|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 色泽 |  |  |  |  |  |  |  |  |
|  | 1号 | 2号 | 3号 | 4号 | 5号 | 6号 | AVG | SD | CV |
| 1号 | 9.0000 | 8.0000 | 8.0000 | 9.0000 | 9.5000 | 9.0000 | 8.7500 | 0.6124 | 6.9985 |
| 2号 | 7.0000 | 8.0000 | 6.0000 | 7.0000 | 6.5000 | 8.5000 | 7.1667 | 0.9309 | 12.9900 |
| 3号 | 6.0000 | 7.0000 | 9.0000 | 8.0000 | 9.0000 | 8.5000 | 7.9167 | 1.2007 | 15.1667 |
| 4号 | 5.0000 | 6.0000 | 9.0000 | 6.0000 | 8.0000 | 8.3000 | 7.0500 | 1.5922 | 22.5839 |
| 5号 | 6.0000 | 5.0000 | 7.0000 | 6.0000 | 6.0000 | 8.0000 | 6.3333 | 1.0328 | 16.3073 |

如例文件（今麦郎品评.xls）

我理解的是：列1-5号为5种产品id，行1-6号为6位品检员打分数值id，avg,sd,cv则是6位品检员打分的平均值，标准差，标准差率。色泽等为该产品的属性。

因此分析该文件，我将其拆分为两个表：

产品表（产品id，属性，数值对应id）

|  |  |  |
| --- | --- | --- |
| productId | Attr1 | Attr2 |
| 0 | DataId1 | DataId2 |

数值表（数值对应id，n位品检员分别对应的打分，avg,sd,cv）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DataId | N1 | N2 | Avg | Sd | Cv |
| 0 |  |  |  |  |  |

由此，我使用python将该excel里的数值取出存在sql server里。以此文件为例，代码如：

但是这样的表格不能解决多个域的问题，灵活性不够。后来我发现了sql server中横纵表的概念。为了灵活性，方便添加种类，我认为也可将数据库设计为纵表，如此表格式如下：

产品表（产品id，属性，数值对应id）

|  |  |  |
| --- | --- | --- |
| productId | Attr | DataId |
| 0 | Attr1 | DataId1 |
| 0 | Attr2 | DataId2 |

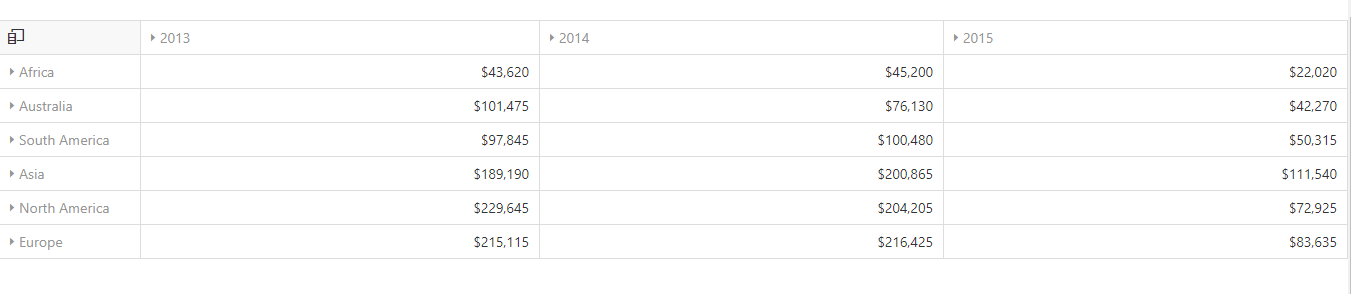
如果使用python将该excel里的数值取出存在sql server里。以此文件为例，代码如：

|  |
| --- |
|  |

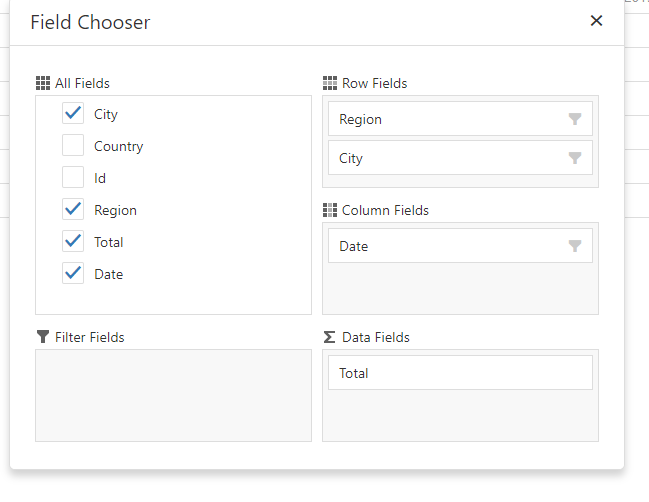
横纵表转换后查询方式如：

|  |
| --- |
|  |

同时，我在查数据库增删域时发现了某个前端组：devextreme。用它可以非常便捷的进行多维度数据的处理。出于兴趣我也做了一个demo,效果如：



点击红圈处可出现以下组件：

（其中数据域可拖动调整）

代码如（文件路径为：pivotGrid/）：



打开index.html可预览，因为组件所需的文件都是通过http引用的，所以速度可能比较慢。但我认为这个组件非常方便漂亮，在分类的处理上简单明了。