

目录

目录	1
如何创建报告	9
报告编辑界面的介绍	9
一键美化	9
1、优雅布局主题	9
2、报表优化主题	9
筛选器的应用	10
报告级筛选器和页面级筛选器	10
列表筛选:	10
文本筛选:	10
日期筛选:	11
动态时间段:	11
静态时间段:	11
报告级筛选器应用	11
页面级筛选器应用	12
报表级筛选器	12
列表筛选:	12
文本筛选:	12
条件筛选:	13
高级筛选:	13
日期筛选:	13
动态时间段:	13
静态时间段:	13
连续筛选	13
列表筛选器	13
列表筛选器用法:	14
列表筛选器的样式编辑:	14
成员的筛选	14
默认值的设置	14
列表筛选器支持默认选中第一项	15
受影响的筛选器	15
树状筛选器	15
树状筛选器制作方式:	15
树状筛选器的样式编辑:	15
日期筛选器	15
时间筛选器用法:	16
快捷选择:	16
动态时间:	16
固定时间:	16
时间筛选器的样式编辑:	16
范围筛选器	16
筛选器组	17
跨数据模型筛选	17
查询筛选器	18
图表联动	18
跳转报表	18
图表的跳转报表	18
普通图表的跳转	18
设置跳转报表的步骤:	18
使用跳转报表的步骤:	18

透视表的跳转	19
设置跳转报表的步骤:	19
使用跳转报表的步骤:	19
表格的跳转	19
设置跳转报表的步骤:	19
使用跳转报表的步骤:	20
跳转报告页	20
跳转网页	20
参数的应用	20
创建参数	20
在计算字段中使用参数	21
参数控件	21
示例: 使用参数做动态的维度	21
全局参数	22
书签	22
非图表控件	23
注释框	23
文本控件	23
矩形控件	23
图片控件	23
Tab控件	24
网页控件	24
圆形	24
直线	24
肘形箭头	24
表计算	24
差异	25
百分比	25
差异百分比	26
总额百分比	27
累积和	27
移动计算	27
排名	28
表计算的动态计算方式	28
计算字段	29
日期函数	29
日期函数的作用	29
示例	29
平台中的日期函数	29
日期加法:DATEDIFF()	29
DATEADD(日期部分, 整数, 日期型表达式)	29
示例	29
日期减法:DATEDIFF()	29
DATEDIFF(日期部分, 开始日期, 结束日期)	29
示例	29
日期部分:DATEPART()	29
DATEPART(日期部分, 日期型表达式)	30
示例	30
日期截断:DATETRUNC()	30
DATETRUNC(日期部分, 日期型表达式)	30
示例	30
转为日期:DATE()	30

DATE (表达式)	30
示例	30
转为日期时间: DATETIME ()	30
DATETIME (表达式)	30
示例	30
日期天数: DAY ()	30
DAY (日期)	30
示例	30
日期月份: MONTH ()	30
MONTH (日期型表达式)	30
示例	30
日期年份: YEAR ()	31
YEAR (日期型表达式)	31
示例	31
当前日期: TODAY ()	31
TODAY ()	31
示例	31
当前日期时间: NOW ()	31
NOW ()	31
示例	31
日期串解析: DATEPARSE ()	31
DATEPARSE (格式串, 字符串)	31
示例	31
数值函数	31
数值函数的作用	31
示例	31
平台中的数值函数	31
绝对值: ABS (数值)	32
示例	32
上取整: CEIL (数值)	32
示例	32
下取整: FLOOR (数值)	32
示例	33
PI ()	33
示例:	33
EXP (数值)	33
示例	33
RADIANS (数值)	33
示例	33
DEGREES (数值)	33
示例	33
SIN (数值)	33
示例	33
COS (数值)	33
示例	33
TAN (数值)	34
示例	34
COT (数值)	34
示例	34
ASIN (数值)	34
示例	34
ACOS (数值)	34

示例	34
ATAN(数值)	34
示例	34
ATAN2(数值1, 数值2)	34
示例	34
POWER(底数, 指数)	34
示例	34
LOG(底数, 真数)	34
示例	35
ROUND(表达式, 小数位数)	35
示例	35
聚合函数	35
聚合函数的作用	35
示例	35
求和聚合:SUM()	35
SUM(表达式)	35
示例	35
平均聚合:AVG()	35
AVG(表达式)	35
示例	36
计数聚合:COUNT()	36
COUNT(表达式)	36
示例	36
去重计数聚合:COUNTD(表达式)	36
COUNTD(表达式)	36
去重计数聚合:MEDIAN(表达式)	36
MEDIAN(表达式)	36
百分位聚合:PERCENTILE(表达式, 百分位)	36
PERCENTILE(表达式, 百分位)	36
最大值聚合:MAX(表达式)	36
MAX(表达式)	36
最小值聚合:MIN(表达式)	36
MIN(表达式)	36
属性值聚合:ATTR(表达式)	37
ATTR(表达式)	37
去重拼接:GROUP_CONCAT_DISTINCT(表达式, 分隔符)	37
逻辑函数	37
逻辑函数的作用	37
示例	37
IF条件	37
IF 条件 THEN 返回值1 ELSE 返回值2	37
示例	37
IFF条件: IIF(条件, 值1, 值2)	37
IIF(条件, 值1, 值2)	37
示例	37
CASE: CASE 表达式 WHEN 值1 THEN 返回值1 ... [ELSE 默认返回值]	37
CASE 表达式 WHEN 值1 THEN 返回值1 ... [ELSE 默认返回值]	37
示例	37
比较运算: 表达式1 比较符 表达式2	37
表达式1 比较符 表达式2	37
示例	38
逻辑与: 布尔值1 AND 布尔值2	38

布尔值1 AND 布尔值2	38
示例	38
逻辑或: 布尔值1 OR 布尔值2	38
布尔值1 OR 布尔值2	38
示例	38
逻辑否: NOT 布尔值	38
NOT 布尔值	38
示例	38
空值判断: ISNULL(表达式)	38
ISNULL(表达式)	38
示例	38
最小值比较: MINIMUM(表达式1, 表达式2, ...)	38
MINIMUM(表达式1, 表达式2, ...)	38
示例	38
最大值比较: MAXIMUM(表达式1, 表达式2, ...)	38
MAXIMUM(表达式1, 表达式2, ...)	38
示例	39
是否空: IFNULL(表达式1, 表达式2)	39
IFNULL(表达式1, 表达式2)	39
示例	39
若空则零: ZN(表达式)	39
ZN(表达式)	39
示例	39
字符函数	39
字符串函数的作用	39
示例	39
平台中的数值函数	39
转为字符串: STR(表达式)	40
STR(表达式)	40
示例	40
截取字符串: SUBSTR(字符串, 开始位置, [长度])	40
SUBSTR(字符串, 开始位置, [长度])	40
示例	40
字符串长度: LENGTH(字符串)	40
LENGTH(字符串)	40
示例	40
替换字符串: REPLACE(字符串, 子串, 替换串)	40
REPLACE(字符串, 子串, 替换串)	40
示例	40
字符串转大写: UPPER(字符串)	40
UPPER(字符串)	40
示例	40
字符串转小写: LOWER(字符串)	41
LOWER(字符串)	41
示例	41
移除前导空格: LTRIM(字符串)	41
LTRIM(字符串)	41
示例	41
移除尾随空格: RTRIM(字符串)	41
RTRIM(字符串)	41
示例	41
JSON字段提取: JSON_EXTRACT(字段访问器, JSON字符串)	41

JSON_EXTRACT(字段访问器, JSON字符串)	41
示例	41
包含:CONTAINS(字符串, 子串)	41
CONTAINS(字符串, 子串)	41
示例	41
结尾包含:ENDWITH(字符串, 子串)	41
ENDWITH(字符串, 子串)	41
示例	42
开头包含:STARTWITH(字符串, 子串)	42
STARTWITH(字符串, 子串)	42
示例	42
开头截取:LEFT(字符串, 整数)	42
LEFT(字符串, 整数)	42
示例	42
结尾截取:RIGHT(字符串, 整数)	42
RIGHT(字符串, 整数)	42
示例	42
查找字符串位置:FIND(字符串, 子串, [起始位置])	42
FIND(字符串, 子串, [起始位置])	42
示例	42
正则捕获提取:REGEXP_EXTRACT_NTH(表达式, 模式, 索引)	42
REGEXP_EXTRACT_NTH(表达式, 模式, 索引)	42
示例	42
正则替换:REGEXP_REPLACE(字符串, 模式, 被替换串)	42
REGEXP_REPLACE(字符串, 模式, 被替换串)	42
示例	43
类型转换函数	43
类型转换函数的作用	43
示例	43
转为日期:DATE()	43
DATE(表达式)	43
示例	43
转为日期时间:DATETIME()	43
DATETIME(表达式)	43
示例	43
转换为字符串:STR(表达式)	43
STR(表达式)	43
示例	43
日期串解析:DATEPARSE()	43
DATEPARSE(格式串, 字符串)	43
示例	43
转为浮点数:FLOAT()	43
FLOAT()	44
示例	44
转为整数:INT()	44
INT()	44
示例	44
表计算函数	44
FIRST()	44
FIRST()	44
示例	44
INDEX()	44

INDEX()	44
示例	44
LAST()	44
LAST()	44
示例	44
SIZE()	44
SIZE()	45
示例	45
LOOKUP(表达式, [偏移量]) along [维度]	45
LOOKUP(表达式, [偏移量])	45
示例	45
RUNNING_AVG(表达式) along [维度]	45
RUNNING_AVG(表达式) along [维度]	45
示例	45
RUNNING_COUNT(表达式) along [维度]	45
RUNNING_COUNT(表达式) along [维度]	45
示例	45
RUNNING_MAX(表达式) along [维度]	45
RUNNING_MAX(表达式) along [维度]	45
示例	45
RUNNING_MIN(表达式) along [维度]	46
RUNNING_MIN(表达式) along [维度]	46
示例	46
RUNNING_SUM(表达式) along [维度]	46
RUNNING_SUM(表达式) along [维度]	46
示例	46
WINDOW_AVG(表达式, [开始位置, 结束位置]) along [维度]	46
WINDOW_AVG(表达式, [开始位置, 结束位置]) along [维度]	46
示例	46
WINDOW_COUNT(表达式, [开始位置, 结束位置]) along [维度]	46
WINDOW_COUNT(表达式, [开始位置, 结束位置]) along [维度]	46
示例	46
WINDOW_MAX(表达式, [开始位置, 结束位置]) along [维度]	46
WINDOW_MAX(表达式, [开始位置, 结束位置]) along [维度]	46
示例	46
WINDOW_MIN(表达式, [开始位置, 结束位置]) along [维度]	46
WINDOW_MIN(表达式, [开始位置, 结束位置]) along [维度]	47
示例	47
WINDOW_SUM(表达式, [开始位置, 结束位置]) along [维度]	47
WINDOW_SUM(表达式, [开始位置, 结束位置]) along [维度]	47
示例	47
WINDOW_VAR(表达式, [开始位置, 结束位置]) along [维度]	47
WINDOW_VAR(表达式, [开始位置, 结束位置]) along [维度]	47
示例	47
WINDOW_VARP(表达式, [开始位置, 结束位置]) along [维度]	47
WINDOW_VARP(表达式, [开始位置, 结束位置]) along [维度]	47
示例	47
WINDOW_STDEV(表达式, [开始位置, 结束位置]) along [维度]	47
WINDOW_STDEV(表达式, [开始位置, 结束位置]) along [维度]	47
示例	47
WINDOW_STDEVP(表达式, [开始位置, 结束位置]) along [维度]	47
WINDOW_STDEVP(表达式, [开始位置, 结束位置]) along [维度]	48

示例	48
RANK(表达式, ["asc" "desc"]) along [维度表达式]	48
RANK(表达式, ["asc" "desc"]) along [维度表达式]	48
示例	48
RANK_UNIQUE(表达式, ["asc" "desc"]) along [维度表达式]	48
RANK_UNIQUE(表达式, ["asc" "desc"]) along [维度表达式]	48
示例	48
RANK_MODIFIED(表达式, ['asc' 'desc']) along [维度]	48
RANK_MODIFIED(表达式, ['asc' 'desc']) along [维度]	48
示例	48
RANK_DENSE(表达式, ["asc" "desc"]) along [维度表达式]	48
RANK_DENSE(表达式, ["asc" "desc"]) along [维度表达式]	48
示例	48
跨视图粒度计算	48
分析模块	50
趋势线分析	50
条件一: X轴跟Y轴有至少1个连续度量字段, 如下方示意图所示。	50
条件二: X轴(Y轴)为连续日期字段, Y轴(X轴)为连续度量字段, 如下方示意图所示。	50
聚类分析	50
数据字典	51
转换字段类型	51
创建地理字段	51
创建组	52
聚合方式	52
聚合方式支持百分位数	52
数据格式	52
数据桶	52
层级	53
创建层级	53
上卷下钻	53
聚焦下钻	53
支持的图形	54
排序	54
图表的排序	54
表格的排序	55
总计	55
一、普通图表的总计设置	55
二、透视表的总计设置	55
三、表格的总计设置	55
参考线	56
智能推荐	56
智能推荐用法:	56
智能问答	56
功能限制:	57
智能分析	57
数据医生	57
报告整体	57
单个组件	58
功能配置	58

如何创建报告

完成数据模型的建立后，我们便可以创建报告进行数据分析跟可视化了，下面我们分步骤介绍，如何在平台上新建一个报告：

1. 进入报告模块，点击新建报告。



2. 进入报告编辑界面，点击添加数据模型将需要分析的数据模型添加进报告。



3. 选择报告要存放的路径后，新建报告。



4. 在画布上添加需要的控件，并将数据字段拖入控件的数据区，完成图表的绘制。



报告编辑界面的介绍

报告的编辑界面如图所示，我们可以将其划分为如下几个区域：

数据区：显示可供使用的数据字段。

画布区：我们可在画布上自由绘制数据可视化报告。



控件区：显示可添加至画布上的各类图表控件、辅助控件。

控件配置区：在画布上选中已添加的图表控件时，此区域会显示该控件的配置项，我们将数据区的字段拖进控件配置区以绘制图表。




一键美化

1、优雅布局主题

进入报告操作页面，[优雅布局]美化内容为图表位置的布局调整  点击后进入操作页面，[优雅布局]美化内容为图表位置的布局调整，[深度优化]即将上线，敬请期待~ 

选择[优雅布局]，预览美化后效果，同时可点击右上角[前后对比]，对比美化前与美化后效果图

若对美化后效果满意，点击[立即应用]，生成美化后的报告页，也可选择[返回编辑]，继续编辑该报告  对齐规则

- 步骤一：竖列第一列左对齐

从左到右找到竖列的第一个控件或控件组左对齐，左对齐距离画布左边距为20px，若第一竖列中控件中存在边距小于20px的控件，则以最小左边距为基准左对齐

- 步骤二：左对齐控件横向覆盖区域顶对齐


左对齐的控件，自上而下在横向区域高度范围内，与相应左对齐的控件顶对齐

- 步骤三：横向同组的组件差值范围内的同类控件大小进行平均

同一类控件间的宽度差值在10%以内，需对该控件进行大小一致的调整，调整原则以同类控件宽度值中最小的宽度进行统一

同一类控件间的高度差值在10%以内，需对该控件进行大小一致的调整，调整原则以同类控件高度值中最小的宽度进行统一

- 步骤四：控件行分组间的平均分布规则

行分组内控件基于画布宽度进行平均分布计算，如果平均分布后控件之间的间距小于20px，则保持现有规则，大于20px，则控件以最左侧控件为基准，20px为间距往右侧排列 

2、报表优化主题

一键美化新增[报表优化]主题，继[优雅布局]对图表位置进行合理排布后，针对图表内细节调整的主题也上线了，根据用户当前所建图表，自动生成合适的图表样式，避免了图表内细节调整的痛苦，既显著降低了分析门槛，又提供了更为丰富的可视化效果。

在报告编辑页面，创建完图表后，点击一键美化，进入美化主题选择页面。

[优雅布局]下方新增[报表优化]主题，点击预览可查看优化后图表详情 [报表优化]主要图表及图表优化点：

- 指标卡：创建指标卡时默认勾选[字体大小适应窗口]，美化后取消

- 折线图：折线图线条尺寸在美化后默认为12

- 表格：在表格内美化后维度字段居左，度量字段居右

- 柱状图：图表内柱状图尺寸调整为40

- 仪表盘：美化后应用主题一

- 区域图：区域图样式中取消标记点

- 存在X、Y轴的图形且包含轴名称的，一键美化后轴名称隐藏且图例标题隐藏 美化后如图所示： 在美化时，可自由切换主题颜色查看美化效果，也可[优雅布局]+[报表优化]两种主题叠加使用，制作出精美的报告

筛选器的应用

报告级筛选器和页面级筛选器

报告级筛选器：整份报告，该字段的筛选将应用到同一模型所有的报表。报告级筛选器支持对维度字段的筛选，可进行列表筛选、文本筛选和日期筛选

页面级筛选器：当前页面，该字段的筛选将应用到同一模型所有的报表。页面级筛选器支持对维度字段的筛选，可进行列表筛选、文本筛选和日期筛选

列表筛选：

- 手动：指用户输入字段成员名，该匹配是精确匹配，比如用户输入“东”，只会筛选字段成员“东”
- 列表：用户选择字段成员名
- 包含：指选出字段成员，可多选
- 排除：指排除字段成员，可多选
- 使用全部：指包含字段所有的成员，包括未来新增的字段成员

文本筛选：

- 包含以下所有条件：指and，所有条件的交集
- 包含以下任一条件：指or，所有条件的并集
- 包含：字段成员任意位置有输入的文本

- 开头是：字段成员的开头是输入的文本
- 结尾是：字段成员的结尾是输入的文本
- 不包含：字段成员任意位置不包含输入的文本
- 等于：字段成员是输入的文本，精确匹配
- 可增加条件筛选

日期筛选：

动态时间段：

日期会动态变化。提供了5种动态时间段：今天、昨天、本年至今、本月至今、本周至今，还提供了自定义功能。

自定义：

- 前N天：
 - 输入数字，代表前N天
 - 时间锚点：今天，指包括今天。昨天，指不包括今天。
 - 可更改时间单位：年、月、周、天、时、分、秒
- 开始日期：
 - 日期选择：指开始的日期
 - 时间锚点：今天，指开始日期到今天。昨天，指开始日期到昨天。

静态时间段：

选择一段日期，该日期是固定的。

报告级筛选器应用

1. 在页面的空白区域点击，将出现报告设置入口。

2. 点击进入报告设置的属性面板，进入数据面板。

3. 将维度字段拖入筛选器。

比如将维度地区拖入筛选器，将触发列表筛选和文本筛选。

- 1) 在列表选项中，勾选东北、华北，则报告只包含这两个地区的数据。

- 2) 在文本选项中，输入包含“北”，则报告包含“东北”、“华北”、“西北”，以及未来新增含有“北”的成员。

4. 将时间维度字段拖入筛选器。

比如将时间类型字段发货日期拖入筛选器，将触发时间筛选。

- 1) 动态时间，今天，选择的是今天2017-4-10。
- 2) 动态时间，自定义，包含今天在内的前30天。

(1) 打开自定义面板，选择前30天，时间锚点：今天。

(2) 选择的时间段是2017-3-12至2017-4-10。

3) 静态时间，选择2015-1-1至2015-12-31。

5. 筛选支持多个维度。

页面级筛选器应用

1. 在页面的空白区域点击，将出现页面设置入口。

2. 点击进入页面设置的属性面板，进入数据面板。

3. 将维度字段拖入筛选器。

报表级筛选器

报表级筛选器指：该字段的筛选将应用到本报表。

报表级筛选器支持对维度字段的筛选，可进行列表筛选、文本筛选、条件筛选和高级筛选。

报表筛选器支持对日期类型字段的筛选，可进行日期筛选。

报表级筛选器支持对度量字段的筛选，可进行连续范围的选择。

列表筛选：

- 手动：指用户输入字段成员名，该匹配是精确匹配，比如用户输入“东”，只会筛选字段成员“东”
- 列表：用户选择字段成员名
- 包含：指选出字段成员，可多选
- 排除：指排除字段成员，可多选
- 使用全部：指包含字段所有的成员，包括未来新增的字段成员

文本筛选：

- 包含以下所有条件：指and，所有条件的交集
- 包含以下任一条件：指or，所有条件的并集
- 包含：字段成员任意位置有输入的文本
- 开头是：字段成员的开头是输入的文本
- 结尾是：字段成员的结尾是输入的文本
- 不包含：字段成员任意位置不包含输入的文本
- 等于：字段成员是输入的文本，精确匹配
- 可增加条件筛选

条件筛选：

（地区）按照**销售额的求和**，大于等于100000，指筛选销售额大于100000的地区。

销售额可更换为其他的度量。

聚合方式**求和**也可以调整为其他类型。

大于等于也可更换为其他条件，支持更换为大于，等于，小于，小于等于，不等于。

高级筛选：

（地区）按照**销售额的求和**，**最大N个**，指筛选出销售额排名前N的地区。

销售额可更换为其他的度量。

聚合方式**求和**也可以调整为其他类型。

最大：指排名前N

最小：指排名后N

日期筛选：

动态时间段：

日期会动态变化。提供了5种动态时间段：今天、昨天、本年至今、本月至今、本周至今，还提供了自定义功能。

自定义：

- 前N天：

-- 输入数字，代表前N天

-- 时间锚点：今天，指包括今天。昨天，指不包括今天。

-- 可更改时间单位：年、月、周、天、时、分、秒

- 开始日期：

-- 日期选择：指开始的日期

-- 时间锚点：今天，指开始日期到今天。昨天，指开始日期到昨天。

静态时间段：

选择一段日期，该日期是固定的。

连续筛选

范围：可以设定筛选的数值范围。数值范围只能输入在默认数值范围内的值。

默认数值范围是根据报表上放置的维度分组后，产生的最小值和最大值。本例中，则是指每个地区每类产品的销售数量的数值范围。

列表筛选器

一个列表筛选器对应一个维度字段。

列表筛选器只对当前页面的图表起作用。

列表筛选器在编辑状态下编辑，使用必须在预览状态或者阅览状态下。

列表筛选器用法：

1. 选择**列表筛选**控件，在画布上画出**列表筛选器**。



2. 在列表筛选器的数据面板的操作，将维度字段**地区**拖入**维度**。默认是关联当前页面的同一数据模型的所有报表。



3. 列表筛选器完成了编辑。在预览状态下使用：



4. 列表筛选器在阅览状态下使用：



列表筛选器的样式编辑：

1. 选择画布上的**列表筛选器**，点击列表筛选器的**样式**面板。

2. 筛选器有4种样式：列表多选，列表单选，下拉多选，下拉单选。

3. 可以修改标题区的背景，标题区文字的颜色、字号和字体。

4. 可以修改选项区的文字颜色、字号和字体。

5. 可以修改整个列表筛选器的背景颜色和边框。



成员的筛选

对列表筛选器维度成员筛选后，在列表筛选器中只显示已经被筛选过后的成员。对成员的筛选支持：列表筛选、文本筛选、条件筛选、高级筛选。

比如“省份”，只让用户筛选“黑龙江、吉林、辽宁”东北三省。

1. 在维度放入**省份**。

2. 在成员筛选，放入**地区**。

3. 在筛选器中，选择列表，选择**东北**。

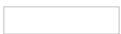
列表筛选则只显示黑龙江、吉林、辽宁这3个省。

默认值的设置

在编辑状态里，可以设置列表筛选器的默认值。设置完后，预览状态下，用户见到的是勾选了默认值的筛选器。

比如华东区是公司最关心的区域

1. 在维度放入**地区**。



2. 点击**默认值设置**中的**设置**按钮。



3. 在弹出的窗口中选择**华东**。



4. 设置完成后，默认值将影响关联的图表。



列表筛选器支持默认选中第一项

试想这样一个场景，数据库的数据每天都会更新，而我们想要始终看到最新的一天数据，这时候就可以借助此项功能来实现。

如图所示，我们对列表筛选器控件中的**订单日期**字段降序排序后，将默认值设置为**默认选中第一项**，则每次打开报告时，该列表筛选器会默认选中第一项日期，也即最新的有数据的日期。



受影响的筛选器

列表筛选器可以影响本页面的其他列表筛选器。影响是单向的。

比如地区筛选器，影响省份筛选器、城市筛选器。

1. 新建3个列表筛选器，分别放入**地区**、**省份**、**城市**。

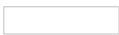


2. 设置**地区**影响**省份**和**城市**筛选器。



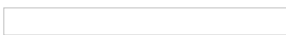
完成了设置，在预览界面查看效果。

3. 预览界面，选择**地区**下的**华东**，则**省份**和**城市**都受到了影响。省份只有华东地区的省份，城市只出现了华东地区的城市。



树状筛选器

树状筛选器的使用方法如图所示



树状筛选器控件跟平台里的列表筛选器控件很相似，不同之处在于列表筛选器控件只能添加一个维度字段，而树状筛选器控件可以添加多个维度字段。各字段在树状筛选器内按照树形结构进行展示。

树状筛选器制作方式：

1. 选择**树状筛选**控件，在画布上添加**树状筛选器**。

2. 选中**树状筛选器**，将维度字段**地区**、**省份**拖入**维度**。默认会关联当前页面的同一数据模型的所有报表。



树状筛选器的样式编辑：

1. 选中**树状筛选器**，点击操作面板中的**样式面板**。

2. 筛选器有2种选择方式：**列表**和**下拉**。

3. 可以修改标题区的名称，文字的颜色、字号、字体以及标题区的背景。

4. 可以修改选项区的文字颜色、字号和字体。

5. 可以修改整个树状筛选器的背景颜色和边框。



日期筛选器

一个日期筛选器对应一个时间类型字段

日期筛选器可以跨数据模型关联

日期筛选器在编辑状态下编辑，可以在编辑状态、预览状态或者阅览状态重新设置时间范围，只有编辑状态的时间范围会保存。

时间筛选器用法：

1. 选择筛选器tab下的日期筛选控件，在画布上画出日期筛选器。



2. 在日期筛选器的数据面板的操作，将时间类型字段订单日期拖入维度。默认是关联当前页面的同一数据模型的所有报表。



3. 点击日期筛选器数据面板中的设置，弹出时间范围设置窗口。



4. 对时间范围可进行快捷选择，动态时间段和静态时间段的设置。

快捷选择：

快捷选择提供了全部时间、前7天、前14天、未来7天、今天、明天、本周、本月、本季度、本年、上周同期等等快捷选项，选中某项后会在右上角显示所选的具体日期时间。



动态时间：

时间会动态变化。提供了4种动态时间段：前N天、前第N天、后N天以及自定义选择开始结束时间。

- 前N天、前第N天、后N天
- -- 输入数字，代表前N天

-- 可选择是否包含今天

-- 可更改时间单位：年、月、周、天、时、分、秒

- 开始时间：可选全部时间、前第n天、前第n周的周一、前第n月的1号（以上的n均可以自主填写），或者直接选择某个单日
- 结束时间：可选昨天、上周、上月、后第n天、后第n周的周日、后第n月的最后一日（n同样可以自主填写）

固定时间：

选择一段时间，该时间是固定的。

5. 在编辑状态下编辑日期筛选器并应用，预览状态可重新调整时间范围：



时间筛选器的样式编辑：

1. 选择画布上的时间筛选器，点击时间筛选器的样式面板。
2. 可以选择筛选器的时间类型：时间范围、年月或单日。
3. 可以修改整个时间筛选器的背景颜色和边框。



范围筛选器

当我们想要用数值型字段对图表进行筛选时，比如“筛选出销售额总和大于80万的省份”，可以借助范围筛选器控件。

下面举例给大家演示范围筛选器控件的用法。 筛选前的图表如图所示。

我们可以选择筛选类型（这里我们选择“至少”），并输入要筛选的范围（这里我们输入“800000”），这样的含义即是“筛选出销售额大于800000”。应用筛选后的效果如图所示，我们即筛选出了销售额大于80万的省份。

我们还可以点击“加载范围”去计算图表中的销售额取值范围，加载后筛选器控件内会显示取值范围，如图所示。

我们可以滑动滑块或直接在输入框内输入数值来选择要筛选的数值范围（这里我们筛选销售额在80万跟100万之间的省份）。

那么如何来添加这样一个范围筛选器呢，操作过程很简单，在画布上添加一个范围筛选器控件，拖入数值型字段，并勾选要关联的图表即可。

筛选器组

平台可将多个筛选器组成筛选器组，组内的筛选器变更选项后不会立即对图表生效，可以等组内多个筛选器都完成变更操作后再一起对图表生效。这样一来，减少了不必要的数据查询，可提高需要变更多个筛选器选项时的操作体验。

筛选器组的使用效果如下图所示：

将筛选器组成筛选器组的操作步骤如下：

第一步：添加一个**筛选器组**控件。

第二步：选中**筛选器组**控件，在设置面板中勾选需要影响的筛选器，被选择的筛选器将组成筛选器组，并受**筛选器组**控件影响。

跨数据模型筛选

列表筛选器和时间筛选器支持跨数据模型的筛选。

比如报告中有两个数据模型A和B，两个数据模型中都有地区这个字段，目前可以使用其中一个模型的地区列表筛选器，来控制两个基于两个模型的图表。

举个例子

假设现在报告中有2个图表，分别来自两个不同的数据模型；现在用A模型的地区字段做列表筛选器，希望能对这2个图表一起生效。

先使用模型A“超市数据”的地区字段来做列表筛选器，然后进行跨数据模型关联设置。

选择需要关联的模型，以及相应的字段。

设置完成后，点击**确定**，同时在列表筛选器的报表关联设置中，选择你需要关联的**报表**。

设置完成后，在预览状态下，你选择列表筛选器中的数据，可以看到，基于两个不同的数据模型的报表都发生了改变。

查询筛选器

编辑者可查看影响单张报表的筛选器汇总，在图表的预览状态下，点击图表控件更多操作的查看关联筛选器，可以看到所有影响该图表数据的筛选器汇总。

图表联动

可以在图表之间设置图表联动，设置联动后，对一张图表的数据项的选择可以对另一张图表进行筛选。

效果如图所示



当在左侧饼图选中**中南**地区时，右侧气泡图将显示中南地区的城市，同理，左侧饼图选**华东**地区时，右侧气泡图将显示华东地区的城市。

图表联动的配置非常简单，选中要做为图表筛选器的图表，点击右上角的选项，选择**创建图表联动**，在弹出的设置页中选择要关联的图表，即可完成图表联动的设置。操作示例如图所示：



跳转报表

图表的跳转报表

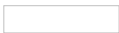
图表的跳转报表指的是通过一张报表，点击该报表的图形，会跳转到另外一张报表。

普通图表的跳转

比如：地区销售额的图表，点击图表的柱子，可以查看某个地区的销售明细。本例中，只取了2012年和2013年的数据，应用了一个筛选器。

设置跳转报表的步骤：

1. 制作起始报表



2. 制作跳转报表



3. 回到起始报表，鼠标移动到Y轴的度量**销售额**，Hover点击下拉菜单，选择**设置跳转报表**。



4. 在设置跳转窗口，选择需要跳转的报表。点击**确定**完成了跳转报表的设置。



5. 高级设置的说明



蓝色区域的筛选：是起始报表传递的筛选条件

黄色区域的筛选：是起始报表拥有的筛选器

使用跳转报表的步骤：

1. 在起始报表的浏览状态下，鼠标左键选择**柱子**，再单击鼠标右键，点击**跳转报表**。



2. 将会跳转到一个新的窗口。



透视表的跳转

比如：地区销售汇总表, 点击表里值，可以查看某个地区的销售明细。本例中，只取了2012年的数据，应用了一个筛选器。

设置跳转报表的步骤：

1. 制作起始报表



2. 制作跳转报表



3. 回到起始报表，鼠标移动到属性面板标签中的度量度量值，Hover点击下拉菜单，选择设置跳转报表。



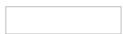
4. 在设置跳转窗口，选择需要跳转的报表。点击确定完成了跳转报表的设置。



5. 完成后的起始报表



6. 高级设置的说明



蓝色区域的筛选：是起始报表传递的筛选条件

黄色区域的筛选：是起始报表拥有的筛选器

使用跳转报表的步骤：

1. 在起始报表的浏览状态下，鼠标左键选择值，再单击鼠标右键，点击跳转报表。



2. 将会跳转到一个新的窗口。



表格的跳转

比如：地区销售汇总表, 点击表里值，可以查看某个地区的销售明细。本例中，只取了2012年的数据，应用了一个筛选器。

设置跳转报表的步骤：

1. 制作起始报表



2. 制作跳转报表



3. 回到起始报表，鼠标移动到列销售额，Hover点击下拉菜单，选择设置跳转报表。



4. 在设置跳转窗口，选择需要跳转的报表。点击确定完成了跳转报表的设置。



5. 高级设置的说明



蓝色区域的筛选：是起始报表传递的筛选条件

黄色区域的筛选：是起始报表拥有的筛选器

使用跳转报表的步骤：

1. 在起始报表的浏览状态下，点击有下划线的值。



2. 将会跳转到一个新的窗口。



跳转报告页

平台还支持跳转至报告页，先给大家展示一下功能效果。

如下方动图所示，我们在柱形图中选中关心的数据（这里我们选择销售额最高的华东地区），右键，在弹出的工具栏中选择**跳转报告**，我们将会跳转至更详细的报告页，同时会对该报告进行筛选，只查看华东地区的数据。



那么，如何来设置跳转报告呢，它的设置方式跟设置跳转报表类似。

在**报告编辑**界面，在柱形图对应的**数据面板**中，鼠标移动至**销售额**字段后右键，在下拉菜单中选择设置跳转报告页。



在弹出的设置窗口中，选择要跳转的报告，跟报告页（这里我们选择事先制作好的“被跳转报告页”），确定后即完成了跳转报告页的设置。



之后在报告的浏览界面即可使用跳转功能。

跳转网页

可对字段设置跳转链接，设置后，点击字段的成员项，可打开链接对应的网页。效果如图所示，点击**渠道**，可进入渠道对应的网站：

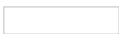


下面将分步骤介绍如何实现。

第一步： 右键选择要设置跳转的字段，在弹出的菜单中选择**设置跳转链接**。



第二步： 在弹出的设置界面中输入要跳转的链接，可输入固定的链接，比如www.baidu.com，也可以在链接中插入字段，跳转时则会把字段的取值作为链接。



第三步： 配置好链接后，在浏览界面点击对应的字段成员即可进行跳转。

参数的应用

创建参数

1. 点击**参数BETA**控件>**新建参数**。



2. 打开**新建参数**的面板，输入参数名称、选择参数的类型、选择参数的数据格式。

指定参数接受值的方式。可从以下选项中选择：

任意 — 参数控件是字段中的简单类型

区间 — 参数控件可用于选择指定范围中的值

列表 — 参数控件提供可供选择的可能值的列表

这些选项的可用性由数据类型确定。

例如，字符串参数只能接受列表。 如果选择**列表**，则必须指定值列表。在列表中可输入值。

如果选择**区间**，则必须指定最小值、最大值和步长。例如，可以定义介于1至10的整数范围，并将步长设置为 2。

编辑完成后，单击**添加**。完成参数的创建，创建完成后，在**管理参数**界面中可以看到创建完的参数

在计算字段中使用参数

您可通过参数来动态修改计算中的值。您可以使用参数而不是手动编辑计算（以及所有依赖计算）。然后在要更改值时，可打开参数控件并更改值；使用该参数的所有计算都会更新。

若要在计算中使用参数，在计算字段的编辑框中能够插入参数，放在公式中的一个新位置，或替换当前公式的一个部分：

参数控件

在工具栏中选择**参数BETA**>**添加参数控制器**。

在绘图区中拖一个矩形后，会出现**参数控制器**。

在**参数控制器**的数据面板中，选择参数，这里会列出当前报告中您所创建的所有参数，可以进行选择。

选择完成后，可以在样式中选择参数控件的样式，样式分为滑块和输入框、可以修改标题区、选项区和背景边框。

滑块：

输入框：

示例：使用参数做动态的维度

使用参数一般可以分为两个步骤，一个就是创建参数，一个就是使用参数

1. 创建参数

创建名称为动态维度的字符串类型的参数



2. 使用参数，在计算维度中插入参数，如下图一个简单的case语句，对参数进行了判断。



新建图表，把刚才创建的计算维度拖入到X轴中



然后图表区域中加入参数控件



在预览模式下，可以通过切换参数来实现对图表的动态变化



全局参数

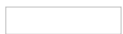
项目中心—全局参数管理，可以查看已有的全局参数并对其进行编辑与删除的操作，全局参数可支持跨报告使用，对报告进行批量处理，提升处理效率。

全局参数目前支持类型包括整数、小数、字符串、日期、日期和时间；数据格式可选自动、数值和百分比；可选范围包括任意、区间、列表。

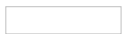


引用全局参数

数据源为Restful API时，参数配置中的value值支持引用全局参数，输入#可引用全局参数，下拉中展示当前用户拥有查看权限的参数列表，可点击参数右侧箭头，点击【详情】可查看对应参数设置详情



数据模型设置页面，选择添加【自定义视图】时，SQL语句中使用#唤起全局参数的下拉框，若设置全局参数后不建议抽取



时间参数控制器

产品现支持时间参数控制器，可同时控制日期字段的起止时间，使用更为灵活。

书签

在浏览报告时，有的报告提供了筛选器、参数等控件，报告的读者会根据自己关心的数据，选择筛选器的选项，比如下图，我们选择查看江苏、浙江、上海三个地区对应的销售额数据。



倘若需要经常查看浙江、江苏、上海的销售额，那我们便可以将当前报告的状态保存成一个书签，下次访问书签，报告将自动进入到书签记录的状态（本例中，即地区筛选器选中华东，省/自治区筛选器选中浙江、上海、江苏，Tab控件选中Tab2）。



添加书签时，可以选择该书签是否对他人可见，若选择所有人可见，则所有拥有该报告浏览权限的用户都可以看到该书签。



添加书签后，可以在书签列表中访问已添加的书签。



作为报告制作者，如果事先知道不同读者对数据的侧重点，也可以提前针对不同读者提前添加多个书签，报告的读者可以直

接访问书签快速筛选报告数据。



当前，书签会保存报告的如下状态： 1、当前所在报告页； 2、筛选器控件的选中项； 3、参数字段的选中项； 4、Tab控件的选中项； 5、表格的自定义指标。

非图表控件

注释框

平台支持添加注释框控件，可以添加在报告的任意位置，鼠标移动到控件上，即会浮窗显示提示信息。演示效果如图所示：



添加控件的方式如图所示：



注释框提供大中小三种尺寸，如图所示：



注释框支持插入度量，如图所示：



文本控件

1. 拖入文本控件后，用户可以在其中编辑文本。
2. 修改文字的颜色、大小、字体（目前提供微软雅黑和宋体）。
3. 对文字进行加粗、斜体、下划线、删除线。
4. 修改文字的对齐方式，左对齐、居中对齐、右对齐。
5. 对选中的部分文字设置url链接。
6. 文本支持插入度量，从而实现文字描述跟数字结合。



文本中支持插入度量，你可以设置度量的聚合方式。



在预览模式下，点击该文字，会跳转到相应的网页。



可以修改文本框的背景颜色、边框颜色、边框线条、以及圆角方式。



矩形控件

可以修改矩形控件的背景颜色、边框颜色、圆角大小。



图片控件

当用户点选图片控件后，系统会自动弹出选择框，让你选择需要上传的图片选择完成后，系统默认显示图片的原始大小，用户可以调整图片大小 右侧的图片设置面板中，可以让用户选择是否要保持长款比（默认勾选）、可以更改图片的边框、圆角大小。



Tab控件

平台可以添加**Tab**控件，每个**Tab**控件可添加多个标签页，每个标签页内可放置多张图表，可灵活切换标签页进行展示。

演示效果如下方图片所示：



制作方式如下

1. 在辅助工具中，选择“**Tab**”，在画布上添加一个**Tab**控件。



2. 双击画布上的**Tab**控件，进入控件编辑状态，将其他图表拖入**Tab**控件（这里我们拖入一个饼图）。



3. 我们也可以直接在**Tab**控件内创建图表，在**Tab**控件的编辑状态下（双击**Tab**控件进入编辑状态），点击图表控件添加图表，会直接将图表添加在**Tab**控件里面。

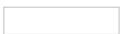


另外，**Tab**控件内不需要的图表，可以删除，或者移出**Tab**控件。移出的方式：双击**Tab**控件进入编辑状态，然后选中**Tab**控件内不需要的图表直接移出即可，如下图所示。



网页控件

平台可以添加**网页**控件，每个控件可添加网址链接，即可快速导航至该网页，同时也能进行清除网址重新编辑的操作，如图所示。



同时，**网页**控件支持UniqueID传参，在网址后输入#，可快速插入用户账号，当用户登录时，**网页**控件的URL即会带上用户的账号信息。



圆形

报告新增**圆形**辅助工具，选择**圆形**，画布中出现圆形框，可手动调节圆形大小，还可在样式面板修改圆形的边框样式和大小。



直线

报告新增**直线**辅助工具，选择**直线**，画布中出现直线，可手动调节直线长度、方向，还可在样式面板修改直线的边框样式和大小。



肘形箭头

报告新增**肘形箭头**辅助工具，选择**肘线**，画布中出现肘线，可手动调节肘线的长度、方向，还可在样式面板修改边框样式、大小和两个端点的箭头类型。



表计算

有时，仅展示原始数据并不满足我们的需求。如下图，展示的是每年各地区的销售额。



我们可能希望观察各地区销售额的占比，原始数据里并没有这样的信息，这种情况下可以借助表计算得到。如下图所示，我们使用了**总额百分比**表计算。



如果我们想观察每年各地区销售额的排名情况，则可使用排名表计算，结果如下图所示。



提供了7种表计算：差异、百分比、差异百分比、总额百分比、累积和、移动计算、排名。具体的使用方法可参考相关章节。

选中度量字段进行右键，下拉菜单内即可进入表计算。



表计算的配置面板如下。新用户可能对面板中的分区、寻址有疑惑。



分区为表计算划定**范围**，寻址为表计算指定**方向**。比如我们需要在对每年的地区进行排名，分区就是**年**，寻址是**地区**。



如果我们对调一下，把**地区**变为分区，**年**变为**寻址**，那么我们得到的就是每个地区下各年的销售额排名，如下图所示：



差异

差异是一种表计算。它计算的是分组中某一项与其前一项的差异，如：2014年“销售额”是100，2015年是200，则表计算后，销售额的差异是

$$\$ 100 = 200 - 100 \$$$

即

$$\$ D_x = A_x - A_{\{x-1\}} \$$$

注：这里的x-1并不是一定的，可以和其他项做计算，参照下文

差异只能出现在数字型的度量上，如要计算“销售额”的差异，首先需要把要计算的“销售额”胶囊拖拽到左边的面板上，并点击小三角找到表计算菜单选项(如下图)：



打开表计算面板后，在计算类型中选择第一项



计算内容是决定当前视图中，哪些维度用来分区计算，哪些用来在区分好的组里面按照顺序做计算的，如上图中配置，我们可以得到以下图表中显示的计算结果。



可以看到，每一个分组的第一项因为没有前值，所以计算结果为空。

我们根据邮寄方式划分了四个小组分别作差异计算，在每个分组中沿着年(订单日期)中的2011 2012 2013 2014 按顺序计算与前一项的差异。

如果想要和其他项做差异计算，可以打开图2的配置面板中最下面的配置菜单



百分比

百分比是一种表计算。它计算的是分组中某一项与其前一项的比值(以百分比呈现)，如：2014年“销售额”是100，2015年是200，则表计算后，销售额的百分比是

$$\$ 200\% = 200 \div 100 \times 100\% \$$$

即

$$\$ P_x = A_x \div A_{x-1} \times 100\% \$$$

注：这里的\$ x-1 \$并不是一定的，可以和其他项做计算，参照下文

百分比只能出现在数字型的度量上，如要计算“销售额”的百分比，首先需要把要计算的“销售额”胶囊拖拽到左边的面板上，并点击小三角找到表计算菜单选项(如下图)：



打开表计算面板后，在计算类型中选择第二项



计算内容是决定当前视图中，哪些维度用来分区计算，哪些用来在区分好的组里面按照顺序做计算的，如上图中配置，我们可以得到以下图表中显示的计算结果。



可以看到，每一个分组的第一项因为没有前值，所以计算结果为空。

我们根据邮寄方式划分了四个小组分别作差异计算，在每个分组中沿着年(订单日期)中的2011 2012 2013 2014 按顺序计算与前一项的百分比。

对比上下两张图表，我们可以发现2014年销售额是2013年的200%。

如果想要和其他项做百分比计算，可以打开配置面板中最下面的配置菜单



差异百分比

差异百分比是一种表计算。它计算的是分组中某一项与其前一项的差异相对其自身的比值(以百分比呈现)，如：2014年“销售额”是100，2015年是200，则表计算后，销售额的差异百分比是

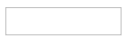
$$\$ 100\% = (200 - 100) \div 100 \times 100\% \$$$

即

$$\$ P_x = (A_x - A_{x-1}) \div A_{x-1} \times 100\% \$$$

注：这里的\$ x-1 \$并不是一定的，可以和其他项做计算，参照下文

差异百分比只能出现在数字型的度量上，如要计算“销售额”的百分比，首先需要把要计算的“销售额”胶囊拖拽到左边的面板上，并点击小三角找到表计算菜单选项(如下图)：



打开表计算面板后，在计算类型中选择第三项



计算内容是决定当前视图中，哪些维度用来分区计算，哪些用来在区分好的组里面按照顺序做计算的，如上图中配置，我们可以得到以下图表中显示的计算结果。



可以看到，每一个分组的第一项因为没有前值，所以计算结果为空。

我们根据邮寄方式划分了四个小组分别作差异百分比计算，在每个分组中沿着年(订单日期)中的2011 2012 2013 2014 按顺序计算与前一项的百分比。

对比上下两张图表，我们可以发现2014年销售额相比2013年的增长了100%。

如果想要和其他项做差异百分比计算，可以打开配置面板中最下面的配置菜单



总额百分比

总额百分比是一种表计算。它计算的是分组中某一项占它所在分组中总额的百分比，如：上海的“销售额”是200，全国销售额是2000，则表计算后，销售额的总额百分比是

$$10\% = 200 \div 2000 \times 100\%$$

$$\text{即 } P_x = A_x \div \Sigma A \times 100\%$$

总额百分比只能出现在数字型的度量上，如要计算“销售额”的总额百分比，首先需要把要计算的“销售额”胶囊拖拽到左边的面板上，并点击小三角找到表计算菜单选项(如下图)：



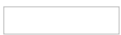
打开表计算面板后，在计算类型中选择第四项



计算内容是决定当前视图中，哪些维度用来分区计算，哪些用来在区分好的组里面按照顺序做计算。这里，总额百分比并没有前后的概念所以寻址的顺序不影响结果，寻址中的邮寄方式代表着我们计算每个邮寄方式占有所有邮寄方式总额的百分比。如上图配置，以及当前面板的配置



我们可以得到以下图表中显示的计算结果。



可以看到，每一个邮寄方式的扇形上显示的标签即为总额百分比的值。标准级的销售额占总额的58.14%。

累积和

累积和是一种表计算。它计算的是分组中某一项与其前面的所有项的总和，如：2014年“销售额”是400，2011，2012，2013年分别是100，200，300，则表计算后，销售额的累积和是

$$1000 = 100+200+300+400$$

即

$$CS_x = \Sigma_{i=0}^x A_i$$

注：累积和支持基于当前结果再做其他表计算，叫做二次表计算，参照下文

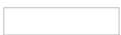
累积和只能出现在数字型的度量上，如要计算“销售额”的累积和，首先需要把要计算的“销售额”胶囊拖拽到左边的面板上，并点击小三角找到表计算菜单选项(如下图)：



打开表计算面板后，在计算类型中选择第五项



计算内容是决定当前视图中，哪些维度用来分区计算，哪些用来在区分好的组里面按照顺序做计算的，如上图配置，我们可以得到以下图表中显示的计算结果。



我们根据邮寄方式划分了四个小组分别作累积和计算，在每个分组中沿着年(订单日期)中的2011 2012 2013 2014 按顺序计算与其前面所有值的和。

对比上下两张图表，我们可以发现标准级的2014年的总销售额约为1000万。

我们可以继续在这个累积和上做表计算，只需要在配置面板中最下面的复选框中打上勾，并在下面的二次表计算类型中做进一步的计算



移动计算

移动计算是一种表计算。它计算的是分组中某一项与其前后几项的聚合计算的结果(求和、平均、最大值、最小值)，如：

2014年“销售额”是400，2013，2015年分别是200，600，则表计算后，销售额的前后一项平均值的移动计算是

$$\$ 400 = (200 + 400 + 600) \div 3 \$$$

即

$$\$ WS_x(-1,1) = AGG_{\{x-i\}^{x+1}} \$$$

注1: -1,1为确定前后项的参数，可调，在设置时只需要写绝对值 注2: 移动计算的也可以包含当前项 注3: 累积和支持基于当前结果再做其他表计算，叫做二次表计算，参照下文

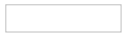
如要计算“销售额”的移动计算，首先需要把要计算的“销售额”胶囊拖拽到左边的面板上，并点击小三角找到表计算菜单选项(如下图):



打开表计算面板后，在计算类型中选择第六项



计算内容是决定当前视图中，哪些维度用来分区计算，哪些用来在区分好的组里面按照顺序做计算的，如上图中配置，我们可以得到以下图表中显示的计算结果。



我们根据邮寄方式划分了四个小组分别作累积和计算，在每个分组中沿着年(订单日期)中的2011 2012 2013 2014 按顺序计算与其前面两项的值求和。

对比上下两张图表，我们可以发现标准级的2014年与前两年的销售额总和约为800万。

图2中除了可以指定聚合计算的类型，前后多少项，是否包含当前项，还可以选中包含前面所有项或后面所有项。

若没有足够的项则留空：当前面或后面没有能够满足移动计算的足够数量的项时，不进行计算，留空。

我们可以继续在这个移动计算上做表计算，只需要在可以图2的配置面板中最下面的复选框中打上勾，并在下面的二次表计算类型中做进一步的计算



排名

排名是一种表计算。它计算的是分组中某一项在它所在分组中排序后的名次。

如要计算“销售额”的排名，首先需要把要计算的“销售额”胶囊拖拽到左边的面板上，并点击小三角找到表计算菜单选项(如下图):



打开表计算面板后，在计算类型中选择第七项：排序，还可以选择按升序或降序进行排序。



计算内容是决定当前视图中，哪些维度用来分区计算，哪些用来在区分好的组里面按照顺序做计算。如上图中配置，以及当前面板的配置



我们可以得到以下图表中显示的计算结果。



对照下图，我们可以看到上海的销售是排名第一的。

表计算的动态计算方式

表计算新增了动态的计算方式，“动态”指的是设置计算方式时，不需要指定具体的维度，只需要指定计算方向，平台会自动将需要的维度放置在分区、寻址面板。

比如，在下图中，我们对于透视表进行排名表计算，当寻址方式设置为“表（横穿）”时，会横向对表中的数据进行排名，也就是对各年的销售额进行排名。



如果我们将寻址方式设置为“表（向下）”，则会纵向对表中的数据进行排名，也就是对各地区的销售额进行排名，如图所示。



因为动态表计算不需要指定具体的维度，所以当图表的字段发生变更时，会自动根据新的字段进行计算，比如，用“类别”字段替换掉“地区”字段，则会自动按照新的字段进行表计算，计算各类别的排名。



计算字段

有时，数据表中的原始维度和度量并不满足我们的需求，因此平台提供了计算字段功能，利用各种函数新建新建字段。新建计算字段时能够快速复用上次计算字段的内容，实现继续创建计算字段。



日期函数

日期函数的作用

日期函数允许您对数据源中的日期进行操作。比如，您有一个日期字段（2018-1-1），您可以使用日期函数（例如 DATETRUNC 函数）依据这些现有值创建新的日期值。

示例

`DATEADD('day', 1, #2018-1-1#) = 2018-1-2` 这个函数相当于把日期加了1天

平台中的日期函数

日期加法：DATEADD（日期部分，整数，日期型表达式） - 返回指定日期，该日期的指定 `date_part` 中添加了指定的数字间隔。

日期加法:DATEDIFF()

DATEADD（日期部分，整数，日期型表达式）

计算开始日期和借宿日期的日期部分的差值，将日期型表达式的日期部分增加整数，可使用的日期部分包括“year”，“quarter”，“month”，“week”，“day”，“hour”，“minute”，“second”；函数返回一个日期。

示例

`DATEADD('month', 1, [订单时间])`

该表达式表示将[订单时间]增加一个月

日期减法:DATEDIFF()

DATEDIFF（日期部分，开始日期，结束日期）

计算开始日期和结束日期的日期部分的差值，可使用的日期部分包括“year”，“quarter”，“month”，“week”，“day”，“hour”，“minute”，“second”；函数返回一个整数。

示例

`DATEDIFF("DAY", [发货时间], [收货时间])` 表示从发货到收货所用的天数

日期部分:DATEPART()

DATEPART (日期部分,日期型表达式)

返回指定日期型表达式的某个部分 (由日期部分参数指定需要的部分), 可使用的日期部分包括 "year", "quarter", "month", "week", "day", "dayofweek", "hour", "minute", "second"; 函数返回一个整数。

示例

当 [日期] 为 1990-03-16 时 DATEPART("month", [日期]) = 3

日期截断:DATETRUNC()

DATETRUNC (日期部分,日期型表达式)

将日期型表达式的精度截断到指定部分 (由日期部分参数指定), 可使用的日期部分包括 "year", "quarter", "month", "day", "hour", "minute", "second"; 函数返回一个日期。

示例

当 [日期] 字段的值为 1988-08-31, 1988-09-11, 1990-03-16... 时, DATETRUNC("month", [日期]) 的结果为 1988-08-01, 1988-08-01, 1990-03-01...

转为日期:DATE()

DATE (表达式)

在给定的数字 字符串或日期表达式的情况下返回日期, 数字按1970年1月1日的秒数计算日期。

示例

DATE(1487030400) 返回日期2017年2月14日 DATE("2017-02-14") 返回日期2017年2月14日 DATE("2017-02-14 10:10:10") 返回日期2017年2月14日

转为日期时间:DATETIME()

DATETIME (表达式)

在给定的数字 字符串或日期时间表达式的情况下返回日期时间, 数字按1970年1月1日的秒数计算日期时间。

示例

DATETIME("2017-02-14") 返回日期时间2017年2月14日 0时0分0秒 DATETIME("2017-02-14 10:10:10") 返回日期时间2017年2月14日 10时10分10秒 DATETIME(1487035400) 返回日期时间2017年2月14日 1时23分20秒

日期天数:DAY()

DAY (日期)

以整数的形式返回给定日期的天。

示例

DAY(DATE("2017-04-19")) = 19

日期月份:MONTH()

MONTH (日期型表达式)

以整数的形式返回给定日期的月份。

示例

MONTH(DATE("2004-04-12")) = 4

日期年份:YEAR()

YEAR (日期型表达式)

以整数的形式返回给定日期的年份。

示例

`YEAR (DATE ("2017-04-19")) = 2017`

当前日期:TODAY()

TODAY()

返回当前日期，精确到天

示例

`TODAY ()` 返回当前日期，精确到天

当前日期时间:NOW()

NOW()

返回当前日期和时间，精确到秒

示例

`NOW ()` 返回当前日期时间，精确到秒

日期串解析:DATEPARSE()

DATEPARSE (格式串, 字符串)

将字符串转成指定格式的日期

示例

`DATEPARSE ("yyyy-mm-dd", "2017-02-14")` 返回日期2017年2月14日 `DATEPARSE ("yyyy/mm/dd hh:mi:ss", "2017/02/14 18:30:15")` 返回日期时间2017年2月14日 18点30分15秒 `DATEPARSE ("yyyy-mm-dd hh:mi:ss am", "2017-11-02 03:30:15 PM") =` 返回日期时间2017年11月2日 15点30分15秒

数值函数

数值函数的作用

数字函数允许您对字段中的数据值执行运算。字段函数只能用于包含数字值的字段。例如，您可能有一个名为“利润差异”的字段，其中包含利润中的差异值。其中一个值可能为 -700。您可以使用 `ABS` 函数返回该数字以及该字段中所有其他数字的绝对值。

示例

`ABS ([利润差异])`

因此 `ABS (-700) = 700`

平台中的数值函数

`ABS` - 返回给定数字的绝对值。

`ACOS` - 返回给定数字的反余弦。结果以弧度表示。

`ASIN` - 返回给定数字的正弦。结果以弧度表示。

- ATAN - 返回给定数字的反正切。结果以弧度表示。
- ATAN2 - 返回两个给定数字 (x 和 y) 的反正切。结果以弧度表示。
- CEILING - 将数字舍入为值相等或更大的最近整数。
- COS - 返回角度的余弦。以弧度为单位指定角度。
- COT - 返回角度的余切。以弧度为单位指定角度。
- DEGREES - 将以弧度表示的给定数字转换为度数。
- DIV - 返回将整数 1 除以整数 2 的除法运算的整数部分。
- EXP - 返回 e 的给定数字次幂。
- FLOOR - 将数字舍入为值相等或更小的最近整数。
- LN - 返回数字的自然对数。
- LOG - 返回数字以给定底数为底的对数。
- MAX - 返回两个参数 (必须为相同类型) 中的较大值。
- MIN - 返回两个参数 (必须为相同类型) 中的较小值。
- PI - 返回数字常量 pi: 3.14159。
- RADIANS - 将给定数字从度数转换为弧度。
- ROUND - 将数字舍入为指定位数。
- SIGN - 返回数字的符号。
- SIN - 返回角度的正弦。以弧度为单位指定角度。
- SQRT - 返回数字的平方根。
- SQUARE - 返回数字的平方。
- TAN - 返回角度的正切。以弧度为单位指定角度。
- ZN - 如果表达式不为 Null, 则返回该表达式, 否则返回零。使用此函数可使用零值而不是 Null 值。

绝对值: ABS(数值)

ABS(数值)

返回给定数字的绝对值

示例

因此 $ABS(-700) = 700$

上取整: CEIL(数值)

CEIL(数值)

将数字舍入为等于或大于数值的最接近整数

示例

$CEIL(3.1415) = 4$

下取整: FLOOR(数值)

FLOOR(数值)

将数字舍入为等于或小于值的最接近整数

示例

$\text{FLOOR}(200.7) = 200$, $\text{FLOOR}(-1.5) = -2$

PI ()

PI(数值)

返回数字常量 pi: 3.14159。

示例:

$\text{PI}() = 3.1415926$

EXP(数值)

EXP(数值)

返回 e 的给定数字次幂。

示例

$\text{EXP}(3) = e^3$

RADIANS(数值)

RADIANS(数值)

将以度数表示的数字转换为弧度。

示例

$\text{RADIANS}(45) = \text{PI}() / 4$

DEGRESS(数值)

DEGRESS(数值)

将以弧度表示的数字转换为度数。

示例

$\text{DEGREES}(\text{PI}() / 4) = 45$

SIN(数值)

SIN(数值)

返回角度的正弦。以弧度为单位指定角度。

示例

$\text{SIN}(\text{PI}() / 4) = 0.707106$

COS(数值)

COS(数值)

返回角度的余弦。以弧度为单位指定角度。

示例

$\text{COS}(\text{PI}() / 4) = 0.707106$

TAN(数值)

TAN(数值)

返回角度的正切。以弧度为单位指定角度。

示例

$\text{TAN}(\text{PI}() / 4) = 1$

COT(数值)

COT(数值)

返回角度的余切。以弧度为单位指定角度。

示例

$\text{COT}(\text{PI}() / 4) = 1$

ASIN(数值)

ASIN(数值) 返回数字的反正弦。结果以弧度表示

示例

$\text{ASIN}(1) = 1.570796$

ACOS(数值)

ACOS(数值) 返回数字的反正弦。结果以弧度表示

示例

$\text{ACOS}(-1) = 3.1415926$

ATAN(数值)

ATAN(数值) 返回数字的反正切。结果以弧度表示

示例

$\text{ATAN}(180) = 1.56524$

ATAN2(数值1, 数值2)

ATAN2(数值1, 数值2) 返回两个给定数字的反正切，结果以弧度表示。

示例

$\text{ATAN2}(2, 1) = 1.107148$

POWER(底数, 指数)

POWER(底数, 指数) 返回底数的指数次方。

示例

$\text{POWER}(2, 3) = 2 \times 2 \times 2 = 8$

LOG(底数, 真数)

LOG(底数, 真数) 返回真数以底数为底的对数。

示例

LOG(2, 16) = 4

ROUND(表达式, 小数位数)

ROUND(表达式, 小数位数) 将数字四舍五入为最接近的整数或指定小数位数。

示例

ROUND(3.1415, 2) = 3.14

聚合函数**聚合函数的作用**

聚合函数允许您进行汇总或更改数据的粒度。

例如，您可能想要准确知道您的商店在特定年度有多少订单。您可以使用 COUNTD 函数对公司的准确订单数进行汇总，然后按年对可视化项进行细分。

示例

COUNTD([订单编号])

平台中的数值函数	函数	语法	说明
ATTR	ATTR(expression)		属性值。如果给定表达式对于组中的所有行仅具有单个相同值，则返回该表达式的值，否则返回Null。示例：ATTR("ab") = "ab"，ATTR([field])
AVG	AVG(expression)		平均聚合。返回表达式中所有值的平均值。示例：AVG([成本])
COUNT	COUNT(expression)		计数聚合。返回组中的记录数。示例：COUNT([客户名称])
COUNTD	COUNTD(expression)		去重计数聚合。返回组中不同记录的数量。示例：COUNTD([产品名称])
MAX	MAX(expression)		最大值聚合。返回表达式在所有记录中的最大值。示例：MAX([访问量])
MEDIAN	MEDIAN(expression)		中位聚合。返回表达式在所有记录中的中位数。示例：MEDIAN([利润])
MIN	MIN(expression)		最小值聚合。返回表达式在所有记录中的最小值。示例：MIN([访问量])
PERCENTILE	PERCENTILE(expression, number)		百分位聚合。从给定表达式返回与指定数字对应的百分位处的值。示例：PERCENTILE([利润], 75)
STDEV	STDEV(expression)		样本标准差。基于群体样本返回给定表达式中所有值的统计标准差。示例：STDEV([销售额])
STDEVP	STDEVP(expression)		总体标准差。基于有偏差群体返回给定表达式中所有值的统计标准差。示例：STDEVP([销售额])
SUM	SUM(expression)		求和聚合。返回表达式中所有值的总计。示例：SUM([销售额])
VAR	VAR(expression)		样本方差。基于群体样本返回给定表达式中所有值的统计方差。示例：VAR([销售额])
VARP	VARP(expression)		总体方差。对整个群体返回给定表达式中所有值的统计方差。示例：VARP([销售额])
GROUP_CONCAT_DISTINCT	GROUP_CONCAT_DISTINCT(expression, Delimiter)		返回表达式中所有值的去重拼接结果。

求和聚合：SUM()**SUM(表达式)**

返回表达式中所有值的总计。SUM 只能用于数字字段。会忽略 Null 值。

示例

SUM([销售额])，返回销售额的总计

平均聚合：AVG()**AVG(表达式)**

返回表达式中所有值的平均值。AVG只能用于数字字段。会忽略 Null 值。

示例

AVG([销售额]) : 返回销售额的平均值

计数聚合:COUNT()

COUNT (表达式)

返回组中的项目数。不对 Null 值计数。

示例

COUNT([客户名称]) 返回客户的总数

去重计数聚合:COUNTD(表达式)

COUNTD (表达式)

返回组中不同项目的数量。不对 Null 值计数。

去重计数聚合:MEDIAN(表达式)

MEDIAN (表达式)

返回表达式在所有记录中的中位数。中位数只能用于数字字段。会忽略 Null 值。 注意，目前仅仅支持以下数据源：

- greenplum
- oracle
- sqlserver
- postgresql
- spark
- vertica
- influxdb
- impala

百分位聚合:PERCENTILE(表达式, 百分位)

PERCENTILE (表达式, 百分位)

从给定表达式返回与指定数字对应的百分位处的值。数字必须介于 0 到 1 之间（含 0 和 1），例如 0.66，并且必须是数值常量。 注意，目前仅仅支持以下数据源：

- greenplum
- oracle
- sqlserver
- postgresql
- spark
- vertica
- influxdb
- impala

最大值聚合:MAX(表达式)

MAX (表达式)

返回表达式在所有记录中的最大值。MAX只能用于数字、日期、日期时间字段。

最小值聚合:MIN(表达式)

MIN (表达式)

返回表达式在所有记录中的最小值。MIN只能用于数字、日期、日期时间字段。

属性值聚合:ATTR(表达式)

ATTR(表达式)

如果给定表达式对于组中的所有行仅具有单个相同值，则返回该表达式的值，否则返回Null 示例: ATTR("ab") = "ab", ATTR([field])

去重拼接:GROUP_CONCAT_DISTINCT(表达式, 分隔符)

GROUP_CONCAT_DISTINCT(表达式, 分隔符) 返回表达式中所有值的去重拼接结果。

逻辑函数

逻辑函数的作用

逻辑计算允许您确定某个特定条件为真还是假（布尔逻辑）。例如，您可能希望快速确定您分销商品的每个国家/地区的销售额是高于还是低于特定阈值。

示例

```
SUM([销售额]) > 10000
```

IF条件

IF 条件 THEN 返回值1 ELSE 返回值2

当条件满足时，返回返回值1，否则返回返回值2。

示例

简单用法:

```
IF [销售额] > 100 THEN "优" ELSE "差"。
```

嵌套用法:

```
IF [销售额] > 100 THEN "优" ELSE IF [销售额] > 50 THEN "良" ELSE "差"
```

IFF条件: IIF(条件, 值1, 值2)

IIF(条件, 值1, 值2)

检查某个条件是否得到满足，如果为 TRUE 则返回值1，如果为 FALSE 则返回值2，如果值2不填写，则会返回NULL。

示例

```
IIF([利润] > 0, 盈利, 亏损)
```

```
IIF([利润] > 0, 盈利)
```

CASE: CASE 表达式 WHEN 值1 THEN 返回值1 ... [ELSE 默认返回值]

CASE 表达式 WHEN 值1 THEN 返回值1 ... [ELSE 默认返回值]

当表达式为值1时返回返回值1 ... 否则返回默认返回值。可以不指定默认返回值（但这时候应该确保枚举到了所有情况）。

示例

```
CASE [产品ID] WHEN 1 THEN "家具" WHEN 2 THEN "电器" ELSE STR([产品ID])
```

比较运算: 表达式1 比较符 表达式2

表达式1 比较符 表达式2

比较两个表达式，返回布尔值，可用的比较符有：=, !=, >, <, >=, <=, 分别表示 等于, 不等于, 大于, 小于, 大于等于, 小于等于

示例

```
[销售额] >= 8000
```

逻辑与：布尔值1 AND 布尔值2

布尔值1 AND 布尔值2

逻辑与运算

示例

```
[销售额] >= 0 AND [销售额] <= 8000
```

逻辑或：布尔值1 OR 布尔值2

布尔值1 OR 布尔值2

逻辑或运算

示例

```
[用户名] = "小张" OR [用户名] = "小李"
```

逻辑否：NOT 布尔值

NOT 布尔值

逻辑否运算

示例

```
NOT [用户名] = "小张"
```

空值判断：ISNULL(表达式)

ISNULL(表达式)

判断表达式的值是否为NULL

示例

```
IF ISNULL([用户名]) THEN "匿名用户" ELSE [用户名]
```

最小值比较：MINIMUM(表达式1, 表达式2, ...)

MINIMUM(表达式1, 表达式2, ...)

对同一类型的表达式进行比较，返回两个或多个表达式对于每个记录的最小值。支持数值型、日期型、字符串。

示例

```
MINIMUM(1, 2) = 1,
```

```
MINMUM([销售额],[数量])=34
```

最大值比较：MAXIMUM(表达式1, 表达式2, ...)

MAXIMUM(表达式1, 表达式2, ...)

对同一类型的表达式进行比较，返回两个或多个表达式对于每个记录的最大值。支持数值型、日期型、字符串。

示例

```
MAXIMUM(1, 2) = 2
```

```
MAXIMUM([销售额],[数量])=1000556
```

是否空：IFNULL(表达式1, 表达式2)

IFNULL(表达式1, 表达式2)

如果表达式1不为null，则返回它，否则返回表达式2。

示例

```
IFNULL([field], 0)
```

若空则零：ZN(表达式)

ZN(表达式)

如果表达式不为null，则返回它，否则返回零。

示例

```
ZN(1) = 1
```

字符函数

字符串函数的作用

字符串函数允许您操作字符串数据（即由文本组成的数据）。

例如，可能有一个名字字段。一个成员可能为：Jane Johnson。您可以使用字符串函数将所有字母转成大写计算可能如下所示：

示例

```
UPPER([客户名称]) = JANE JOHNSON
```

平台中的数值函数

CONTAINS - 如果给定字符串包含指定子字符串，则返回 true。

ENDSWITH - 如果给定字符串以指定子字符串结尾，则返回 true。

FIND - 返回字符串中子字符串的索引位置，如果未找到子字符串，则返回 0。

FINDNTH - 返回指定字符串内的第 n 个子字符串的位置，其中 n 由 occurrence 参数定义。

LEFT - 返回字符串最左侧一定数量的字符。

LEN - 返回字符串的字符长度。

LOWER - 返回字符串，其所有字符为小写。

LTRIM - 返回移除了所有前导空格的字符串。

MAX - 对于字符串，MAX 查找数据库为该列定义的排序序列中的最高值。

MID - 返回从索引位置开始的字符串。字符串中第一个字符的位置为 1。如果添加了可选参数 length，则返回的字符串仅包含该数量的字符。

MIN - 对于字符串，MIN 查找排序序列中的最低值。

REPLACE - 在字符串中搜索子字符串，并将其替换为替代子字符串。

RIGHT - 返回字符串最右侧一定数量的字符。

RTRIM - 返回移除了所有尾随空格的字符串。

SPACE - 返回由指定数量的重复空格组成的字符串。

SPLIT - 返回字符串中的一个子字符串，并使用分隔符字符将字符串分为一系列标记。

STARTSWITH - 如果字符串以子字符串开头，则返回 true。会忽略前导空格。

TRIM - 返回移除了前导和尾随空格的字符串。

UPPER - 返回字符串，其所有字符为大写。

转为字符串:STR(表达式)

STR(表达式)

将表达式转化为字符串类型

示例

STR(200) = "200"，返回是字符串类型

截取字符串:SUBSTR(字符串, 开始位置, [长度])

SUBSTR(字符串, 开始位置, [长度])

从开始位置(字符串从1开始计数)截取指定长度的字符串返回，长度为可选参数，不给则默认截取到字符串尾部。

示例

SUBSTR("abc123", 1, 3) = "abc"，SUBSTR("abc123", 1) = "abc123"

字符串长度:LENGTH(字符串)

LENGTH(字符串)

返回给定字符串中的字符数。

示例

LENGTH("abc123") = 6

替换字符串:REPLACE(字符串, 子串, 替换串)

REPLACE(字符串, 子串, 替换串)

返回一个字符串，在该字符串中，子字符串的每次出现都会替换为替换字符串。子串和替换串必须为常量字符串，如果未找到子字符串，则字符串保持不变。

示例

REPLACE("Calculation", "ion", "ed") = "Calculated"

字符串转大写:UPPER(字符串)

UPPER(字符串)

将文本字符串转换为全大写字母。

示例


```
UPPER("product") = "PRODUCT"
```

字符串转小写:LOWER(字符串)

LOWER(字符串)

将文本字符串转换为全小写字母。

示例

```
LOWER("PRODUCT") = "product"
```

移除前导空格:LTRIM(字符串)

LTRIM(字符串)

返回移除了所有前导空格的字符串。

示例

```
LTRIM(" Sales") = "Sales"
```

移除尾随空格:RTRIM(字符串)

RTRIM(字符串)

返回移除了所有尾随空格的字符串。

示例

```
RTRIM("Market ") = "Market"
```

JSON字段提取:JSON_EXTRACT(字段访问器, JSON字符串)

JSON_EXTRACT(字段访问器, JSON字符串)

在JSON字符串中提取指定成分，只能用在gl（已聚合）的字段上。（字段访问器由两种部分组成，一种是“.xxx”的形式，另一种是"[xxx]"的形式，分别用来访问对象和数组，用户提供的访问器必须是一个常量字符串，访问器的首个"."可以省略）。

示例

- `JSON_EXTRACT("[2][1]", "[1, 2, [3, 4]]") = "4"`
- `JSON_EXTRACT("[1]", "[\"a\", \"b\"]") = "b"`
- `JSON_EXTRACT("xy", "{\"xy\":1, \"y\":2}") = "1"`
- `JSON_EXTRACT(". x. z", "{\"x\":{\"z\":1}, \"y\":2}") = "1"`
- `JSON_EXTRACT("x", "{\"x\":{\"z\":1}, \"y\":2}") = "{ z: 1}"`
- `JSON_EXTRACT(". x[1]. y", "{\"x\": [1, {\"y\":2.0}], \"z\":3.0}") = "2"`
- `JSON_EXTRACT(". x[1]. y", "{\"x\": [1, {\"y\":2.0}], \"z\":3.0}") = "2"`

包含:CONTAINS(字符串, 子串)

CONTAINS(字符串, 子串)

如果给定字符串包含指定子字符串，则返回 true。 如果给定字符串不包含指定子字符串，则返回 false。

示例

- `CONTAINS("Calculation", "alcu") = true`
- `CONTAINS("Calculation", "beu") = false`

结尾包含:ENDWITH(字符串, 子串)

ENDWITH(字符串, 子串)

如果给定字符串以指定子字符串结尾，则返回 true。

示例

- `ENDSWITH("Calculation", "tion") = true`

开头包含:STARTWITH(字符串, 子串)

STARTWITH(字符串, 子串)

如果给定字符串以指定子字符串开头，则返回 true。

示例

- `STARTSWITH("Calculation", "Ca") = true`

开头截取:LEFT(字符串, 整数)

LEFT(字符串, 整数)

返回给定字符串开头的指定字符数。

示例

- `LEFT("Calculation", 4) = "Calc"`

结尾截取:RIGHT(字符串, 整数)

RIGHT(字符串, 整数)

从给定字符串结尾起返回指定数量的字符。

示例

- `RIGHT("Calculation", 4) = "tion"`

查找字符串位置:FIND(字符串, 子串, [起始位置])

FIND(字符串, 子串, [起始位置])

返回子字符串在字符串中的位置，如果未找到子字符串，则返回0。起始位置为可选参数，如果给定此参数则从此位置开始查找，字符串中的第一个子字符串为位置1

示例

- `FIND("Calculation", "alcu") = 2`

正则捕获提取:REGEXP_EXTRACT_NTH(表达式, 模式, 索引)

REGEXP_EXTRACT_NTH(表达式, 模式, 索引)

使用正则表达式模式返回给定字符串的子字符串。子字符串与第n个捕获组匹配，其中模式的每一个括号匹配一个捕获组，其中n为给定索引。

示例

- `REGEXP_EXTRACT_NTH("abc 123", "([a-z])+\s+(\d+)", 2) = "123"`

正则替换:REGEXP_REPLACE(字符串, 模式, 被替换串)

REGEXP_REPLACE(字符串, 模式, 被替换串)

返回给定字符串的副本，其中匹配模式替换为替换字符串。

示例

- `REGEXP_REPLACE("abc 123", "\s", "--") = "abc--123"`

类型转换函数

类型转换函数的作用

类型转换函数允许您将字段从一种数据类型转换为另一种数据类型。例如，您可以将数字转换为字符串，比如将年龄值（数字）转换为字符串值，以便平台不聚合它们。

此类任务的计算可能看起来类似于以下内容：

示例

```
STR([年龄])
```

转为日期:DATE()

DATE（表达式）

在给定的数字 字符串或日期表达式 的情况下返回日期，数字按1970年1月1日的秒数计算日期。

示例

```
DATE(1487030400) 返回日期2017年2月14日 DATE("2017-02-14") 返回日期2017年2月14日 DATE("2017-02-14 10:10:10") 返回日期2017年2月14日
```

转为日期时间:DATETIME()

DATETIME（表达式）

在给定的数字 字符串或日期时间表达式 的情况下返回日期时间，数字按1970年1月1日的秒数计算日期时间。

示例

```
DATETIME("2017-02-14") 返回日期时间2017年2月14日 0时0分0秒 DATETIME("2017-02-14 10:10:10") 返回日期时间2017年2月14日 10时10分10秒 DATETIME(1487035400) 返回日期时间2017年2月14日 1时23分20秒
```

转换为字符串:STR(表达式)

STR（表达式）

将表达式转化为字符串类型

示例

```
STR(200) = "200"，返回是字符串类型
```

日期串解析:DATEPARSE()

DATEPARSE（格式串，字符串）

将字符串转成指定格式的日期

示例

```
DATEPARSE("yyyy-mm-dd", "2017-02-14") 返回日期2017年2月14日 DATEPARSE("yyyy/mm/dd hh:mi:ss", "2017/02/14 18:30:15") 返回日期时间2017年2月14日 18点30分15秒 DATEPARSE("yyyy-mm-dd hh:mi:ss am", "2017-11-02 03:30:15 PM") = 返回日期时间2017年11月2日 15点30分15秒
```

转为浮点数:FLOAT()

FLOAT ()

在给定的任何类型的表达式的情况下返回浮点数

示例

$\text{FLOAT}("3") = 3.0$, $\text{FLOAT}(2.1) = 2.1$

转为整数: INT ()

INT ()

在给定的表达式的情况下返回整数, 此函数将结果截断为最接近0的整数

示例

$\text{INT}(\text{true}) = 1$, $\text{INT}(200.6) = 200$, $\text{INT}(-0.6) = 0$

表计算函数

本文介绍表计算函数及其具体用法。

FIRST ()

FIRST ()

返回从当前行到分区中第一行的行数。例如, 下面的视图显示每季度销售额。在 Date 分区中计算 FIRST() 时, 第一行与第二行之间的偏移为 -1。

示例

当前行索引为 3 时, $\text{FIRST}() = -2$ 。

INDEX ()

INDEX ()

返回分区中当前行的索引, 不包含与值有关的任何排序。第一个行索引从 1 开始。例如, 下表显示每季度销售额。当在 Date 分区中计算 INDEX() 时, 各行的索引分别为 1、2、3、4 等。

示例

对于分区中的第四行, $\text{INDEX}() = 4$ 。

LAST ()

LAST ()

返回从当前行到分区中最后一行的行数。例如, 下表显示每季度销售额。在 Date 分区中计算 LAST() 时, 最后一行与第二行之间的偏移为 5。

示例

当前行索引为 3 (共 7 行) 时, $\text{LAST}() = 4$ 。

SIZE ()

SIZE()

返回分区中的行数。例如，下面的视图显示每季度销售额。在 Date 分区中有七行，因此 Date 分区的 Size() 为 7。

```
□
```

示例

当前分区包含五行时 SIZE() = 5。

LOOKUP(表达式, [偏移量]) along [维度]

LOOKUP(表达式, [偏移量])

返回目标行（指定为与当前行的相对偏移）中表达式的值。使用 FIRST() + n 和 LAST() - n 作为相对于分区中第一行/最后一行的目标偏移量定义的一部分。如果省略了偏移量，则可以在字段菜单上设置要比较的行。如果无法确定目标行，则此函数返回 NULL。

下面的视图显示每季度销售额。当在 Date 分区中计算 LOOKUP (SUM(Sales), 2) 时，每行都会显示接下来 2 个季度的销售额值。

```
□
```

示例

LOOKUP(SUM([Profit]), FIRST()+2) 计算分区第三行中的 SUM(Profit)。

RUNNING_AVG(表达式) along [维度]

RUNNING_AVG(表达式) along [维度]

返回给定表达式从分区中第一行到当前行的运行平均值。

下面的视图显示每季度销售额。当在 Date 分区中计算 RUNNING_AVG(SUM([Sales])) 时，结果为每个季度的销售额值的运行平均值。

```
□
```

示例

RUNNING_AVG(SUM([Profit])) 计算 SUM(Profit) 的运行平均值。

RUNNING_COUNT(表达式) along [维度]

RUNNING_COUNT(表达式) along [维度]

返回给定表达式从分区中第一行到当前行的运行计数

示例

RUNNING_COUNT(SUM([Profit])) 计算 SUM(Profit) 的运行计数。

RUNNING_MAX(表达式) along [维度]

RUNNING_MAX(表达式) along [维度]

返回给定表达式从分区中第一行到当前行的运行最大值。

```
□
```

示例

RUNNING_MAX(SUM([Profit])) along [Date], [Quarter] 计算 SUM(Profit) 的运行最大值。

RUNNING_MIN(表达式) along [维度]

RUNNING_MIN(表达式) along [维度]

返回给定表达式从分区中第一行到当前行的运行最小值。

```
□
```

示例

`RUNNING_MIN(SUM([Profit])) along [Date],[Quarter]` 计算 `SUM(Profit)` 的运行最小值。

RUNNING_SUM(表达式) along [维度]

RUNNING_SUM(表达式) along [维度]

返回给定表达式从分区中第一行到当前行的运行总计。

```
□
```

示例

`RUNNING_SUM(SUM([Profit])) along [Date],[Quarter]` 计算 `SUM(Profit)` 的运行总计。

WINDOW_AVG(表达式, [开始位置, 结束位置]) along [维度]

WINDOW_AVG(表达式, [开始位置, 结束位置]) along [维度]

返回给定表达式从分区中第一行到当前行的运行平均值。

```
□
```

示例

`WINDOW_AVG(SUM([Profit]), FIRST()+1, 0) along [Date],[Quarter]` 计算从第二行到当前行的 `SUM(Profit)` 平均值。

WINDOW_COUNT(表达式, [开始位置, 结束位置]) along [维度]

WINDOW_COUNT(表达式, [开始位置, 结束位置]) along [维度]

返回窗口中表达式的计数。窗口用与当前行的偏移定义。使用 `FIRST()+n` 和 `LAST()-n` 表示与分区中第一行或最后一行的偏移。如果省略了开头和结尾，则使用整个分区。

示例

`WINDOW_COUNT(SUM([Profit]), FIRST()+1, 0) along [Date],[Quarter]` 计算从第二行到当前行的 `SUM(Profit)` 计数。

WINDOW_MAX(表达式, [开始位置, 结束位置]) along [维度]

WINDOW_MAX(表达式, [开始位置, 结束位置]) along [维度]

返回窗口中表达式的最大值。窗口用与当前行的偏移定义。使用 `FIRST()+n` 和 `LAST()-n` 表示与分区中第一行或最后一行的偏移。如果省略了开头和结尾，则使用整个分区。

例如，下面的视图显示每季度销售额。Date 分区中的窗口最大值返回所有日期期间的最大销售额。

```
□
```

示例

`WINDOW_MAX(SUM([Profit]), FIRST()+1, 0) along [Date],[Quarter]` 计算从第二行到当前行的 `SUM(Profit)` 最大值。

WINDOW_MIN(表达式, [开始位置, 结束位置]) along [维度]

WINDOW_MIN(表达式, [开始位置, 结束位置]) along [维度]

返回窗口中表达式的最小值。窗口用与当前行的偏移定义。使用 FIRST()+n 和 LAST()-n 表示与分区中第一行或最后一行的偏移。如果省略了开头和结尾, 则使用整个分区。

例如, 下面的视图显示每季度销售额。Date 分区中的窗口最小值返回所有日期间的最小销售额。

**示例**

WINDOW_MIN(SUM([Profit]), FIRST()+1, 0) along [Date],[Quarter] 计算从第二行到当前行的 SUM(Profit) 最小值。

WINDOW_SUM(表达式, [开始位置, 结束位置]) along [维度]**WINDOW_SUM(表达式, [开始位置, 结束位置]) along [维度]**

返回窗口中表达式的总计。窗口用与当前行的偏移定义。使用 FIRST()+n 和 LAST()-n 表示与分区中第一行或最后一行的偏移。如果省略了开头和结尾, 则使用整个分区。

例如, 下面的视图显示每季度销售额。Date 分区中计算的窗口总计返回所有季度的销售额总计。

**示例**

WINDOW_SUM(SUM([Profit]), FIRST()+1, 0) along [Date],[Quarter] 计算从第二行到当前行的 SUM(Profit) 求和。

WINDOW_VAR(表达式, [开始位置, 结束位置]) along [维度]**WINDOW_VAR(表达式, [开始位置, 结束位置]) along [维度]**

返回窗口中表达式的样本方差。窗口用与当前行的偏移定义。使用 FIRST()+n 和 LAST()-n 表示与分区中第一行或最后一行的偏移。如果省略了开头和结尾, 则使用整个分区。

**示例**

WINDOW_VAR(SUM([Profit]), FIRST()+1, 0) along [Date],[Quarter] 计算从第二行到当前行的 SUM(Profit) 方差。

WINDOW_VARP(表达式, [开始位置, 结束位置]) along [维度]**WINDOW_VARP(表达式, [开始位置, 结束位置]) along [维度]**

返回窗口中表达式的样本方差。窗口用与当前行的偏移定义。使用 返回窗口中表达式的有偏差方差。窗口用与当前行的偏移定义。使用 FIRST()+n 和 LAST()-n 表示与分区中第一行或最后一行的偏移。如果省略了开头和结尾, 则使用整个分区。

示例

WINDOW_VARP(SUM([Profit]), FIRST()+1, 0) along [Date],[Quarter] 计算从第二行到当前行的 SUM(Profit) 方差。

WINDOW_STDEV(表达式, [开始位置, 结束位置]) along [维度]**WINDOW_STDEV(表达式, [开始位置, 结束位置]) along [维度]**

返回窗口中表达式的样本标准差。窗口用与当前行的偏移定义。使用 FIRST()+n 和 LAST()-n 表示与分区中第一行或最后一行的偏移。如果省略了开头和结尾, 则使用整个分区。

示例

WINDOW_STDEV(SUM([Profit]), FIRST()+1, 0) along [Date],[Quarter] 计算从第二行到当前行的 SUM(Profit) 标准差。

WINDOW_STDEVP(表达式, [开始位置, 结束位置]) along [维度]

WINDOW_STDEVP(表达式, [开始位置, 结束位置]) along [维度]

返回窗口中表达式的样本标准差。窗口用与当前行的偏移定义。使用 FIRST()+n 和 LAST()-n 表示与分区中第一行或最后一行的偏移。如果省略了开头和结尾, 则使用整个分区。

示例

WINDOW_STDEVP(SUM([Profit]), FIRST()+1, 0) along [Date],[Quarter] 计算从第二行到当前行的 SUM(Profit) 标准差。

RANK(表达式, ["asc"|"desc"]) along [维度表达式]

RANK(表达式, ["asc"|"desc"]) along [维度表达式]

返回分区中当前行的唯一排名。为相同的值分配相同的排名, 允许并列。使用可选的 "asc" 或者 "desc" 参数指定升序或降序顺序。默认为降序。

示例

RANK(sum([销售额]), "desc") along [部门], 利用此函数, 可以按照部门的销售额求和进行降序排名, 其中的排名有并列且会跳过并列名次, 若销售额分别为(6, 7, 7, 9) 则排名结果为(4, 2, 2, 1)。

RANK_UNIQUE(表达式, ["asc"|"desc"]) along [维度表达式]

RANK_UNIQUE(表达式, ["asc"|"desc"]) along [维度表达式]

返回分区中当前行的唯一排名。为相同的值分配不同的排名。使用可选的 "asc" | "desc" 参数指定升序或降序顺序。默认为降序。

示例

RANK_UNIQUE(sum([销售额]), "desc") along [部门], 利用此函数, 可以按照部门的销售额求和进行降序排名, 且排名不会并列, 若销售额分别为(6, 7, 7, 9) 则排名结果为(4, 2, 3, 1)。

RANK_MODIFIED(表达式, ['asc' | 'desc']) along [维度]

RANK_MODIFIED(表达式, ['asc' | 'desc']) along [维度]

返回分区中当前行的调整后竞争排名。为相同的值分配相同的排名。使用可选的 'asc' | 'desc' 参数指定升序或降序顺序。默认为降序。

示例

利用此函数, 将对值集 (6, 7, 7, 9) 进行排名 (4, 3, 3, 1)。

RANK_DENSE(表达式, ["asc"|"desc"]) along [维度表达式]

RANK_DENSE(表达式, ["asc"|"desc"]) along [维度表达式]

返回分区中当前行的排名。为相同的值分配相同的排名, 允许并列, 但不会跳过并列名次。使用可选的 "asc" 或者 "desc" 参数指定升序或降序顺序。默认为降序

示例

RANK_DENSE(sum([销售额]), "desc") along [部门], 利用此函数, 可以按照部门的销售额求和进行降序排名, 其中的排名有并列但不会跳过名次, 若销售额分别为(6, 7, 7, 9) 则排名结果为(3, 2, 2, 1)。

跨视图粒度计算

刚接触可视化分析平台的用户, 可能会认为平台只是一款数据可视化工具, 但其实平台不单单能可视化数据, 还能对数据进行计算分析, 实现复杂的数据分析需求。

本节要介绍的就是平台提供的众多数据分析功能之一——跨视图粒度计算 (其实就是简单的3个函数, 但能实现复杂的分析需求: 比如新客贡献分析、留存分析、复购分析、RFM分析等等)。

既然要跨视图粒度计算，那我们先来理解两个概念：粒度、视图。

1、粒度：什么是数据粒度呢，它指的是一份数据的细化程度，这么解释有些抽象，可以看下一幅漫画（原漫画来自日和动漫）



漫画中，老板的“粗略”、“明细”描述的就是数据的粒度，越明细的数据包含的信息越多，同时也越难解读，所以一般根据需求不同会挑选不同粒度的数据来分析。

2、视图：数据库技术里有对视图的定义，本文中，可以先简单地把视图理解为平台中的一个图表，如下图所示就是两个不同粒度的视图，视图1 细化到省，视图2细化到城市，视图2的细化程度更高，数据更明细，包含的信息也更多：



好了，现在我们已经理解了粒度、视图两个概念，每个视图有自己的粒度，一个视图里只能有一个粒度。

那什么时候需要用到跨视图粒度计算呢，当我们在一个视图里需要用到来自其他视图粒度的数据时，就需要用到跨视图粒度计算了。举个例子：

例1：还是上文的视图1跟视图2，当我们需要在视图2里加一列数据，显示每个城市销售额在该省的占比，比如计算杭州的销售额占浙江省的比例，这时候我们需要“杭州的销售额/浙江的销售额”，但杭州的销售额数据在视图2里，而浙江的销售额数据在视图1里。没法拿到一起计算：



这时候我们可能需要用SQL去取两个不同粒度视图的数据，然后放在一个视图里进行分析，SQL语句如下所示，如果不了解SQL可以忽略，不影响理解跨视图粒度计算。



但其实在平台里，如果在视图2中，需要用到视图1的数据，只需要借助一个简单的函数就可以实现

创建一个计算字段：

```
{ FIXED [省] : sum([销售额]) }
```

它的意思是以“省”的粒度计算销售额的和，也就得到每个省的总销售额。

得到每个省的总销售额之后，我们可以用城市的销售额除以省的总销售额来计算占比啦，如下：

```
sum([销售额])/sum([各省的销售额])
```

如此，我们便实现了目的，在视图2中增加一列展示每个城市的销售额在该省的占比：



平台里一个函数，就实现了本需要写那么长一串SQL语句才能办到的事，是不是很高效率，不懂SQL的小伙伴也可以掌握。

例2：再举一个跨视图粒度计算的例子来帮大家理解，假设我们有这样一个视图，记录客户每个订单发生的时间，粒度细化到每个订单：



这时候，如果我想给每个订单打个标签，标记这个订单是否是新客户的首单（一个新客户下的第一单就被认为是首单），如图中标出的几个订单：



直观上理解，要判断订单是否是首单，只要把订单的时间跟客户第一次下单的时间进行对比，如果相等，说明这个订单是用户的首单，那么我们可以再建一个客户粒度的视图，对每个客户所有的下单日期取最小值，即可得到每个客户的首单日期：



得到每个客户的首单日期后，我们只要把每个订单的日期去跟首单日期比较，如果相等就是首单，但现在遇到了难题，要比较的两个数据来自不同粒度的视图：



这时候，又可以借助跨视图粒度计算来实现目的啦。

创建一个计算字段：

```
{ FIXED [客户名称] : min([下单日期]) }
```

它的意思是以“客户名称”的粒度来计算最小的下单日期，也就是每个客户首单的日期，然后我们可以通过比较来给每个订单打标签啦。（我们的测试数据里，每个用户每天至多下一单，这样只要两个日期是同一天，就判定为首单，如果一天多单则还需要比较两个日期的时分秒是否一致）

```
if DATEDIFF("day", [下单日期], [首单日期]) = 0 then "首单"  
else "非首单"
```

结果如图所示：



打完标签后，我们就可以在图表中直观地区别出销售额中哪些是由新客贡献的，可以看到新客贡献了大部分销售额，说明我们的老客复购效果可能不理想啊。

全文到这里就结束啦，平台的跨视图粒度计算是不是很灵活高效，有需要的小伙伴快去试试吧～

关于跨视图粒度计算的更多文章：

[感动到流泪！数据分析师的福音：跨视图粒度计算](#)

[深入浅出“跨视图粒度计算”--1、理解数据的粒度](#)

[深入浅出“跨视图粒度计算”--2、INCLUDE表达式](#)

[深入浅出“跨视图粒度计算”--3、EXCLUDE表达式](#)

分析模块

趋势线分析

平台的分析模块支持给图表添加趋势线。

当图表满足以下任一条件时，可以添加趋势线。

条件一：X轴跟Y轴有至少1个连续度量字段，如下方示意图所示。



我们可以在分析模块进行趋势线的设置，如图所示，这里我们选择**多项式**类型趋势线，并将趋势线的度设置为3。



另外我们也可以选择趋势线不受颜色的影响，则会针对所有点计算出1根趋势线，如图所示。



条件二：X轴（Y轴）为连续日期字段，Y轴（X轴）为连续度量字段，如下方示意图所示。



我们在图表的分析模块进行趋势线设置后，效果如图所示。



浏览界面下，鼠标移动至趋势线时，还会出现提示框显示趋势线的相关信息。



聚类分析

在图表控件的分析面板，勾选聚类分析后，将根据kmeans++算法对图表上的数据进行分类，会生成**群集**字段，并放置在颜色属性区，用不同颜色来区分不同群集。



在分析面板，我们可以设置群集的个数、聚类分析时依据的指标。



点击“聚类详情”后，我们可以看到更详细的描述，并可以对群集进行重命名。



完成聚类分析后，可将自动生成的群集pill生成新的自定义维度字段



并可利用该字段绘制其他可视化图表



数据字典

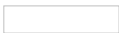
数据字典（修改别名）用于修改离散字段成员的名称，当某个字段成员名称不符合预期的时候可以手动修改，**数据字典（修改别名）**仅针对维度

比如：我想将图表上的字段成员**东山**改成**福建东山**

1. 在schema列表中选择字段成员所属的字段**城市**，右键单击选择**数据字典**。



2. 激活数据字典面板，搜索**东山**这个字段成员。



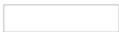
3. 将别名中的**东山**改为**福建东山**。



4. 点击确认后发现schema列表中最底部自定义维度中多了一个新字段**城市_别名**。



5. 将行列面板中的维度字段**城市**替换为**城市_别名**，从图表上来看，**东山**已经改成了**福建东山**。



转换字段类型

维度字段可以被转化为度量字段。在维度列表中右单击需要转化的字段，在出现的弹出菜单中选择**转化为度量**即可。



创建地理字段

有的维度字段中包含了地理信息，但需要我们指定其为地理字段，并编辑相应的地理信息。

1. 单击在维度面板右上方的地图图标，出现**地理位置列表**对话框，如下图所示。



2. 在这里，我们指定**城市**字段为地理字段。在三种映射表中，在城市的下方选择**选择维度**，然后在出现的字段列表中选择“城市”字段。

3. 在对话框的右方会出现一个**数据-定位**的映射表。系统会自动将相应的地理数据映射到其定位，但也有失败的情况，这个时候需要我们自己进行编辑。

比如说，单击**三亚**右方的**无法识别**，在出现的对话框里找到**三亚市**并点击即可。



4. 在编辑完地理映射表之后，单击**确定**，我们可以发现原先的地理位置列表中出现了这条映射。



5. 点击**确定**。我们可以在原先的维度列表中发现**城市**这个字段左边的图标变为了一个地球，这说明我们已经成功完成地理字段的创建。



创建组

我们可以为一个维度字段中所有的成员自定义组别的划分，新的划分会作为一个新的维度字段而存在。

比如，我们右单击**地区**字段，在**创建组(地区)**的对话框中，罗列了该字段下所有的成员，我们可以对这些成员进行分组。

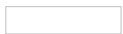
我们首先新建两个组，分别名为**南方**与**北方**，然后再将某个成员添加到对应的组中，如下图所示。



聚合方式

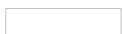
度量默认的聚合方式是**求和**。我们也可以更改某个字段的聚合方式。

比如，右单击度量面板中的**销售额**字段，在弹出菜单中我们发现其默认的聚合方式为“求和”，我们可以选择其他聚合方式，如“平均值”、“计数”、“去重计数”、“最大值”、“最小值”等，单击以确认。



聚合方式支持百分位数

平台的聚合方式还支持百分位计算，操作方式如下方图片所示。



下图则展示了，针对**销售额**度量，不同聚合方式之间的对比。



百分位聚合方式目前只支持部分数据源。其中中位数支持greenplum、oracle、postgresql、impala。百分位数支持greenplum、oracle、postgresql。

数据格式

在使用度量字段制作报表的过程中，我们可以更改度量的数据格式，从而控制报表上该度量的显示方式。



如上图所示，我们已经使用**地区**字段与**销售额**字段制作出了一张柱状图。在柱状图的Y轴上，显示的销售额的格式类似于“3.50M”。我们可以更改数据格式。右单击**求和(销售额)**字段，选择**数据格式**，有如下的对话框出现。



数据格式默认是自动，会选择最合适的格式。我们可以将其自定义为数值或百分比。在数值一栏中，我们可以设置数字小数的位数、数字的单位、数字千分符的显示、以及货币单位的显示。在百分比一栏中，我们可以将数字显示为百分比的形式，并指定其小数的位数。

平台也支持批量设置数据格式，数据模型右侧的更多操作，选中**批量设置数据格式**，在设置面板可以选择需要批量设置的字段及数据类型。



数据桶

可以基于度量创建数据桶，比如下图，基于销售额字段创建数据桶



出现数据桶的编辑页面，可以修改数据桶的名称，可以指定数据桶的大小，默认为10，可以手工修改；工具还会提示当前度量的最小值和最大值，以及差值，供用户在设置桶大小时进行参考。



点击确定后，将在自定义维度中生成一个新的数据桶维度，如下图



向数据面板中添加分桶维度时，每个数据桶都充当一个大小相等的容器，用于对特定的值范围汇总数据。将创建列或行标题，其中每个数据桶标签都指示分配给该数据桶的数字范围的下限。请注意包括下限。比如在图表中，将数据桶字段拖入X轴，将记录数拖入Y轴，就可以看到下图这样的直方图，来看到订单金额区间和数量的分布。



层级

创建层级

在这里，我们想要创建出这样一个层级：国家、地区、城市。

首先，右单击“国家”字段，在弹出菜单中选择“创建层级”，如下图所示

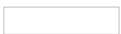


输入新建层级的名称，单击确认。我们可以发现“国家”已经被放入该层级当中。然后，分别右单击“地区”与“城市”字段，单击“加入层级”，可以得到下图所示的层级



上卷下钻

创建层级之后，在我们已经创建的报表中，我们可以发现原先的X轴上的地区字段之前出现了一个加号，如下图所示



进入柱状图本身，我们可以发现当鼠标停留在X轴上时，也会出现一个加号的图标，如下图所示



这是使用上卷下钻的两种方式。无论单击哪里的加号，系统会自动将地区字段所在的层级展开，得到一个新的图表，如下图所示



我们可以使用加号进行数据的下钻，同理，使用减号来进行数据的上卷。

聚焦下钻

聚焦下钻，可以实现用户点击图，下钻到另外一个字段，比如地区下钻到产品类别，出来的是这个地区下面的产品类别的分布数据。

实现方法：

1. 新建层级，具体见创建层级。

创建分析层级，包含地区、类别、子类别。



2. 创建图表，图表的X轴放入带层级的维度地区，Y轴放入度量销售额。



在编辑状态已经完成了聚焦下钻的设置。可以在预览状态使用了。

3. 进入预览界面，鼠标放到图表上，会出现**下钻**开关。

4. 打开**下钻**开关。

5. 点击**华东**的柱子，则会进入华东地区，产品类别的分布数据。

6. 再继续点击**家具**的柱子，则会进入华东地区，家具下，产品子类别的分布数据。

7. 此时，想查看华东地区，办公用品下，产品子类别的分布数据，则先点击返回按钮。

再点击**办公用品**的柱子。

则会进入华东地区，家具下，产品子类别的分布数据。

支持的图形

1. 柱形图、折线图、区域图 层级字段放入X轴或Y轴

2. 普通饼图、环形饼图 层级字段放入颜色

3. 散点图 层级字段放入颜色

4. 透视图 层级字段为轴维度，是最后一个字段

排序

图表的排序

比如用户做了一个简单的柱状图，X轴为类别、Y轴为销售额；现在需要根据销售额降序进行排序

方法一：鼠标移动到Y轴，右键，选择升序 or 降序

方法二：维度下拉菜单中选择排序，可以根据维度的首字母进行排序，也可以根据度量进行排序

方法三：度量Pill下拉菜单中选择排序

选择维度（子类别）按照度量（销售额）的降序

表格的排序

假设用户做了一张如下的表格，需要对销售额进行排序，可以在销售额字段下拉菜单中选择排序（降序），如下图

如果你还想对维度字段（产品类别）进行排序，能够在产品类别字段的下拉菜单中选择排序

排序完成后，你可以看到，首先按照产品类别进行排序，每个类别中的销售额也是按照降序排列

总计

图表控件下画出的图表，在维度字段的下拉列表设置。

表格控件下画出的表格，在度量字段的下拉列表设置。

一、普通图表的总计设置

图表控件下画出的图表

比如：每个地区销售的柱状图

设置总计，即所有地区的销售额

1. 鼠标移动到X轴的维度**地区**，Hover点击下拉菜单，选择**总计**。
2. 选择总计选项中的**列表底端**。

得到总计

二、透视表的总计设置

图表控件下画出的图表

比如：各个年度、各地区、各产品类别的销售额情况的透视表

设置地区的总计，即每个年度下所有地区的销售额

1. 鼠标移动到X轴的维度**地区**，Hover点击下拉菜单，选择**总计**。

得到总计

三、表格的总计设置

表格控件下画出的图表

比如：各个年度、各地区、销售额、利润情况的表格



设置销售额的总计，即所有年度下所有地区的销售额

1. 鼠标移动到列面板中的**销售额**，Hover点击下拉菜单，选择**总计**。

得到总计



参考线

为了辅助分析工作，平台中引用了参考线与参考区间的功能，这一功能的入口在X轴与Y轴的度量数据胶囊的下拉菜单中，如图所示：



参考线的范围会根据数据透视表的范围来画参考线，范围有三种，分别为：全局、单图与单元，它们的意义如下图所示：



参考线、参考范围添加计算值时，有多种聚合方式。例如：百分位、中位数、计数等聚合方式。



参考线既可以是一个常数值，也可以根据一个度量计算的值，比如这里求和(销售额)的平均值。

参考区间的编辑与参考线的编辑类似，只是需要设置两条参考线。

智能推荐

智能推荐是在绘制图表时，智能推荐图表类型

智能推荐用法：

1. 选择**图表**控件，在画布上画出**图表**。



2. 双击图表需要的字段，比如：每个地区中各个类别的销售额情况，涉及到3个字段：维度“地区”，维度“产品类别”，度量“销售额”。



3. 点击工具栏中的**智能推荐**。



4. 选择**智能推荐**点亮的图表类型，其中智能推荐最为推荐的图表是蓝色边框突出。你可选择想要的图表类型，智能推荐将生成对应的图表。



智能问答

当用户进入报告页编辑页面时，在工具栏中会有自然语言的输入入口（当用户没有添加模型时，该功能不可用，需要用户添加数据模型）；



当用户添加完数据模型后，该功能变为启用状态；用户点击后，弹出新的面板，面板上会有输入框，用户可以输入你想问的问题，同时系统会基于当前数据模型，智能的生成一些句子来猜测用户想了解的内容；



当用户输入自然语言提交后，系统会对自然语言进行解析，在该面板的图表展示区自动生成图表；

比如用户输入：



用户可以将图表拖入到页面中，进行后续的编辑；



同时用户也可以直接选择平台推荐的问题，选择后，生成图表；



如果用户输入内容无法匹配到当前的语法，系统会提示用户说当前无法生成图表；并且系统根据用户输入语句中的关键词自动生成几个可能的问句供用户选择；



功能限制：

1、目前产品支持中文输入，但是字段名称要和数据模型中的一致，系统不支持翻译功能；比如用户数据模型中的字段都是英文，例如有以下字段 Region和Sales；此时如果用户输入“每个地区的销售额”，则无法显示；应该要输入“每个Region的Sales”，才能正常显示。

2、暂时不支持即时的反馈，用户一定需要输入完提交后，系统才进行解析。

智能分析

在查看表格时，当鼠标移动至表格区域内，可以发现表格的右上角会出现一个蓝色的按钮，这就是平台的智能分析功能。点击该按钮，平台会自动基于表格的数据进行分析，给出你可能会关心的图表跟分析结论，使用方式如下图所示：



如果正在编辑报告，可以直接将智能分析发现的图表直接添加至画布，并可在该图表的基础上继续编辑。





如果正在浏览报告，则可以将智能分析中的图表导出成图片或添加到智能分析收藏夹，在私人文件夹内可以找到你的智能分析文件夹，这里收录了你曾收藏过的所有智能分析图表，且私人文件夹内的内容只有你个人可见，若需要跟他人共享，需要将报告拷贝或移动至公共文件夹区域。



数据医生

报告整体

(1) 浏览报告时，右上角【数据医生】按钮，点击则开启数据医生的诊断与分析： (2) 编辑报告时，也可点击【数据医生】按钮（即原来智能诊断按钮），对当前编辑的页面进行性能与数据的诊断： 数据医生的诊断分为性能分析、性能诊断、性能统计三部分。

一、性能分析 随着报告的加载，数据医生性能分析的时间轴会动态变化，显示当前报告页的每个组件的查询过程及耗时，用户能够快速定位性能慢的阶段，采取针对性优化措施。

1. 分析结论中会显示当前报告页的组件个数，报告加载完成后有分析结论；

2. 时间轴首先显示整体报告渲染的耗时（从进入报告到加载完成的总耗时）；

3. 报告组件渲染时间展示了各个组件从查询准备到查询、SQL执行、计算、前端渲染等阶段的耗时；

各阶段指标说明如下： 4. 耗时超过20s的组件认为是性能差的组件，对应耗时的柱形会飘红，快速发现性能瓶颈，

从而进行优化： 诊断完成后，诊断结论会根据组件耗时阶段判定性能差的原因，并告知浏览用户联系数仓或平台运维解决性能问题，例如： 二、性能诊断 除了实时的性能分析外，BI平台根据最佳实践从报告优化、报表优化、筛选器优化、模型优化、数据优化五个方面给出性能诊断建议及优化操作，辅助用户创建性能更优的报告给出性能的诊断，完成平台建议的优化措施报告性能可能会更优，具体建议示例：

三、性能统计 根据当前报告页过去30天的响应时长数据统计历史性能趋势，直观展示性能情况。

1. 根据TOP90响应时间（即90分位数的响应时长数据）判定该报告页的性能情况，不同区间的性能等级为： 2. 性能统计的趋势图可切换tab查看不同性能区间的占比情况：

单个组件

（1）对于单个组件，可从图表右侧更多按钮选择“数据医生”，点击后则进入单个组件的诊断： （2）当图表报错时，组件上即有【数据医生】的快速入口，点击后可查看当前图表组件的失败原因、性能分析、异常诊断：

（3）报告数据医生的组件右侧的查询icon点击后亦可查看该组件的性能情况： 单组件的数据医生功能与报告整体类似，提供性能分析、诊断和统计；不赘述。

功能配置

数据医生功能支持编辑报告及浏览报告时使用，若不想让浏览用户使用数据医生功能，则需要项目管理员在项目中心-功能配置中，关闭【浏览态的数据医生功能】：