

Datafactory 实际使用案例

李元

说明：Datafactory 的基础使用方法用户可以查看 Datafactory 使用指导书，本文档是在用户了解 datafactory 的基本使用方法的基础上，通过四个典型的例子来，来让用户对该软件有一个较深刻的理解。

案例说明：

案例一： 外键字段的产生

案例二： 外部数据（TXT 文档）的引入

案例三： 单据编号的产生方法

案例四： 主从表关系数据的建立

1 外键字段的产生

说明：外键是数据库关系中最普通也是使用最平凡的约束方式，我们可以通过二种方式来实现外键约束。一种是 Insert value from a data table,另一种是 Insert value from an sql database

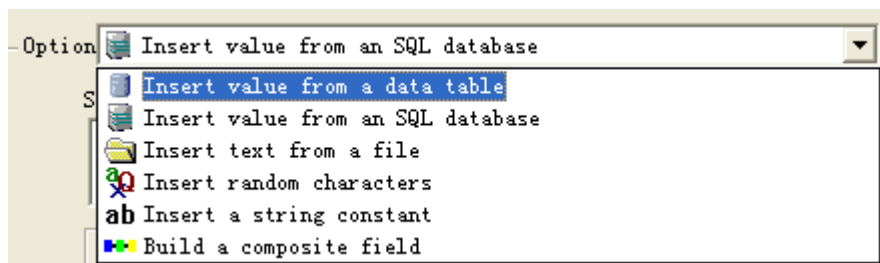


图 1.0

1) Insert value from an sql database 方式

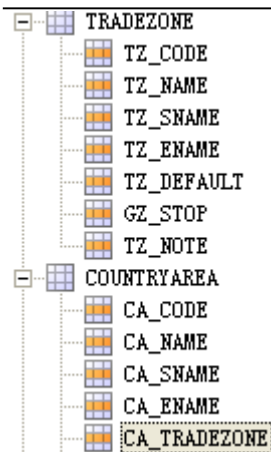


图 1.1

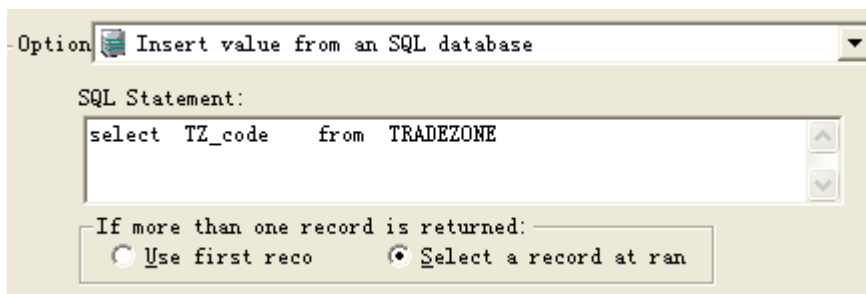


图 1.2

上图中 Countryarea 表中 CA_tradezone 和 Tradezone 表中 TZ_code 之间是外键关系

我们把 CA_tradezone 字段产生的方式先设置成为写 SQL 语句的方式，然后通过

Select TZ_code from Tradezone 这个简单的 SQL 语句使 Ca_tradezone 得到 TZ_code 的值，并且用户可以在 SQL 返回的结果中，筛选所需要的内容 if more than one recode is returned（如果返回的条件超过一行），

Use first record

(只选择第一行作为当前字段值)

Select a record at random

(随机选择一个值作为当前字段值)

注意:

当然这种数据产生方式其实不仅仅可以产生外键关系的数据，因为 S Q L 语句是比较灵活的，读者可以自己常识写一些复杂的 SQL 语句来多尝试一下

2) Insert value from a data table 方式

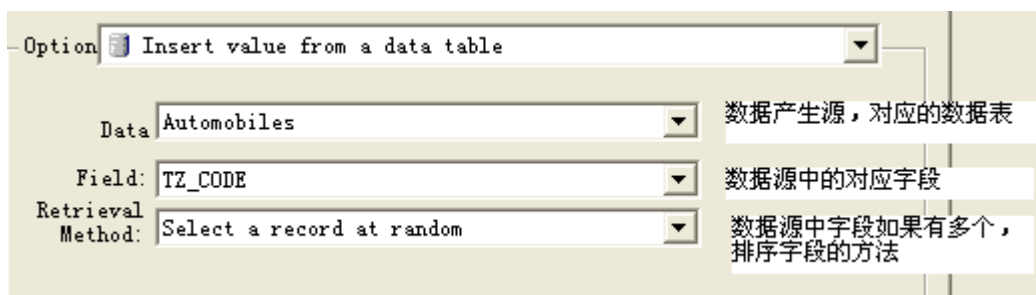


图 1.3

如图选择 Insert value from a data table 方式，分别出现以下设置内容

Data (产生数据的数据源)

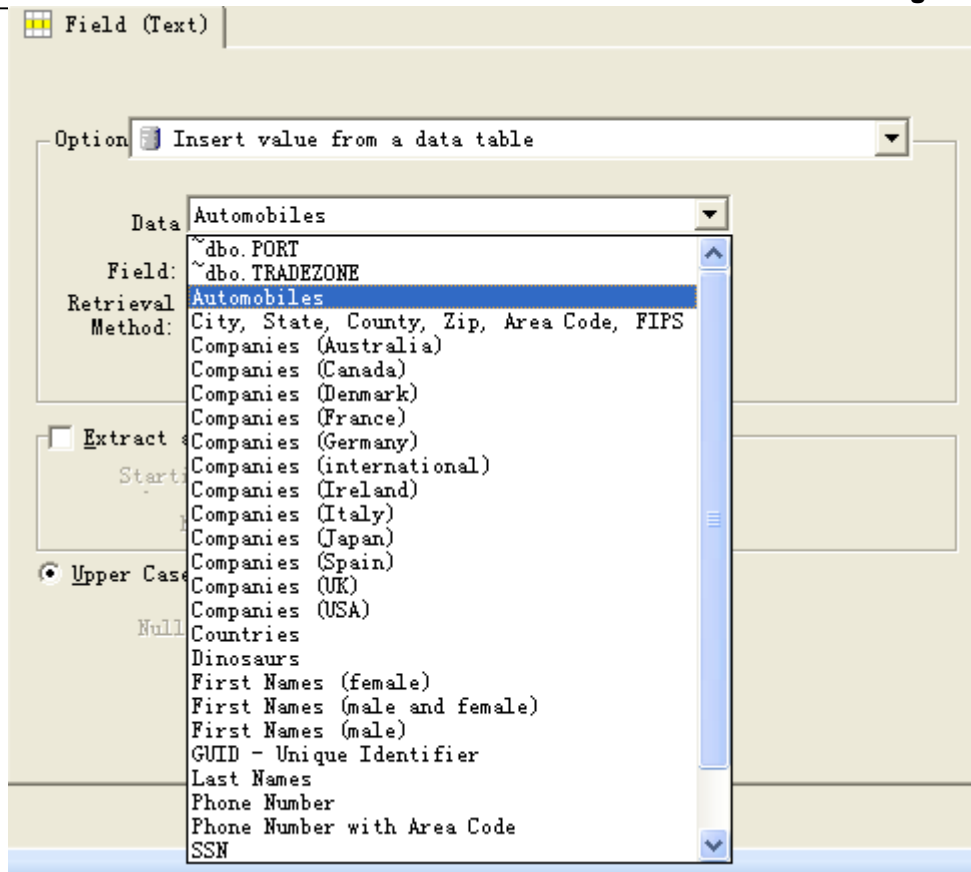


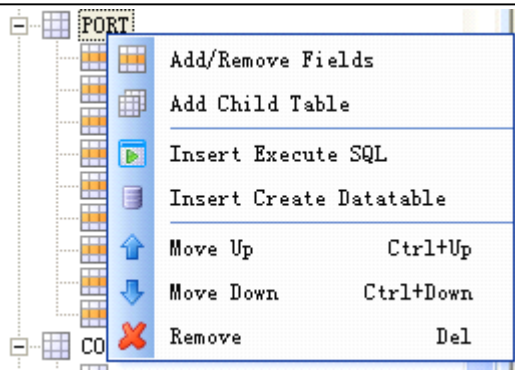
图 1.4



图 1.5

说明: Data 下拉框默认列出了系统中提供的数据源, 包括各个国家的公司名称, 男英文名称, 女英文名称, 电话号码, 区号—电话号码, 等等, 特别说明的是, 用户可以通过勾选图 1.5 中的 Create temporary data table.. (创建一个临时的数据表), 来使在 datafactory 中数据源中有对应的表, 比如选择数据表 TRADEZONE, 然后在图 1.5 中勾选 Create temporary data table, 这样当数据表 TRADEZONE 下面任意使用 Insert value from a data table 方式时, Data 中就会有相应的 TRADEZONE 选项,

注意: datafactory 中左边的数据表是有上下关系的, 用户可以通过右击表, 通过下图中的 Move Up 和 Move Down 来调整表的上下顺序



Fild

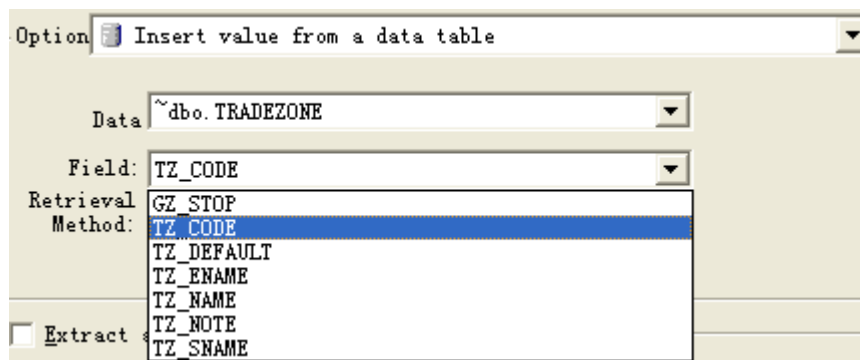


图 1.6

说明: 如图 1.6 中, 列出了当前选择的数据源中的可选字段,

Retrieval Method (取数方法)

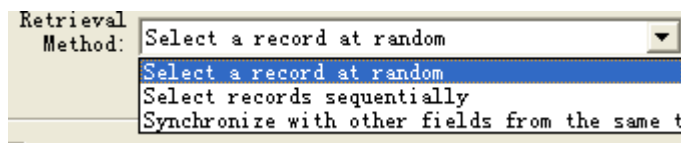


图 1.7

说明: 如图 1.7 中, Select a record at random (随机选择一行)

Select records sequentially (按顺序选择一行)

Synchronize with other fields from the same table (相同的字段从同一个表中)

2 外部数据 (TXT 文档) 的引入

说明: 在测试数据库时, 很多情况要求根据真实的数据来做必要的测试, 这就需要我们可以通过外部自定义的文档来产生数据表中的数据, 比如我现在有一个 TXT 文档, 里面是我想要录入到数据表“港口”的值,

上海, 大连, 香港, 福建, 青岛

首先创建一个 TXT 文件，名字任意，格式如下
port_name:varchar(100) , port_sname:varchar(100)

上海,shanghai
大连,dalian
香港,xianggang
.....

注意： 字段名称 ： 字段类型 (字段类型长度)

引入 TXT 文件的方法:

右击项目名称(比如说这里的 liyuan)，如图 1.8 所示,选择 Insert Create Datatable 进入图 1.9 界面,点选 Import data from file，点击下一步，点击 Browse 选择相应的 TXT 文件,点击下一步，然后为该引入的 TXT 自定义一个名字，下一步，最后完成引入 TXT 文件的操作.

引入成功后，在左边的树型结构中会有如图 1.8.1 中的数据源，图中"国内区域"，"国家地区"，"贸易区"，"港口" 是引入 TXT 后的结果.

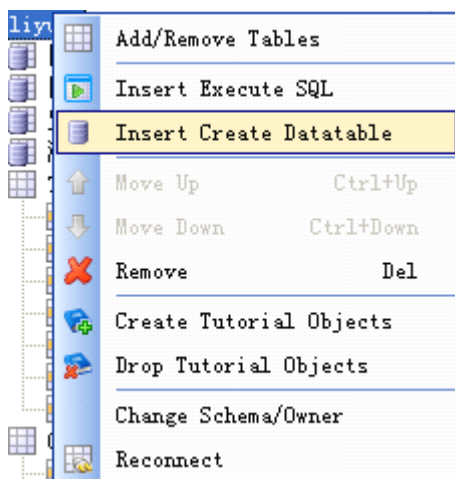


图 1.8



图 1.8.1

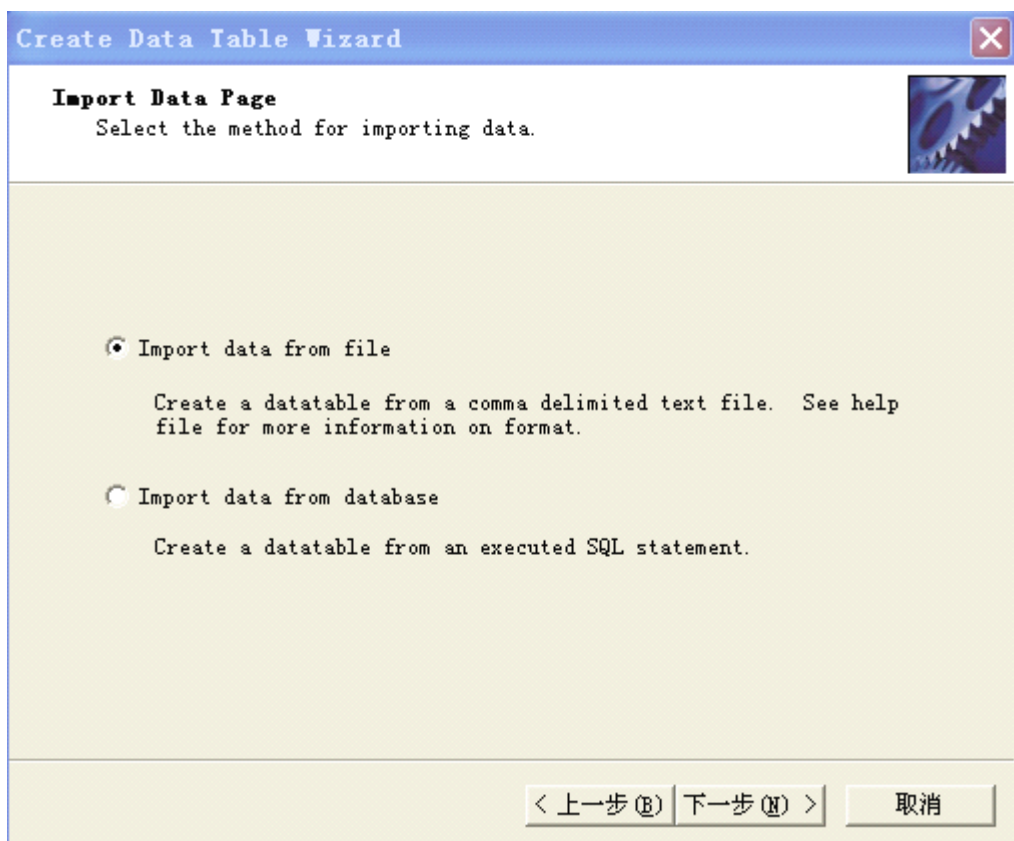


图 1.9

使用引入后的 T X T 字段内容:

如图 2.0 是根据上面的 TXT 引入到 datafactory 中后,产生的数据源.,Data 中显示的是 TXT 文件的名称,Field 里显示的是 TXT 文件中第一行的字段名,当然这里的名字可以完全自定义,

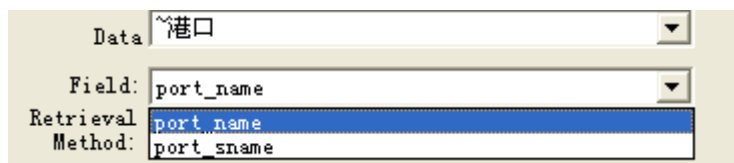


图 2.0

当然用户也可以把几个表的字段内容同时放在一个 TXT 文件里,做一次引入,但是这样不方便以后的数据源维护,不建议使用.

比较遗憾的是,Datafactory 导入的方式只支持 TXT 文件.

图 1.9 中的 Import data from database 是根据一段 SQL 语句来得到数据源,原理跟图 1.0 中使用 Insert value from an sql database 差不多,读者可以自己尝试一下.

3 订单编号的产生

说明：在ERP的测试中，对单据的测试是最重要的，那么我们如何来产生符合实际业务的数据那，比如说单据编号，以及之后要讲解的单据主从关系表的产生。

比如说，系统中的单据编号产生的方式是 TP + 当前日期 + 一个自动增加的流水号
TP20080826001,TP20080826002,.....

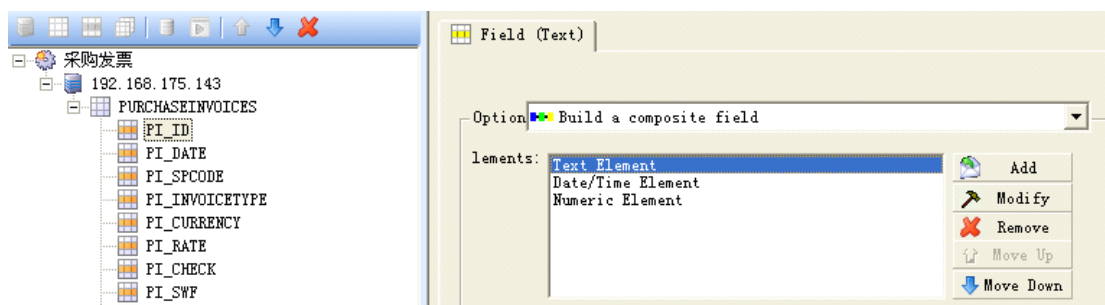


图 2.1

如图 2.1 中的 PI_ID 字段，我们对它做一个单据编号产生方式的设置，

选择 Option 里的 Build a composite field(创建一个合成字段)，当然这个选择只有字段类型是 Text 时才存在，原因读者自己琢磨。

然后我们来对这个合成字段做格式设置，首先单据编号中的 TP 是个固定字符，通过点击图 2.1 中的 add 按钮，得到图 2.2 界面，因为我们要产生 TP 这个固定的字符，所以我们选择 Text 类型，点击 OK,进入如图 2.3 界面，在 Option 中选择对应的 Insert a string constant ,然后在 Constant Text Expression 中输入自定义的字符，比如这里的 TP ,完成后保存设置，以上是对单据编号 TP 字符自动产生方式的设置过程，之后的时间，流水号的产生方式，在这里不再详细解说，原理是一样的，只不过在 Element Type 中选择不同的类型，Date / Time 或 Numeric ，相关截图,图 2.4 是 Data /Time 的设置界面，图 2.5 是 Numeric 的设置界面。

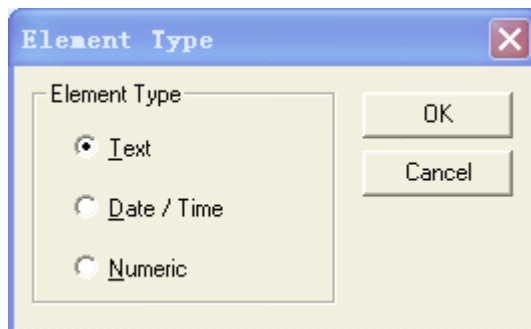


图 2.2

Element 'Text Element' Properties

Field (Text)

Name: Text Element

Option: **ab** Insert a string constant

Constant Text Expression:
TP

☐ Extract a portion of t

Starting character position of: 1

Number of characters to: 1

Null: 0

☐ Unique

☐ Key

确定 取消

图 2.3

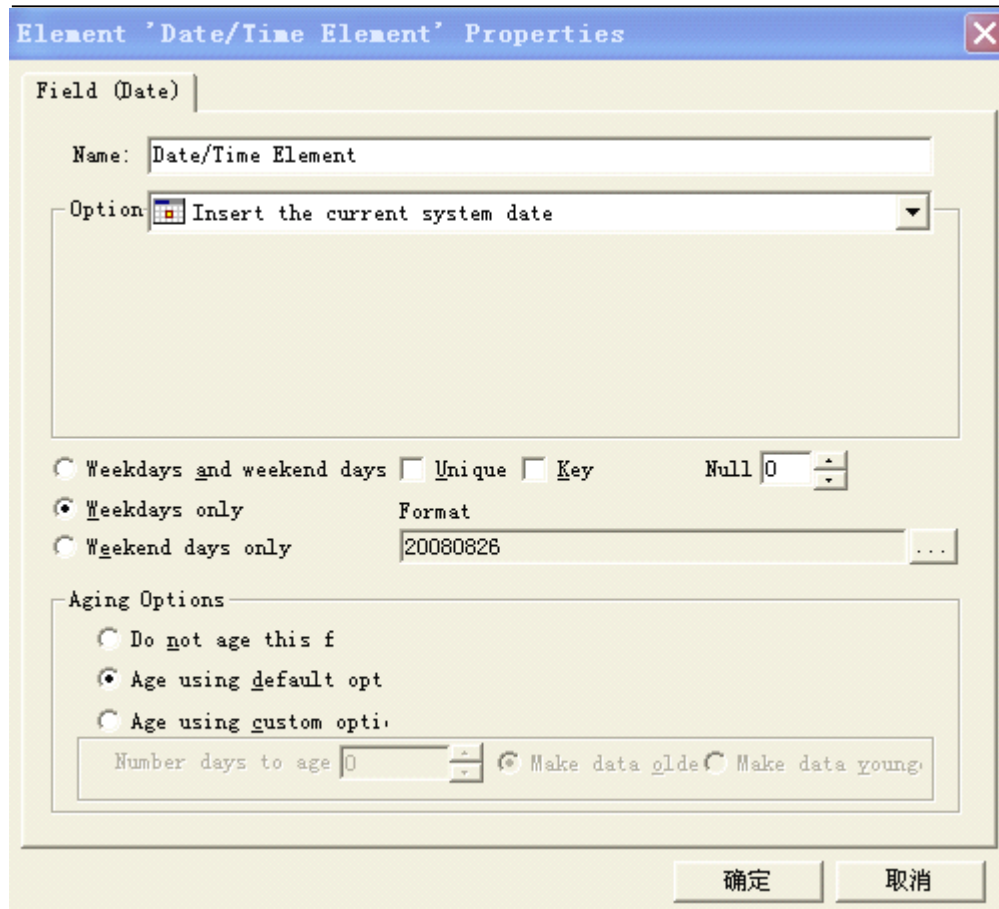


图 2.4

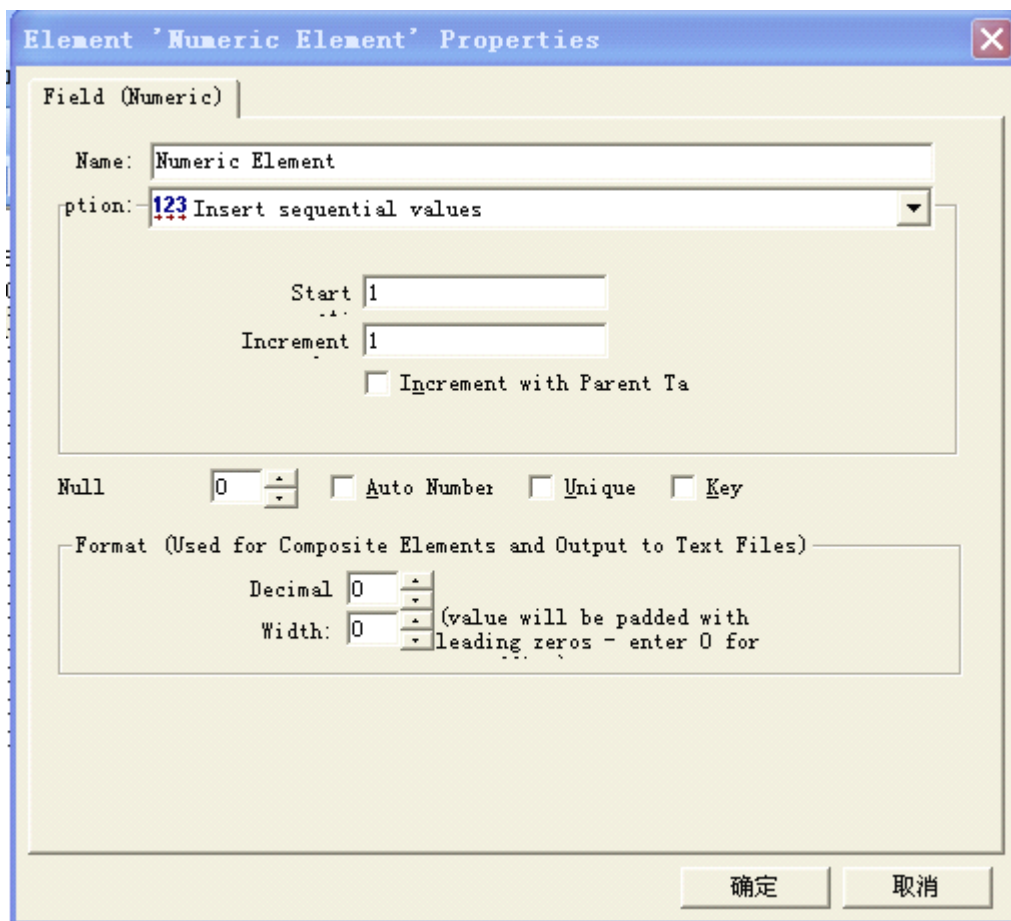


图 2.5

4 主从关系表的创建

说明:

主从表是 ERP 系统中最常见的一种表关系结构，这里先简单介绍一下主从表的结构
比如说有一张单据, TPxxxx001, 单据里有 3 个商品，分别是商品 1, 商品 2, 商品 3.

TPxxxx001	数量	单价	金额
商品 1	10	1.58	15.8
商品 2	20	2.1	42
商品 3	30	3.1	9.3

主表里的数据结构:

CO_ID			
TCxxxx001	单据属性 1	单据属性 2	单据属性 3

从表里的数据结构:

COD_ID	明细编号	数量	单价	...
TCxxxx001	1	10	1.58	
TCxxxx001	2	20	2.1	

TCxxxx001	3	30	3.1	
TCxxxx001	N	N	N	...

说明：主表中是以单据编号为主键的，从表中是以单据编号+明细编号的组合为主键。

我们来分析一下这二个表中三个字段之间的关系，首先要确定的是，主从表之间应该是先有主表记录，再有从表对应的内容，这点是非常重要的，比如上面的单据表结构，首先我们要确定的是主表中的单据编号，然后把该单据编号复制到从表的 COD_ID 中，当然这个时候从表的该单句编号对应的明细可能是 1-99999 的任意数值，对应的实际业务也就是说，一张单据里面可以包括 99999 条明细记录，并且从表里的明细 ID 编号不能重复对于一个 COD_ID 来说。（这是数据库设计方面的知识，读者自己理解）

下面我们来看在 Datafactory 中怎样实现这个逻辑：

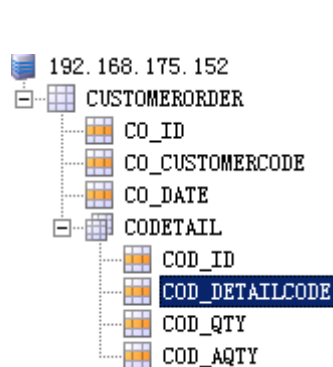


图 2.6

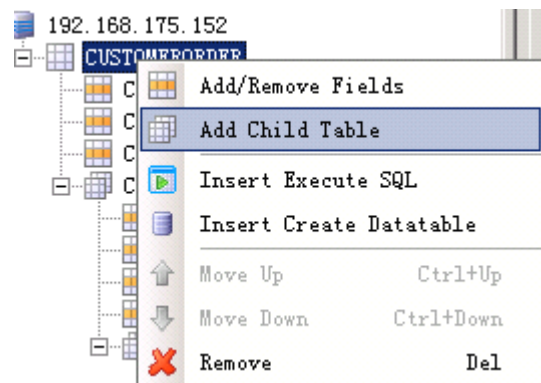


图 2.7

如图 2.6 中，CUSTOMERORDER 表是主表，CODETAIL 是对应的明细表，读者可以通过图 2.7 和图 2.8 的方法来产生主从表关系，

如图 2.8 中，Records 1 to 5 的意思是指，主表中的一条主键 ID 允许对应从表里的最多 5 条明细 ID，（读者注意，对应明细 ID 的数量是 1-5，也就是说产生的单据明细可以是 5 以内的任何值）。Parent Table 中选择主表中的主键字段，child Table 中选择从表中对应的关联字段，比如图中的 CO_ID 和 COD_ID。它们是这二个主从表之间的父子关联字段。当然，图 2.8 中也可能是对多个主从字段之间的关联，这个留给读者自由发挥。

由上面的分析我们知道，COD_DETAILCODE 这个是明细表里的明细编号，对于一个单据编号来说，这个明细编号应该是唯一的，所以我们可以把 COD_DETALCODE 这个字段设置成自动增加的类型，根据上面的设置，我们可以得到下面的二张单据表

主表

TCxxxx001
TCxxxx002
.....

TCxxxx00N

从表

TCxxxx001 1

TCxxxx001	2
.....	
TCxxxx001	N
TCxxxx002	N+1
TCxxxx002	N+2

Add Child Table Wizard

Parent Child Relationship Page
Establish parent child relationship here.

Records
Enter a range for the number of records to insert into the child table for each record added to the parent table.
Records: 1 to 5

Relationships

Child Table: STYLE_SIZE	Parent Table: CUSTOMERORDER
COD_ID	CO_ID

Note: Double click in any row to display the field list for the corresponding table

Remove Selection Remove All

< 上一步(B) 下一步(N) > 取消

图 2.8

关于主从表关系的思考:

1 明细编号 COD_ID 的循环问题

细心的读者可能会注意到，上面明细 id 产生的方式跟实际数据有所不同，实际的数据应该是每张不同的单据明细是各自增加的，当换一个单据编号后，明细 id 应该重置成 1 开始自增，但问题是我们如何知道重表中的当前 COD_ID 是新的那，对于 datafactory 来说，如何知道那。是不是可以用写 SQL 加判断的方式那，这个留给大家讨论，（因为我 SQL 不熟悉☺）

2 主从表数量一致的问题

数据主从表结构的人都知道，一般对于一张单据来说，主表里会有一个总数量，明细表里又存储了各个商品的数量，实际的业务数据总数量肯定会=对应明细数量之和，但是我们在 datafactory 里就很难实现这个数据产生逻辑，（个人觉得不可能），这个也留给读者考虑，欢迎大家把自己对该软件的使用经验拿出来一起交流.....

最后感谢 QUST 公司给我们做了这么好的软件.