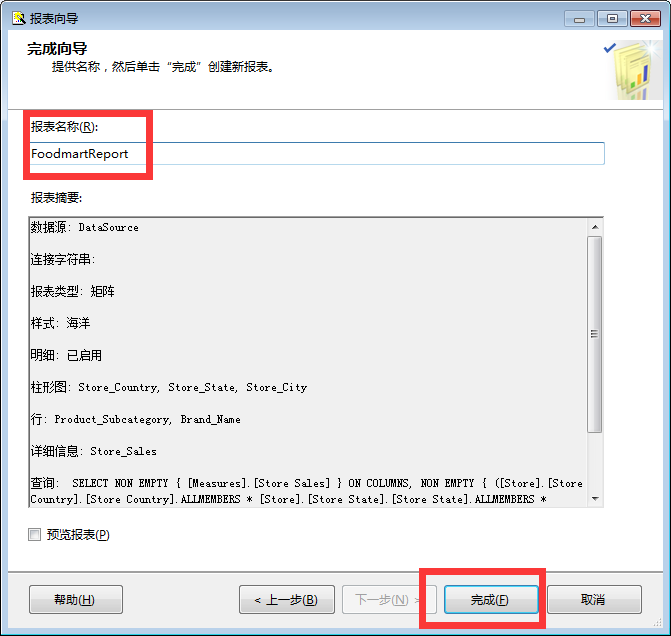


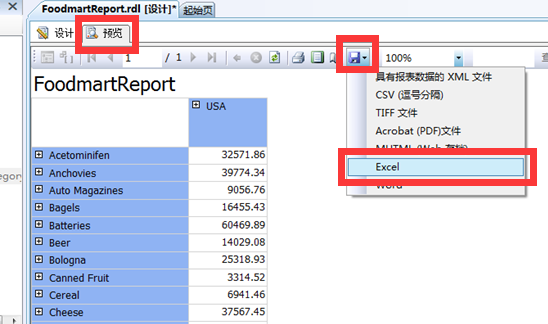


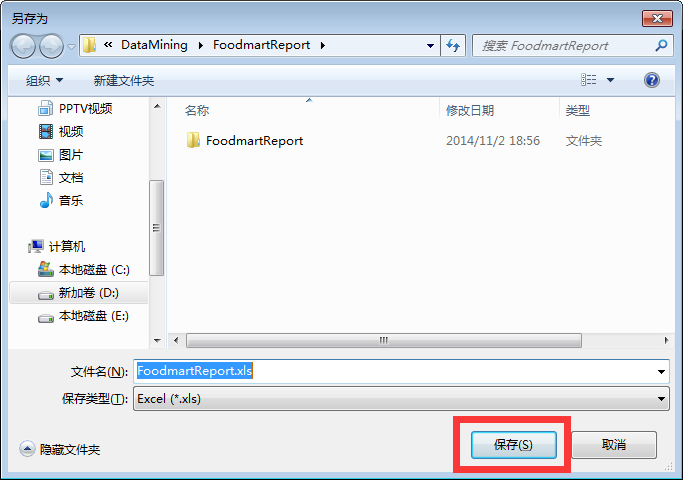






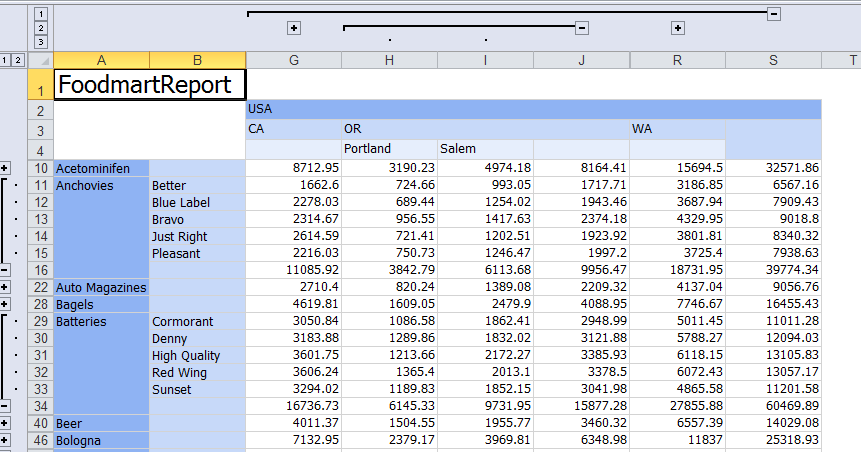






打开刚刚导出的报表文件：

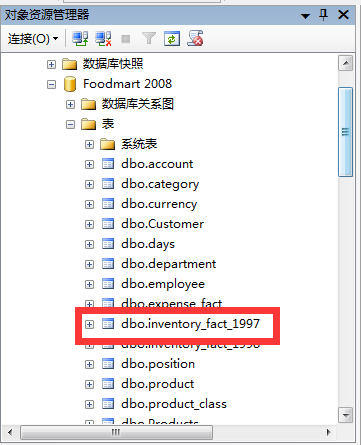




## 作业

使用Food Mart的销售数据建立数据仓库、进行OLAP、建立数据挖掘模型以及生成报表的一系列工作暂告一段落。

现在需要进一步推进项目，我们要使用Food Mart的库存数据进行上述一系列工作。



完成“需求分析”、“建立数据源视图”、“建立维表”、“建立数据仓库”、“建立数据挖掘模型”挖掘Food Mart需要的信息，以及形成报表。