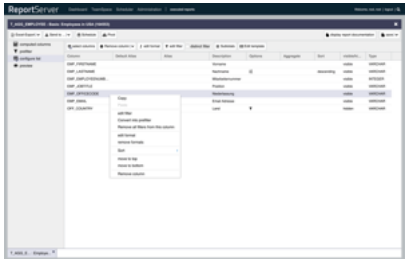


动态列表

动态列表

ReportServer的动态列表是选择的方法，如果您需要快速，用户特定的报告/分析，或者重点是对所选数据的后续处理。

动态列表可以处理几乎任何报告要求，从简单的过滤条件，排序，分组和子总计到更高级的功能，如分析功能，一阶逻辑或预处理方法。



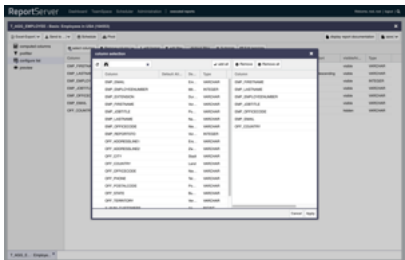
一旦指定，您可以将您的报告存储为所谓的变体，并通过TeamSpace与同事分享。

在本章中，我们将讨论动态列表的基本功能。

如果双击动态列表打开报表视图，您可以配置打开的列表。每个动态列表都基于管理员提供的源数据表。这张表通常很大，可以容易地包含几百列和数百万条目。为了从这么大量的数据中挖掘信息，能够选择合适的子集是至关重要的。为此，ReportServer提供了各种功能强大但直观可操作的工具。

6.1。列选择

每个动态列表的第一步是选择一组列。您将在方面列表配置的工具栏中找到列选择。打开对话框基于常用选择（请参阅第一步（../First-steps/））章节对话框，并显示左侧的现有列和右侧选定的列。



所选列被加载到方面列表配置中，可以进一步调整它们。对于每个选定的列，网格中有一个条目显示以下信息：

柱	列的技术名称。
默认	name由管理员配置的列的明文文本名称。
名称	允许您为列提供自定义名称。
描述	管理员配置的说明。
选项	提示您为列配置的各种属性，例如过滤器或排序方向。
聚合	配置此列的聚合功能（参见第6.8（../Dynamic-Lists/#Aggregation）节）。
排序	允许按照该列的数据对报表进行排序。
隐	允许隐藏列。
类型	包含底层数据类型。

6.2。数据类型

数据库中的每一列都有一个固定的数据类型，用于描述此列可以容纳的数据的形式。有不同的数据类型，例如文本，数字或日期。常见的数据类型有：

VARCHAR	具有最大长度的文本。
整数	整数
DOUBLE / FLOAT	浮点数字。
DECIMAL	十进制数。

CLOB / BLOB	任意长文本/二进制数据。
日期	日期。

列显示在报告中的顺序与方面**列表配置**中的列顺序**相同**。您可以通过简单地将列拖放到所需位置或通过上下文菜单移动它们来更改顺序（右键单击）。

要将当前配置的第一印象更改为“**预览**”方面。该预览显示报告的前**50**条记录。窗口底部的工具栏包含所选条目和列数的信息，并允许您对列表中的条目进行删除。您进一步将报告的运行时间分配到服务器上的运行时间以及从请求直到数据显示在浏览器中的时间。双击一个条目将在新窗口中打开相应的记录。通过上下文菜单（右键单击），您可以直接从预览页面从方面**列表配置**中访问许多功能。

6.3。别名，排序和隐藏列

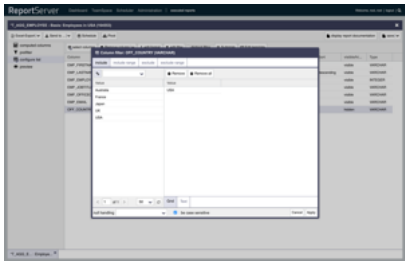
要更改列的名称，请返回到**列表配置**，然后单击与所需列对应的名称列中的单元格。如果管理员为列提供了额外的元数据，您将找到默认的明文名称以及每列的描述。如果您提供自定义名称，则此名称优先于默认名称。

最后报告中的记录顺序可以通过选择单个列的顺序来更改（在“**订单**”下）。如果一个订单配置了多个列，则先遵循先前列的顺序。

在**隐藏**下，您可以决定个别列是否不应该是最终报告的一部分。如果列只能用于数据的排序或过滤，这可能是最好的。

6.4。过滤器

过滤器允许限制最终报告中包含的记录。为此，您可以在每个列上创建明确包含排除记录的某些记录或过滤器的过滤器。在本节中，我们将介绍使用**ReportServer**进行过滤的基础知识。使用表达式语言的高级选项（如通配符过滤或过滤）将在本章的后续章节中进行详细讨论。要在列上指定过滤器，请转到方面**列表配置**，选择列并选择过滤器工具（从工具栏中），或者直接双击该列。



您现在有两种基本方法来描述报告应包含的记录集：

- 1. **Â**» 选择应在最终报告中的记录（包含）
- 2. **Â**» 选择应忽略的记录（排除）。在这种情况下，所有记录除了指定的记录将包含在最终报告中。

如果您同时定义了两者的，包含和排除过滤器，则排除不再适用于整个记录，而只适用于包含过滤器定义的子集。

这意味着，例如，如果您已经包含并排除了相同的值，则相应的记录不是结果集的一部分。

过滤器对话框允许您基于单个值或定义值范围来定义包含以及排除过滤器。为此使用相应的选项卡。

标签的结构类似，它们类似于常用选择对话框。在左侧，您可以根据通过不同过滤器（其他列上的过滤器以及当前列中的过滤器）配置的记录的整体，找到当前列的值。

选择通过双击或拖放操作类似于选择对话框。要定义一个值范围，您需要提供两个连续的值，从而定义集合（从**A**到**B**）。

或者从可用值列表中选择值，您可以将视图切换到文本输入字段，而不用使用该字段输入所需的值或使用剪贴板（即复制和粘贴）。如果您定义范围，则需要将两个值分开一个减号（“**A - B**”）。请注意减号左侧和右侧的空格。通过不指定您创建打开间隔的范围之一。例如，间隔“所有值大于或等于**5**”将被写为“**5**”。

要加载当前列的所有现有值，而不将现有过滤器纳入选择面板中，左键单击搜索字段旁边的链符号（强制一致性）。（通常你不需要使用这个功能。）

默认情况下，过滤器区分大小写，即记录中的值必须与过滤器表达式完全匹配。如果您想忽略大小写，请禁用“**区分大小写**”选项。请注意，这可能会对结果查询的性能产生负面影响。

6.4.1。空单元格（NULL）

定义过滤器时应注意的数据库的特点是它们处理空单元格。通过**空单元格**，我们了解一个不包含任何值的单元格（在数据库中，这称为**NULL**值）。**NULL**特别是与空字符串“”或数字**o**不同。

由于**NULL**与任何其他值不同，您将通过在该列上定义任何过滤器来自动排除**NULL**值。如果仅定义了排除过滤器，则这可能是直观的，而关系代数中的默认处理则为现代数据库系统提供了基础。

因此，如果您在列上定义过滤器，但不想删除空单元格，那么您必须包括它们。相反，如果列上没有其他过滤器，则只需明确排除空单元格即可。要配置空单元格的处理，请使用过滤器对话框中的相应下拉框。

6.4.2。过滤浮点数（浮点数/双倍）

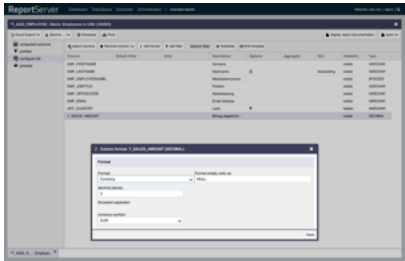
请注意，根据定义，测试浮点数相等是有问题的。因此，如果要在float或double类型的列上进行过滤，则应该定义一个过滤器范围。例如，而不是过滤值5.1，而是使用5.0009 - 5.1001的范围过滤器。

6.4.3。不一样：删除重复

通过在方面列表配置中的“ 删除重复 ”选项，可以禁止在结果集中包含重复记录。

6.5。格式

在ReportServer中，您可以在aspec 列表配置中单独定义每列的输出格式。为此，请选择一列，然后单击工具栏中的按钮格式或从上下文菜单中选择格式。这将打开根据底层数据类型略有不同的格式对话框。



提供以下格式：

数	该值被解释为一个数字，并显示配置的小数位数。或者，您可以启用数千分隔符。
百分	该值被解释为百分比。
科学的	数字用科学记数法写成。
货币	值被解释为数字，并附加货币符号。
日期	该值被解释为指定目标格式的日期和输出。如果基础数据类型不是日期类型，则必须另外指定源格式。在这种情况下，还需要指定如何处理错误的日期。通过选项清除错误日期并替换错误的日期，您可以配置如何完成。如果启用了干净的错误日期，则格式正确但日期不正确的日期（例如，35/03/2012）将变为最可能的正确日期（在本例中为04/04/2012）。对于不正确格式的值（例如，“上一个星期二”），第二个选项允许您定义替换。在这里，您还可以提供使用\$表达式{} -用替换表达式语言值的现有价值。请参阅附录表达语言 (../Expression-language/)。
文本	将值解释为文本 - 以这种方式，例如，保留前面的零。
模板	允许使用\$ {} - 表达式语言表达式。请参阅附录表达语言 (../Expression-language/)。

请注意，如果将报导出为Excel，则格式将直接在Excel中进行配置。

6.6。报告文档

通过按钮文档，您可以访问当前报告的文档。这样可以概览整个报告的配置。但是请注意，文档始终基于您的变体的最新存储版本。有关报告文档的更多信息，请参阅附录A. (../Report-documentation/)

Employees in USA

09.10.2014

Employees in USA

09.10.2014

Created: 09.10.2014
Last changed: 09.10.2014
Creation: null
Edit: 485

General

UUID: 74c2d9f3-4f18-483a-9c30-392945b3c9d8
Base report: T_AGE_EMPLOYEE - Basic
Remove duplicate rows: No
Subtotals: No

Columns

EMP_FIRSTNAME

Hidden: no

Order: -

Empty cells: -

Aggregation: -

EMP_LASTNAME

Hidden: no

Order: descending

Empty cells: -

Aggregation: -

EMPLOYEEID

Hidden: no

Order: -

Empty cells: -

Aggregation: -

EMP_EMPLOYEEIDNUMBER

Hidden: no

Order: -

Empty cells: -

Aggregation: -

EMP_JOBTITLE

Hidden: no

Order: -

Empty cells: -

Aggregation: -

EMP_JOBCODE

Hidden: no

Order: -

Empty cells: -

Aggregation: -

EMP_EMAIL

Hidden: no

Order: -

Empty cells: -

Aggregation: -

EMP_ADDRESS

Hidden: yes

Order: -

Empty cells: -

Aggregation: -

EMP_COUNTRY

Hidden: yes

Order: -

Empty cells: -

Aggregation: -

EMP_CITY

Hidden: no

Order: -

Empty cells: -

Aggregation: -

UUID: 74c2d9f3-4f18-483a-9c30-392945b3c9d8

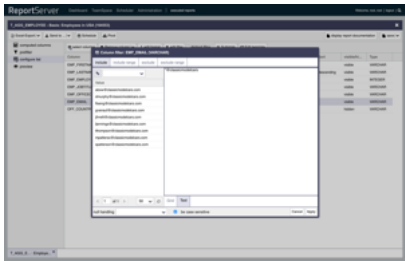
Page 1 of 1

6.7. 使用通配符进行过滤（*，？）

除了在过滤器对话框中指定范围，您还可以使用通配符指定多个值。通配符*匹配任意任意的字符序列。这包括空序列。例如：“fish*”与值“钓鱼”以及值“鱼”相匹配。相比之下，通配符？匹配一个任意字符。

通配符可用于简单的包含和排除过滤器以及范围过滤器。此外，您还可以在过滤器对话框的搜索掩码中使用通配符。

请注意，在过滤器范围内使用通配符时，ReportServer会尝试选择最大可能的范围。也就是说，如果表达式匹配多个值，ReportServer将选择“最低”值作为间隔的下限，将“最大”选为上限。另外请注意，如果对于其中一个边界，没有值匹配，那么该表达式描述一个空的间隔。当定义包含范围时，可能会导致意外的结果：例如，“a* - z*”为空，如果不存在以“z”开头的值。



6.8. 聚合

汇总描述根据某些分组属性对数据进行汇总或压缩的过程。对于底层数据中存在的每个组，结果集将包含一个数据记录。让我们以性别和年龄的特征列表为例。可能的汇总是按性别分列的平均年龄。在这种情况下，具有n个数据记录的列表将被汇总到具有每个性别的一个记录的结果。

因此，当使用聚合时，我们需要区分描述记录属于哪个组的属性（在示例中为性别）的属性，以及使用聚合函数将要汇总到单个值的属性。

ReportServer提供以下聚合功能

- Â» 平均值：计算每组的平均值
- Â» 计数：计算每组的记录数
- Â» 最大值：输出每组的最大值
- Â» 最小值：输出每组的最小值
- Â» 总和：计算每组所有值的总和
- Â» 差异：计算统计学差异
- Â» 计数不同：计算每个组的记录，但是，仅考虑不同的值

如果为其中一个列选择了聚合函数，那么没有聚合的所有其他列将被视为分组列。报表不可能包含既不是分组列也不是聚合列的列。要定义列的聚合，请转到列表配置，并从列表中选择聚合函数，单击网格中的相应单元格。

6.8.1. 小计

如果启用聚合，则每个组中只有一个条目将存在于生成的报告中。如果您还希望在每个组中输出底层条目，您可以使用列表配置工具栏中的“小计”功能。在对话框中，您可以指定要用于分组的列。所有不属于此列表且不具有聚合功能的列将被考虑用于显示单个组记录。

6.8.2. 过滤和聚合

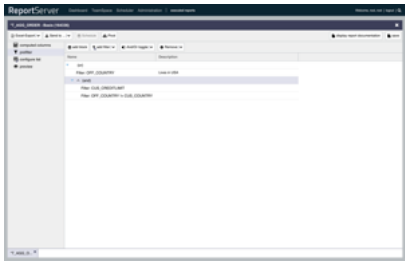
方面列表配置中的所有过滤器都作用于可见的最终结果。在聚合列上进行过滤时，这意味着聚合首先出现，并且过滤器对聚合的结果起作用。如果我们在示例中的聚合时间列上放置一个过滤器，则可以筛选平均年龄，而不是进入聚合的值。因此，过滤器“30-”不改变两个组中的任何组的聚合值，但是抑制结果集中具有小于30的任何组。

如果您熟悉数据库语言SQL，则聚合列上的过滤器对应于SQL中的HAVING过滤器。

要偏离所描述的行为，并过滤进入聚合的值，而不是过滤聚合的结果，您可以使用下一节中描述的所谓的预过滤器。

6.8.3. 预过滤

预过滤器是一种强大的工具，用于选择进入报告的数据。它们的应用范围远远超出了方面列表配置中可用的简单过滤器。三个主要区别是：



将过滤器与ANDs和OR组合

方面列表配置中的过滤器总是一起工作（AND），即当且仅当满足所有过滤条件时，记录才在结果集中。对于预过滤器，另一方面，您可以创建任意组合的连接和断开（ANDs和OR）。

比较列之间的值

列比较允许创建基于数据记录的两个属性之间的关系的过滤条件。一个例子可以是列A中的值与列B中的值不同的所有记录。

过滤进入聚合的值

如上所述，方面列表配置中的过滤器总是作用于可见结果，这意味着在聚合的情况下，过滤器对聚合的结果起作用。相比之下，在任何聚合之前，总是先评估预过滤器，从而允许过滤进入聚合的值。在我们的示例中，例如，您可以计算所有年龄大于21岁的人的平均年龄。请注意，如果没有选择聚合，则在这方面没有任何差异。

6.8.4。预滤波表达式的组合

预过滤器包含一组过滤器表达式，任意嵌套并与ANDs和OR组合。

让我们考虑一个具有四个过滤器表达式的示例（请注意，最后一个过滤器是列比较）：

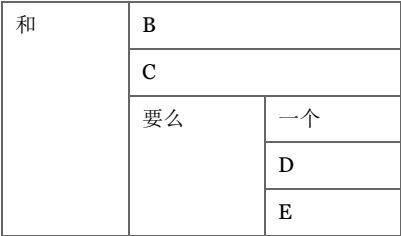
```
1 | A := Age > 30
2 | B := sex = female
3 | C := place of residence = New York
4 | D := Age < 5
5 | E := place of birth = place of residence
```

表达方式

```
1 | (B AND C AND (A OR D OR E))
```

描述了居住在纽约的所有女性，其中30岁，5岁以下或在纽约出生的女性。

ReportServer将此表达式可视化为所谓的树：



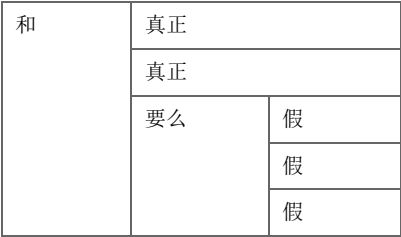
AND和OR节点称为封闭下面节点的块。在该示例中，第一AND块因此包含节点B和C以及OR块。OR块又包含块A，D和E。

要评估树，也就是计算特定数据记录的真值，ReportServer首先评估各个过滤器表达式的真值。每个评估产生的值为TRUE或FALSE。

示例记录：

```
1 | Age = 25
2 | Sex = female
3 | place of residence = New York
4 | place of birth = Boston
```

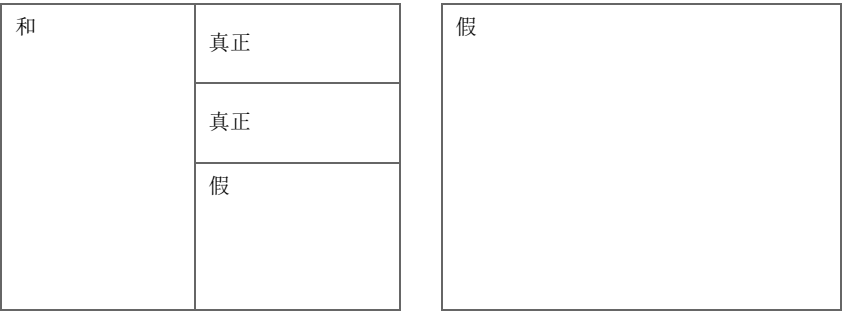
这产生以下树



在下一步骤中，ReportServer计算块的值。为此，ReportServer递归地标识不包含任何其他块的块，并计算其值的逻辑运算（AND或OR）。

因此，当且仅当所有封闭值为TRUE时，AND块被评估为TRUE。另一方面，如果OR值至少有一个值为TRUE，则为TRUE。

当块被评估时，它被替换为相应的值。重复此过程，直到根块被评估为止。在我们的例子中，这通过两个步骤完成：



因此，示例记录不是结果集的一部分。

您可以通过工具栏将过滤器和块添加到预过滤器。它们总是插入当前选定的块下方（如果未选择块，则为根块）。您可以通过将节点拖动到新位置来移动节点。请注意，块内的过滤器顺序与结果无关。此外，AND块下方的块自动转换为OR块，反之亦然，OR块下方的块转换为OR块。

通过“和/或切换”，您可以将所有AND块与OR块交换，反之亦然。

6.9。过滤器中的计算

`$ {}` - 表达式语言也可以在过滤器表达式中使用。而不是一个值，您可以使用文本模式提供表达式。除了默认替换（见附录B (../Expression-language/)），还提供了以下对象：

今天	允许基于当前日期写入表达式的日历对象。例如，这可以用于过滤不超过7天的所有发票。
聚集	提供基于当前列的所有值访问计算。这可以例如用于异常值检测。
分析性	提供访问分析功能。例如，您可以过滤列的前10％

6.9.1。使用今天的对象

“today”对象是一个日历对象，您可以使用该日历对象来指定相对于当前日期的日期，也就是说，对象使用当前日期和时间进行初始化。在以下功能的帮助下，您可以操作日期（或时间）

第一天	将日历对象设置为当月的第一天的午夜
最后一天	将对象设置为当前月份最后一天的最后一秒
addDays	将作为参数给出的天数添加到当前日期（参数可以为负数以减去日期）
addMonths	将指定的月数添加到当前日期（参数可以为负数以减去日期）
addYears	将指定的年数添加到当前日期（参数可以为负数以减去日期）
setDay	将日历设置为特定日期
setMonth	将日历设置为特定的月份
setYear	将日历设置为特定年份
clearTime	清除时间字段，即时间设置为午夜
addHours	将指定的小时数添加到当前日期（参数可以为负数以减去天数）
addMinutes	将指定的分钟数添加到当前日期（参数可以为负数以减去日期）
addSeconds	将指定的秒数添加到当前日期（参数可以为负数以减去日期）
setHours	设置小时字段
setMinutes	设置分钟字段
setSeconds	设置秒字段
格式	此功能根据指定的掩码格式化日期。如果底层数据类型不是日期类型，而是文本类型，则可能需要这样做（有关更多信息，请参阅附录C. (../Date-Format/)）

例
您要过滤上个月的所有发票。这可以使用以下包含范围完成：

`$ {today.firstDay ()} .addMonths (-1) } - $ {today.firstDay ()} .addSeconds (-1) }`

如果列的类型为VARCHAR（即文本列）和数据，例如格式化为日/月/年，则必须使用格式函数指定格式：

`$ {today.firstDay ()} .addMonths (-1) .format (“dd / MM / yyyy”) } - $ {today.firstDay ()} .addSeconds (-1) .format (“dd / MM / yyyy”) }`

有关使用日期格式的更多信息，请访问<http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html> (<http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html>)。

6.9.2。使用agg对象

“agg”对象提供对整个列的数据的聚合功能的访问。这允许您定义例如基于列的平均值的过滤器表达式。

以下功能可用：

- Â» avg：计算平均值。
- Â» count：计算值的数量。
- Â» countDistinct：计算当前属性的distinct（即不同）值的数量。

- Â» sum: 总结列的值。
- Â» 方差: 计算列的方差。
- Â» max: 计算最大值。
- Â» min: 计算最小值。

公式 `{agg.avg () - 10} - {agg.avg () + 10}` 定义了围绕该列的平均值的宽度 **20** 的间隔。例如，该表达式可以用作范围过滤器来查找与平均值偏离太多的值。

如果在范围过滤器中使用公式，则负（减法）和函数之间不能有任何空格，因为 **ReportServer** 将会将负值识别为范围分隔符。

要计算 **ReportServer** 需要扫描整个表的聚合函数。如果报表建在一张大桌子上，可能需要一些时间。

6.9.3。使用分析对象

与 **agg-object** 类似的分析对象提供了对给定属性的所有值计算结果的函数的访问。与始终返回单个值的 **agg-object** 相反，分析对象计算一组值。因此，分析对象只能用于简单的包含和排除过滤器表达式; 不在范围表达式中。

分析对象可以使用以下功能：

- Â» top (n)：选择前 **n** 个值
- Â» bottom (n)：选择最小的 **n** 值
- Â» topGrouped (n, 'column name')：根据列“列名”分组，选择此列的前 **n** 个值
- Â» bottomGrouped (n, 'column name')：根据列“列名”分组选择此列的最小 **n** 个值

如果 **n** 是整数，那么 **n** 被解释为要返回的值的数量。如果 **n** 是 **0** 和 **1** 之间的十进制数，那么它将被解释为百分比。因此，顶部 (**0.1**) 匹配前 **10**%。

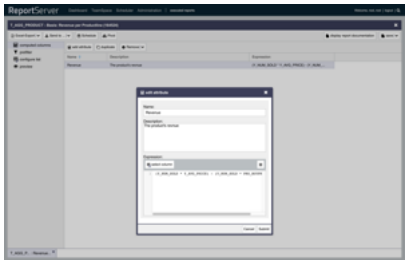
对于“分组”功能，您必须另外提供列的名称。这些值将首先根据此列进行分组，然后每组将计算顶部（相应的，底部）的值。

在包含销售价值的属性上定义的公式 `{analy.topGrouped (5, 'CUS_COUNTRY')}` 将过滤每个国家/地区的前 **5** 名销售额（如果 **CUS_COUNTRY** 是包含客户国家/地区的属性的名称）。因此，生成的报告包含所有在其国家/地区前 **5** 名的客户。

类似于 **agg** 函数，**ReportServer** 在计算任何分析函数时需要扫描整个表。如果报表建在一张大桌子上，可能需要一些时间。

6.10。计算列

使用计算列，您可以通过创建基本数据中不存在的新属性来增强报表，但可以以某种方式计算。计算规则直接在 **SQL** 中定义，可以直接在数据库上进行快速处理。另一方面，这意味着根据所使用的数据库（请向您的管理员询问更多信息）可用的表达式和功能可能略有不同。在本节中，我们只介绍了一个非常基本的可能性子集。有关更多信息，请参阅 **SQL** 的任何好的介绍。



如果您添加了计算列（方面计算列，添加属性），则可以将其用作报表中的任何其他列。这意味着如果计算列包含在结果中，还需要将计算列添加到所选列的列表（宽高比配置）。

定义计算列时，您只能将计算量仅作为报表基础数据中可用的属性，即计算列不能相互依赖。

要配置计算列，请转到方面计算列。在这里您可以添加新属性并更改现有属性。请注意，计算列的名称需要为空格和唯一的字母数字。

SQL 提供了大量的可能性来定义计算列，并且仅描述这些列的一小部分将超出本手册的范围。但是，我们至少要给出少数代码示例，用于经常发生的用例。

我们将以下示例放在具有四列的表中：

A, B: 含有文本的列

C, D: 包含整数的列

对于简单的计算，您可以使用基本的算术运算，如C + D来定义两个属性的和。要连接两个文本属性，您可以（使用大多数数据库）使用|| -operator。在某些情况下也可以使用+操作符或函数调用CONCAT（A，B）。

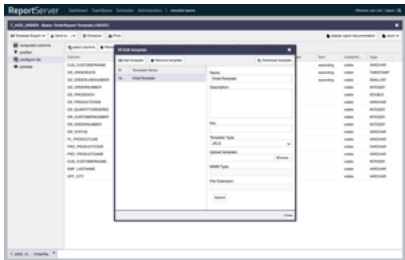
使用CASE表达式可以定义条件。例如，这允许您根据其价值对值进行分类。

```
1 CASE
2   WHEN COLUMN_NAME < 500 THEN 1
3   WHEN COLUMN_NAME < 1000 THEN 2
4   WHEN COLUMN_NAME < 2000 THEN 3
5   WHEN COLUMN_NAME < 3000 THEN 4
6   ELSE 0
7 END
```

请注意，第一个匹配的表达式获胜，并且所有可能的返回值都需要具有相同的数据类型。

6.11。模板

除了标准的导出格式（如EXCEL或CSV），动态列表允许将结果数据直接导出到预准备的模板中。在本节中，我们将介绍基本的模板功能。



在方面列表配置中，您可以通过按钮编辑模板访问功能。打开的对话框显示左侧当前可用的模板，并显示右侧当前所选模板的详细信息。通过工具栏，您可以添加新模板，下载当前选定的模板或删除模板。

模板由名称，描述和类型组成。目前，以下模板类型可用：

JXLS	允许使用Microsoft Excel定义模板。
速度	允许定义纯文本模板。
XSLT	允许定义XML模板。

如果你添加了一个新的模板，你需要上传相应的文件。在以下段落中，我们为每个模板类型提供了一个小例子。完整的文档超出了本手册的范围。更多信息可以在网上找到：

JXLS	http://jxls.sourceforge.net/1.x/index.html (http://jxls.sourceforge.net/1.x/index.html)
速度	http://velocity.apache.org/ (http://velocity.apache.org/)
XSLT	http://www.w3.org/TR/xslt (http://www.w3.org/TR/xslt)

6.11.1。Excel模板与JXLS

ReportServer集成了jXLS模板引擎，允许您直接在Microsoft Excel中准备复杂的模板，并直接从报表平台填写。通过专门的指令，您可以管理数据的插入方式和位置。对jXLS提供的可能性的详细描述超出了本文档的范围，因此我们将只会快速了解基础知识。有关jXLS项目页面上的完整文档，请访问<http://jxls.sourceforge.net/1.x/reference/tags.html> (<http://jxls.sourceforge.net/1.x/reference/tags.html>)。

6.11.2。基础

使用两个基本结构形成模板：标签和替换。一个所谓的标签围绕一个区域，并定义如何解释这些单元格：

```
1 <jx:forEach items="${data}" var="department">
2   ${department.name} | ${department.chief}
3 </jx:forEach>
```

在示例中，示例中的每一行在Excel文档中成为一行，而管道符号（|）标记单元格的末尾，即第二行由两个单元格组成。同样重要的是要注意，命令应该从一个单元格的开头开始，并且不存在前导的空格。

在该示例中，标签<jx: forEach ...>标记块的开头。该块由对应的终端标签</ jx: forEach>关闭。数据中的每个记录重复块内的所有单元格（通过替换“数据”将记录提供给模板引擎）。要访问“forEach-block”中的各个属性，可以使用“var”属性中定义的变量：在示例“department”中。要访问单个属性，使用\$，其中“department”是绑定到当前数据行的变量，“colname”是属性的名称（即相应列的名称）。请注意，变量名称区分大小写。


```
1  <?xml version="1.0"?>
2  <xsl:stylesheet version="1.0"
3  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4  xmlns:xhtml="http://www.w3.org/1999/xhtml">
5  <xsl:template match="/">
6      <myXmlFormat>
7          <xsl:apply-templates select="//xhtml:tr"/>
8      </myXmlFormat>
9  </xsl:template>
10 <xsl:template match="xhtml:tr[position() = 1]">
11     <attributes>
12         <xsl:apply-templates mode="attributes" />
13     </attributes>
14 </xsl:template>
15 <xsl:template match="xhtml:th" mode="attributes">
16     <attribute>
17         <xsl:value-of select="."/>
18     </attribute>
19 </xsl:template>
20 <xsl:template match="xhtml:tr[position() > 1]">
21     <record>
22         <xsl:apply-templates mode="values"/>
23     </record>
24 </xsl:template>
25 <xsl:template match="xhtml:td" mode="values">
26     <value>
27         <xsl:value-of select="."/>
28     </value>
29 </xsl:template>
30 </xsl:stylesheet>
```

结果可能如下所示：