永洪创建数据集 Z-Data Modeler

用户使用手册

版权声明

免责声明

通讯方式

北京永洪商智科技有限公司

北京市朝阳区光华路 9 号光华路 SOHO 二期 C 座 9 层（100020）

电话：(86-10)-58430919

邮箱：public@yonghongtech.com

网站：http://www.yonghongtech.com

目录

第 1 章 : Z-Data Modeler 简介

Z-Data Modeler 是 Yonghong Z-Suite 的核心模块之一，作为软件的第一级接口与数据源相连接，将原始数据按照客户的需求进行筛选和优化，为后续数据分析操作提供输入。由于当前的数据类型、数据结构、数据范围和数据库类型各种各样且纷繁复杂，不能够直接用于数据分析，必须按照用户需求和 Yonghong Z-Suite 的规范进行过滤，才能成为系统可以识别的数据，才可以直接用于数据分析。因此 Data Modeler 目的就是提供各种不同的条件，严格筛选数据，输出适合的数据给其他功能模块使用。

Z-Data Modeler 包括创建数据集和数据集引擎两个主要模块。用户通过创建数据集定义各种数据集条件，为后续的数据分析提供分析模型和输入准则。当大量的原始数据进入到 Data Modeler 以后，数据集引擎就根据设定好的条件对于数据进行排列组合，生成数据集列表，为以后的数据分析提供服务。针对于不同用户对于数据集的需求和自身所能提供的技术的复杂度， Data Modeler 提供了各种不同级别的数据集。

创建数据集

创建数据集是使用 Yonghong Z-Suite 的开始，通过定义各种不同的数据集条件，将原始数据转换成为系统所需的数据类型和数据模式，为后续操作提供各种输入。

创建数据集模块目前支持九种数据数据集，通过定义数据集条件，连接各种不同的数据库类型，提供各种数据过滤服务，产生最终的数据集表单。

数据建模奠定了数据块技术的基础，让您操作异构数据源时使用统一的数据块的数据。

您将使用创建数据集来连接到不同的数据源，创建数据集，并确定语义层。这些数据集和语义层（逻辑模型），您可以直接使用，或者可以进一步在创建数据集中创建复杂的，组合的原子数据块。

创建数据集，可以访问关系数据库和文件。数据库包括数据仓库，数据集市，大型机。文件包括电子表格。

启动创建数据集

1）点击 Yonghong Z-Suite 产品的启动快捷方式。

2）打开浏览器，然后在地址栏中输入 http://hostname:8080/bi/Viewer, 登陆到客户端。这里的 hostname 是您的机器名，如果是本机访问，可以用 localhost。8080 是默认端口号，如果在安装产品时修改了默认的端口号，请采用正确的端口号。

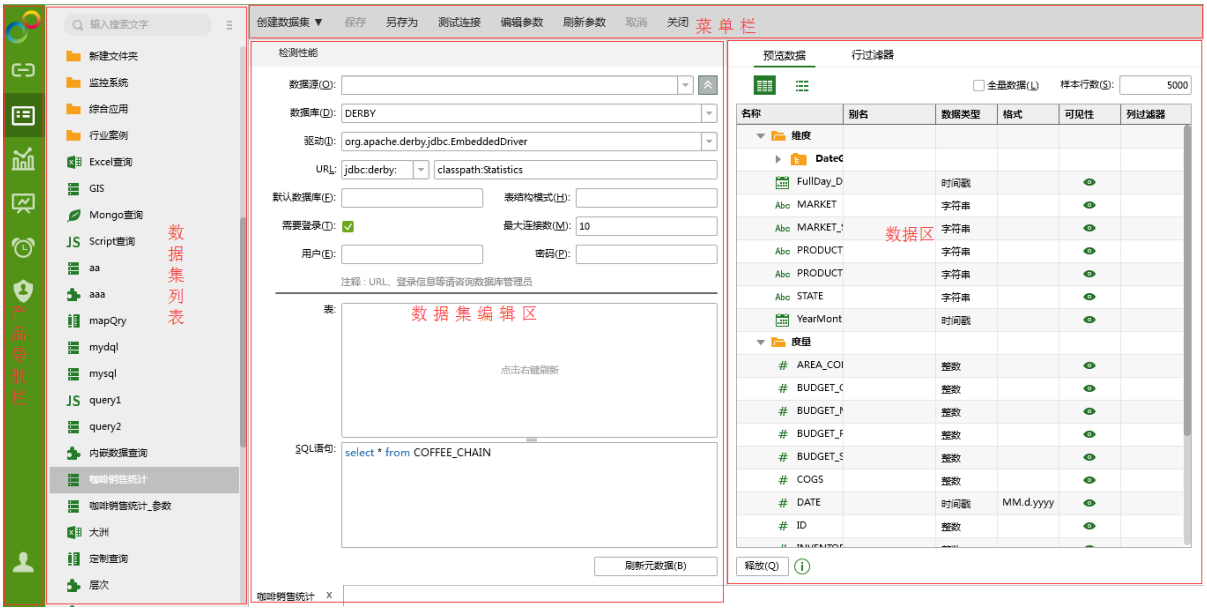
3）输入用户名和密码后登陆到主页面

4）点击创建数据集按钮后，进入到创建数据集的界面。

打开创建数据集，就可以开始创建数据集，来创建数据集源和搭建数据模型，以供制作报告使用。

创建数据集界面

创建数据集界面左边是产品导航栏，数据集界面整体分为四部分，菜单栏、数据集列表、数据集编辑区、元数据区。



菜单栏

创建数据集

点击此按钮，弹出下拉菜单，用户可创建不同类型的数据集。本产品提供九种类型的数据集：SQL 数据集，Mongo 数据集，Excel 数据集，组合数据集，Script 数据集，数据集市数据集，内嵌数据集，定制数据集，自服务数据集。

保存

当用户保存新建的数据集时，会弹出另存为对话框，用户可设定保存路径以及数据集的名称。当用户打开已经存在的数据集后，对该数据集进行修改，可点击保存按钮直接保存。

另存为

另存已打开的数据集。

测试连接

测试当前数据集中的数据库是否连接成功。

编辑参数

参数，也叫参变量，是一个变量。可通过编辑参数对话框对参数进行添加、删除和收集。编辑参数对话框如下图所示。

【添加 / 删除】用户点击添加按钮时，弹出名称编辑对话框，用户可设定参数的名称。在设定好参数后，不支持再对此参数进行重命名操作。如果是收集到的参数，则在此对话框中不能删除此参数，但可以对此参数进行编辑。当被收集的参数在初始设置的地方被删除后，则在此参数对话框中该参数处于可被删除的状态。

【信息】显示参数的被引用的位置。

【类型】用户设定该参数的数据类型。

【默认】用户可设定参数的默认值。用户可设定单个值、多个值或者空值。当用户选择多个值时，用逗号分开输入多个值。

【弹出】当勾选此项时，用户在刷新元数据、测试连接、预览数据集或在编辑器中打开此数据集时，会弹出参数值输入对话框，如下图所示。在此对话框中存在设定的默认值。

【参与报表 “ 参数过滤 ” 的过滤策略】勾选 参与报表 “ 参数过滤 ” 的过滤策略 后，自定义的参数会受仪表盘属性里的参数未选值策略的控制。详见永洪编辑报告手册。

注意事项：

如在 SQL 数据集中的 SQL 筛选语句为：

select * from COFFEE_CHAIN where MARKET_SIZE=?{market_size}

在参数对话框中设置收集到的参数 market_size 为弹出状态，点击预览数据集按钮，不输入参数值，直接点击确定按钮，则报错，不能传递空参数值。倘若用户想点击确定按钮时能够刷出数据来，则需要更改 SQL 语句为：

select * from COFFEE_CHAIN<market_size>where MARKET_SIZE=?{market_size}</market_size>

【参与报表“参数过滤”的过滤策略】报表中针对参数过滤都未选择的情况，可以设置默认是空数据集还是全部数据。在编辑参数中勾选此选项后在报表中也会应用相应的未选值策略，不勾选则不会应用。

【可选值】用户可在已有的数据集中选择一数据集，选择一数据段作为标签，一数据段作为值。如果参数不勾选 【弹出】，则直接将这个数据集对应的值传递给这个参数。如果参数勾选 【弹出】，则在弹出参数值输入对话框时，其数据将会以指定的方式显示。本产品提供四种显示方式，选择框、列表、复选框、单选框。

假设用户设定以选择框的形式弹出参数值输入框，用户在点击刷新元数据按钮、点击测试连接按钮、点击预览数据集按钮、在编辑器中打开此数据集时，参数值输入对话框中显示默认的数据集值，如下图所示。

在编辑参数中添加的参数也支持手动输入参数值。如图所示：

在编辑参数的对话框中：

【类型】用户可设定当前参数的数据类型。

【可选值】在可选值中选择内嵌数据，右侧按钮则显示为录入数据，点击录入数据，进入录入数据对话框。如图所示：

【添加】点击右侧的“添加”按钮，在值和标签栏里输入数据。

【删除】选中数据行，点击“删除”按钮，可删除输入的数据。

【上移 / 下移】选中数据行，点击上移或下移，可改变数据的上下位置。

【确定】点击“确定”，确定录入的数据。

【取消】取消录入数据。

【应用】录入数据应用显示在组件上。

在对话框中录入数据，其中值不允许为空，标签允许为空，如图所示：

假设用户设定以列表的形式弹出参数值输入框，用户在点击刷新参数按钮、在编辑报告中再次打开此即席数据集报表仪表盘时，参数值输入对话框中显示录入的数据，如下图所示：

如果录入标签，则显示标签；如果没有录入标签，则显示值。

刷新参数

对设置了弹出状态的参数重新输入参数值。如存在一参数 a, 处于弹出状态，并且存在默认值 12，如下图所示。

当用户点击刷新参数按钮时，也将会把此参数的默认值刷出来，如下图所示。

取消

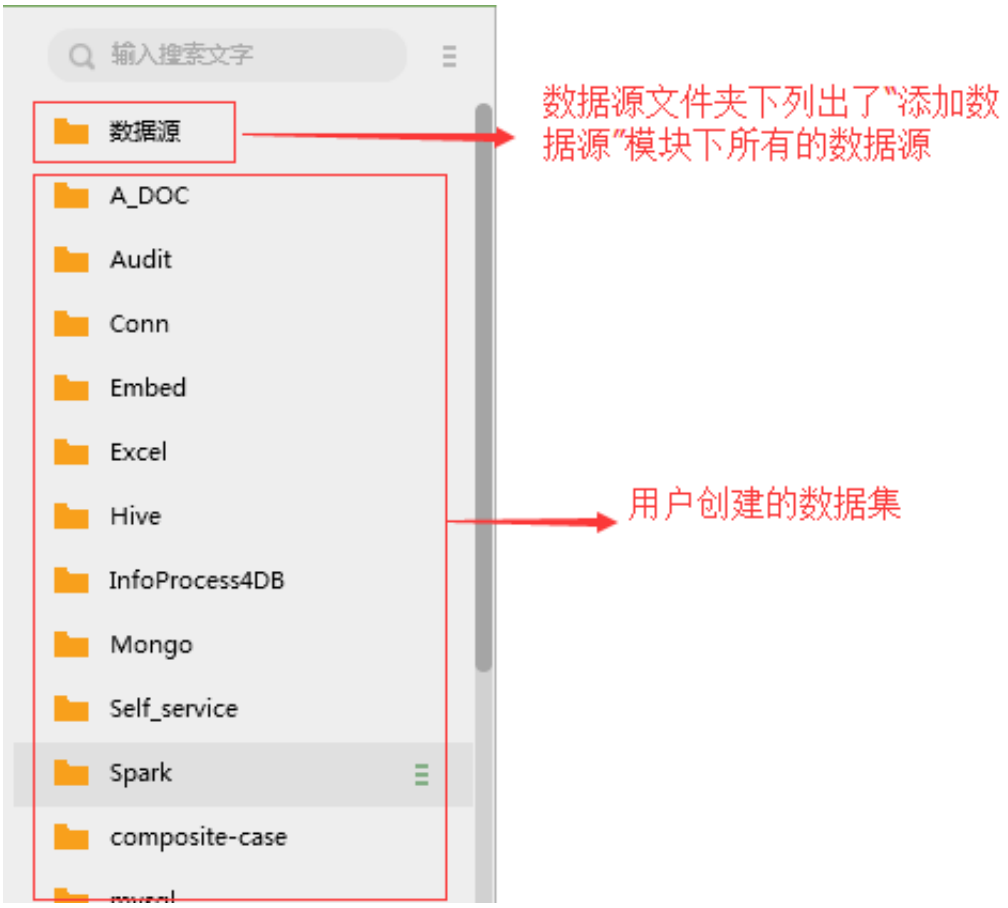
当某一操作长时间未响应时可点击此按钮取消当前操作

关闭

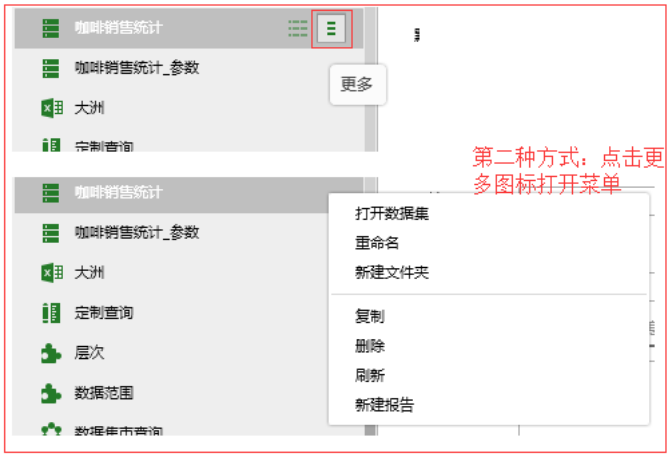
点击此按钮，可关闭当前的数据集。倘若用户尚未保存对当前数据集的修改，将会弹出提示对话框。

数据集列表

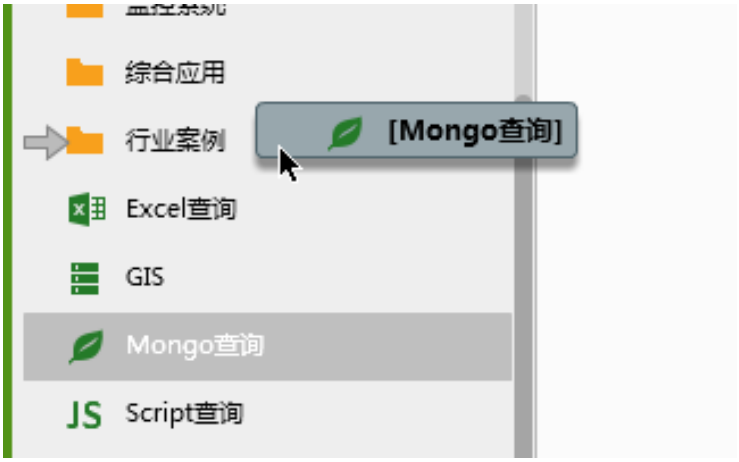
数据集列表分为两部分，一是数据源，二是用户所创建数据集。



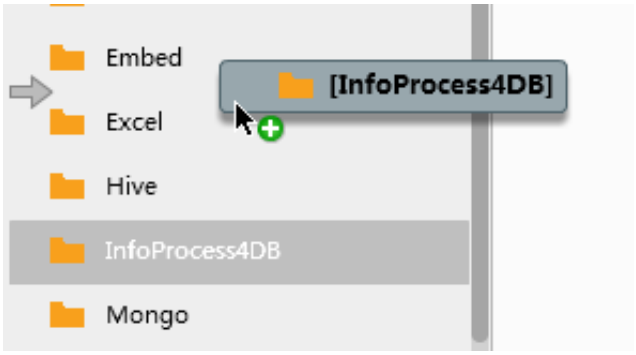
用户可以创建文件夹、对已有数据集重命名、移除等。有两种实现方式 1. 选中内容，点击鼠标右键 2. 点击内容上的更多图标，如下图所示



在数据集列表中支持鼠标的拖拽来更改数据集位置。如下图所示，用户通过鼠标拖拽来把数据集 “Mongo 查询 ” 移动到 “ 行业案例 ” 文件夹中。



在 bi.properties 中配置属性 manual.sort.repository=true，数据集列表中支持鼠标的拖拽来更改数据集、数据源、文件夹之间的排序。如下图所示，用户通过鼠标拖拽来把文件夹 “InfoProcess4DB” 移动到文件夹 “Embed” 和 “Excel“ 之间。



搜索数据集

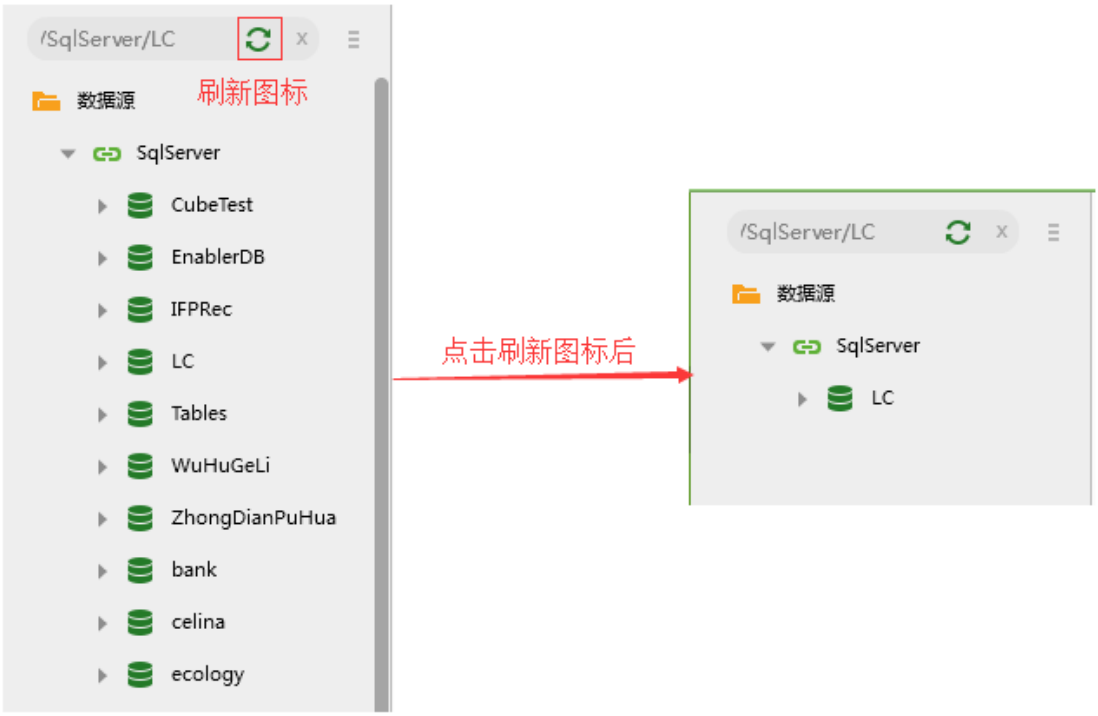
根据输入的文字搜索名字中包含此文字的文件夹，数据集等。

数据源搜索

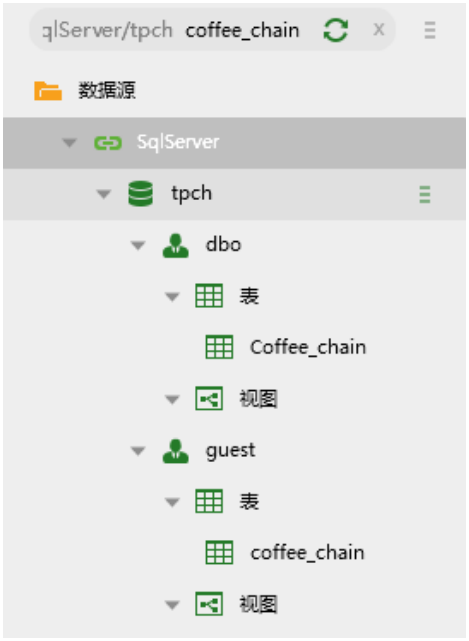
点击数据源搜索图标，进入搜索。

数据源搜索的用法以 sqlserver 数据库为例。点击搜索图标后，展开数据源，点击选择默认数据库

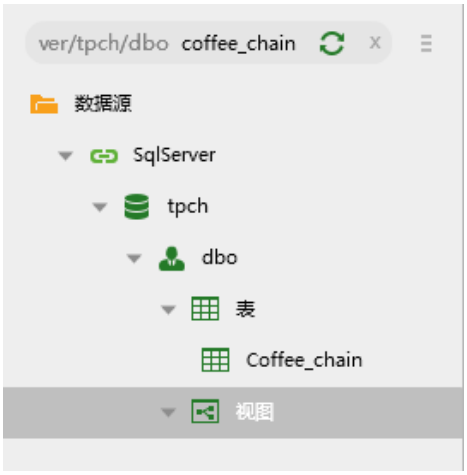
tpch，点击刷新图标后，如下图所示。



输入搜索内容 coffee_chain，展开节点，搜索内容如下图所示。



选中表结构模式 dbo，点击刷新图标，内容如下图所示。



点击刷新图标旁的清除图标，返回数据集列表。如果数据源搜索之前对数据集列表进行了搜索则会返回搜索后的列表。

打开数据源

在数据集列表区域，选中数据源文件夹下的数据源，右键或点击更多图标选择打开数据源选项，在“添加数据源”模块打开选中的数据源。

打开数据集

用户在数据集列表区域，右键或点击更多图标选择打开数据集选项来打开选中的数据集。

创建数据集

用户在数据集列表区域，选中数据源文件夹下的数据源，右键或点击更多图标选择创建数据集选项来新建数据集。

重命名

对已存在的数据集、数据源或文件夹进行重命名。用户首先选中需要重命名的数据集或文件夹，然后右键选择重命名选项即可。

新建文件夹

用户在数据集列表区域右键选择新建文件夹选项来创建文件夹。

复制 & 粘贴

对已存在的数据集或文件夹进行复制。重名时，名字后面自动加后缀“_ 副本”。当用户复制文件夹，在此文件夹下面的数据集也一并被复制。

删除

移除列表中用户不需要的数据集或文件夹。当用户移除文件夹后，在此文件夹中的数据集也将一并被移除。

刷新

刷新当前的数据集列表。

新建报告

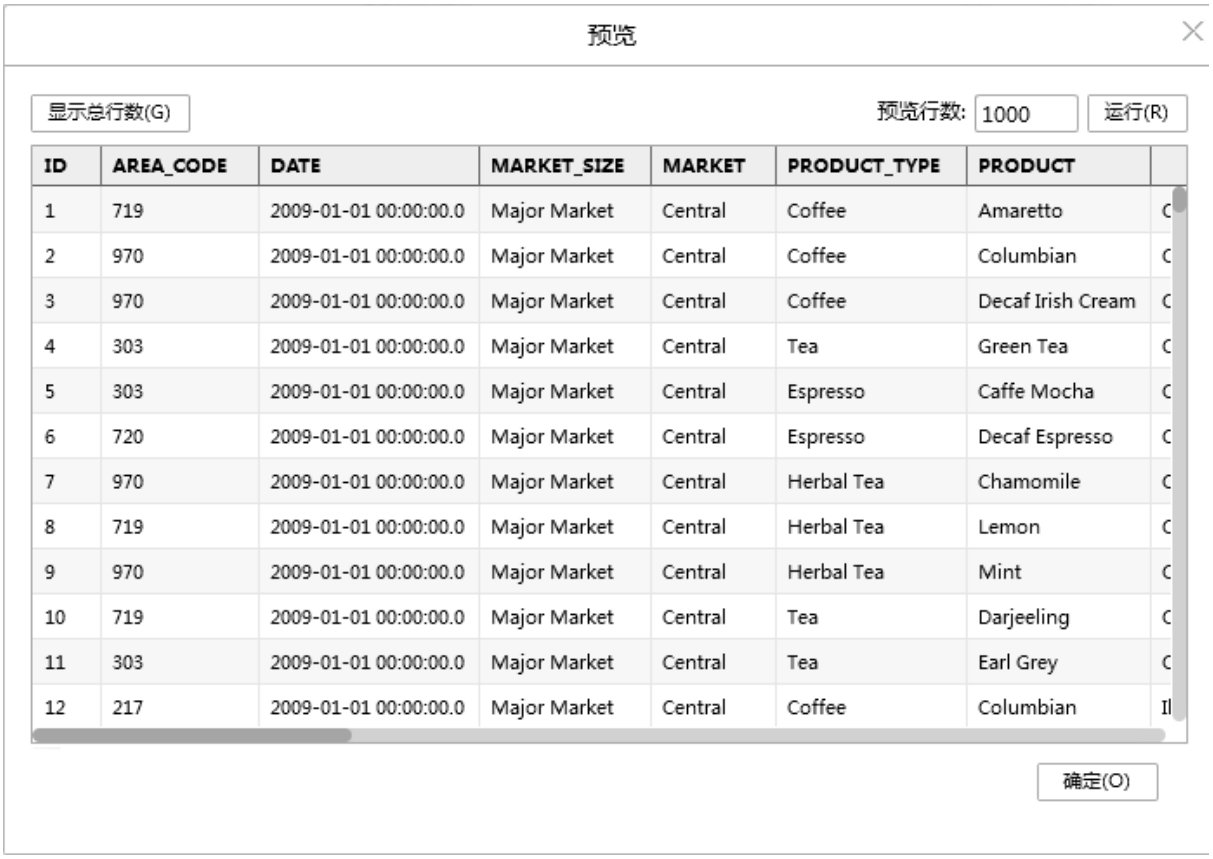
选中数据集或数据源下的表、视图，点击新建报告能够在制作报告模块打开新建页面，绑定的资源树上显示绑定的数据集、表或视图。

预览数据集

当鼠标悬停在数据集列表的某一列时，会在该列的右边显示“预览数据集”的按钮，如下图所示

当鼠标移动到“预览数据集”按钮时，按钮的底部显示灰色的边框，并显示 tooltip：预览数据集，如下图所示：

点击“预览数据集”按钮后，可预览选中的数据集，预览对话框如下图所示：

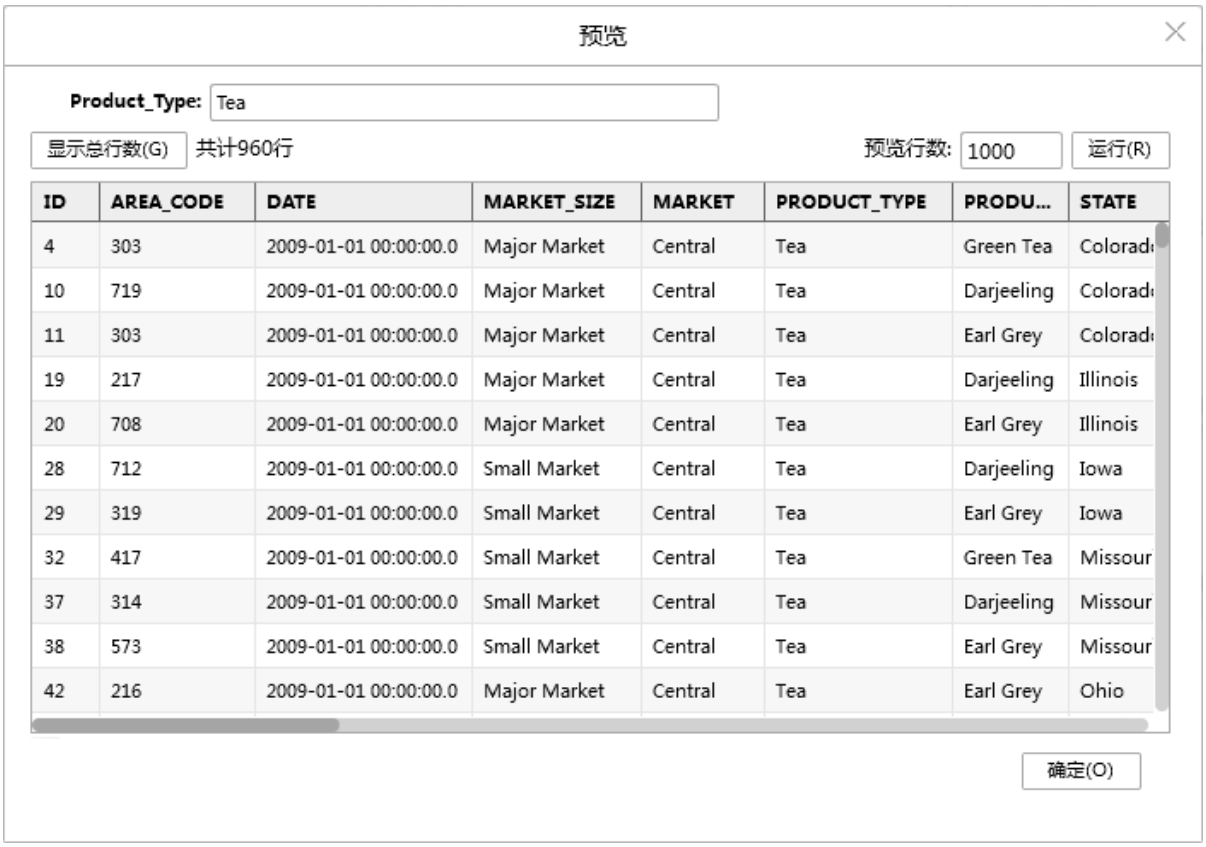


ID	AREA_CODE	DATE	MARKET_SIZE	MARKET	PRODUCT_TYPE	PRODUCT
1	719	2009-01-01 00:00:00.0	Major Market	Central	Coffee	Amaretto
2	970	2009-01-01 00:00:00.0	Major Market	Central	Coffee	Columbian
3	970	2009-01-01 00:00:00.0	Major Market	Central	Coffee	Decaf Irish Cream
4	303	2009-01-01 00:00:00.0	Major Market	Central	Tea	Green Tea
5	303	2009-01-01 00:00:00.0	Major Market	Central	Espresso	Caffe Mocha
6	720	2009-01-01 00:00:00.0	Major Market	Central	Espresso	Decaf Espresso
7	970	2009-01-01 00:00:00.0	Major Market	Central	Herbal Tea	Chamomile
8	719	2009-01-01 00:00:00.0	Major Market	Central	Herbal Tea	Lemon
9	970	2009-01-01 00:00:00.0	Major Market	Central	Herbal Tea	Mint
10	719	2009-01-01 00:00:00.0	Major Market	Central	Tea	Darjeeling
11	303	2009-01-01 00:00:00.0	Major Market	Central	Tea	Earl Grey
12	217	2009-01-01 00:00:00.0	Major Market	Central	Coffee	Columbian

【预览行数】预览数据集时所显示的行数。默认为 1000 行。预览行数可以修改，修改后，点击 ” 运行 ” 按钮，则可按照用户设置的预览行数进行显示。

【显示总行数】点击”显示总行数”按钮后，会在此按钮的后面显示出所预览数据集的总行数。如下图所示：

倘若用户设定了弹出参数，则在预览对话框中存在参数项，用户可根据需求来设定参数值。



数据集编辑区

不同类型的数据集，数据集编辑区界面各不相同。各类数据集的详细介绍见下文。

数据区

数据区包括预览数据和行过滤器两部分，预览数据又分元数据和细节数据。

在预览数据的元数据区可显示用户需要进行统计的所有数据段名称，用户还可自定义数据段等。元数据区如下图所示：

同步数据集数据

同步数据集数据，用于将所创建的数据集进行同步。请参考同步数据集数据章节中的介绍。

全量数据

当勾选全量数据时，样本行数处于置灰状态。用户在编辑报告中可以对数据集的全部数据进行编辑，在编辑模式下绑定框的左上角显示“ 全量数据 ”。

样本行数

用户可设定采集样本数据的行数。假设用户设定为 5000 行，则用户在编辑报告中只能对前 5000 行数据进行编辑，在编辑模式下绑定框的左上角显示“ 数据样本行数 5000”。

名称

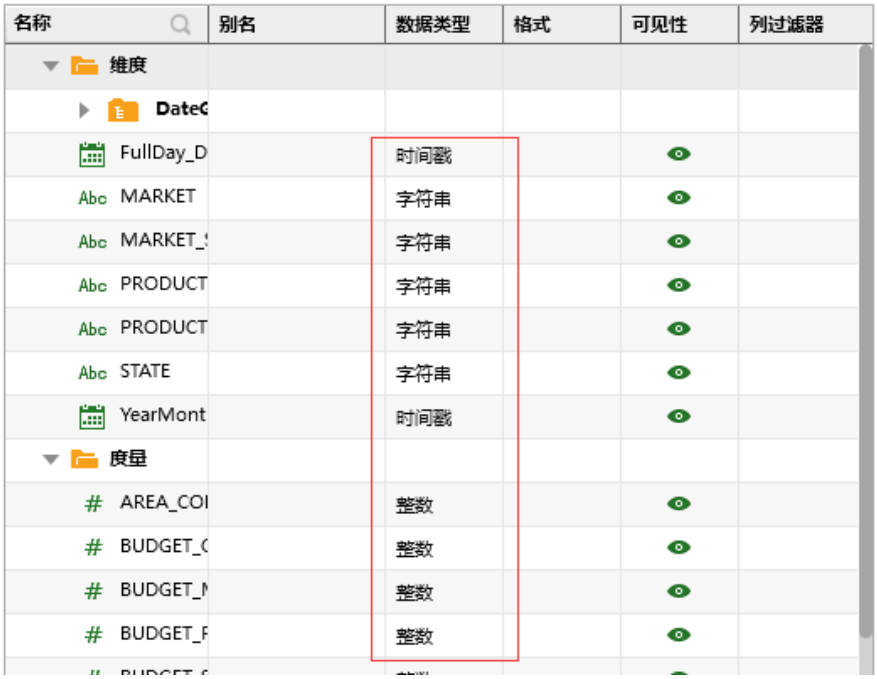
在名称列数据段被分为两组，一组是作为维度，一组是作为度量。默认字符、字符串、布尔、字节类型的数据段被划分到维度目录下。其余的类型数据段被划分到度量目录下。对于用户自定义的数据段：日期型层次、日期型列、数据范围、分析算法、自循环列默认存放在维度目录下，其他类型的数据段根据数据类型划分。

别名

用户可给数据段设置别名，在编辑器中显示的是该数据段的别名。

数据类型

用户可以在部分数据集（如内嵌数据集、 Excel 数据集等）中修改相应数据段的数据类型。



格式

可编辑数据集中列的格式，在编辑报告中可按照设置的格式显示。当单击格式与列的交叉处时，将会显示提示文字：编辑，单击编辑，则会弹出格式对话框，如下图所示。

可通过格式对话框中的选项设置列的格式。

例如：将时间戳类型的列：DATE, 设置成日期类型的格式，则需要在弹出的格式对话框中选择日期选项，设置后在格式中显示日期的格式，如下图所示：

名称	别名	数据类型	格式	可见性	列过滤器
维度					
度量					
# AREA_COI		整数			
# BUDGET_C		整数			
# BUDGET_M		整数			
# BUDGET_P		整数			
# BUDGET_S		整数			
# COGS		整数			
# DATE		时间戳	MM.d.yyyy		

在编辑报告中绑定在表格上后，DATE 的格式为日期类型的，如下图所示：

咖啡销售统计
DATE
01.1.2009
01.1.2009
01.1.2009
01.1.2009
01.1.2009
01.1.2009
01.1.2009
01.1.2009
01.1.2009
01.1.2009

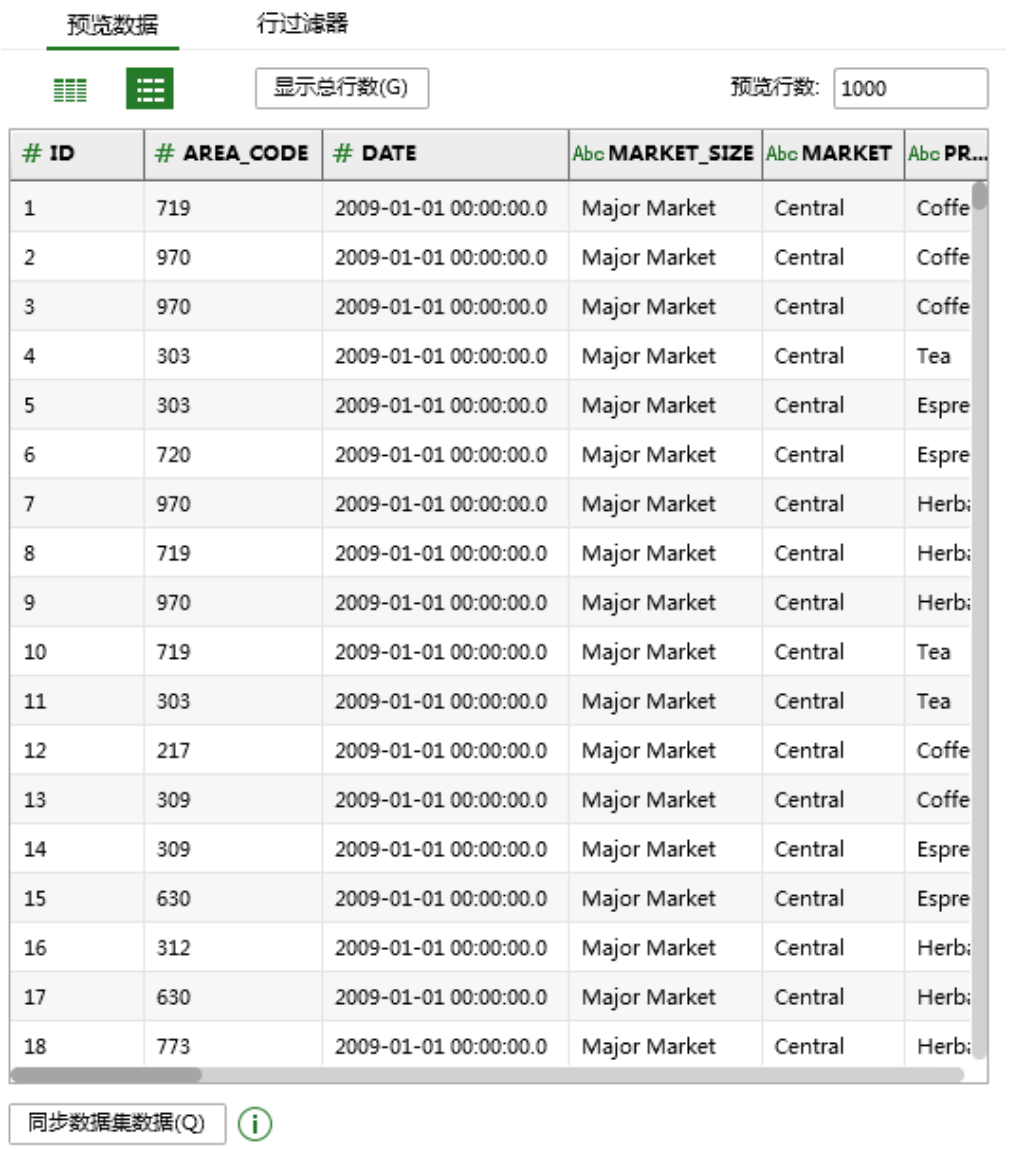
可见性

可见，用于设置数据集中列的可见性。请参考 VPM 章节中的介绍。

列过滤器

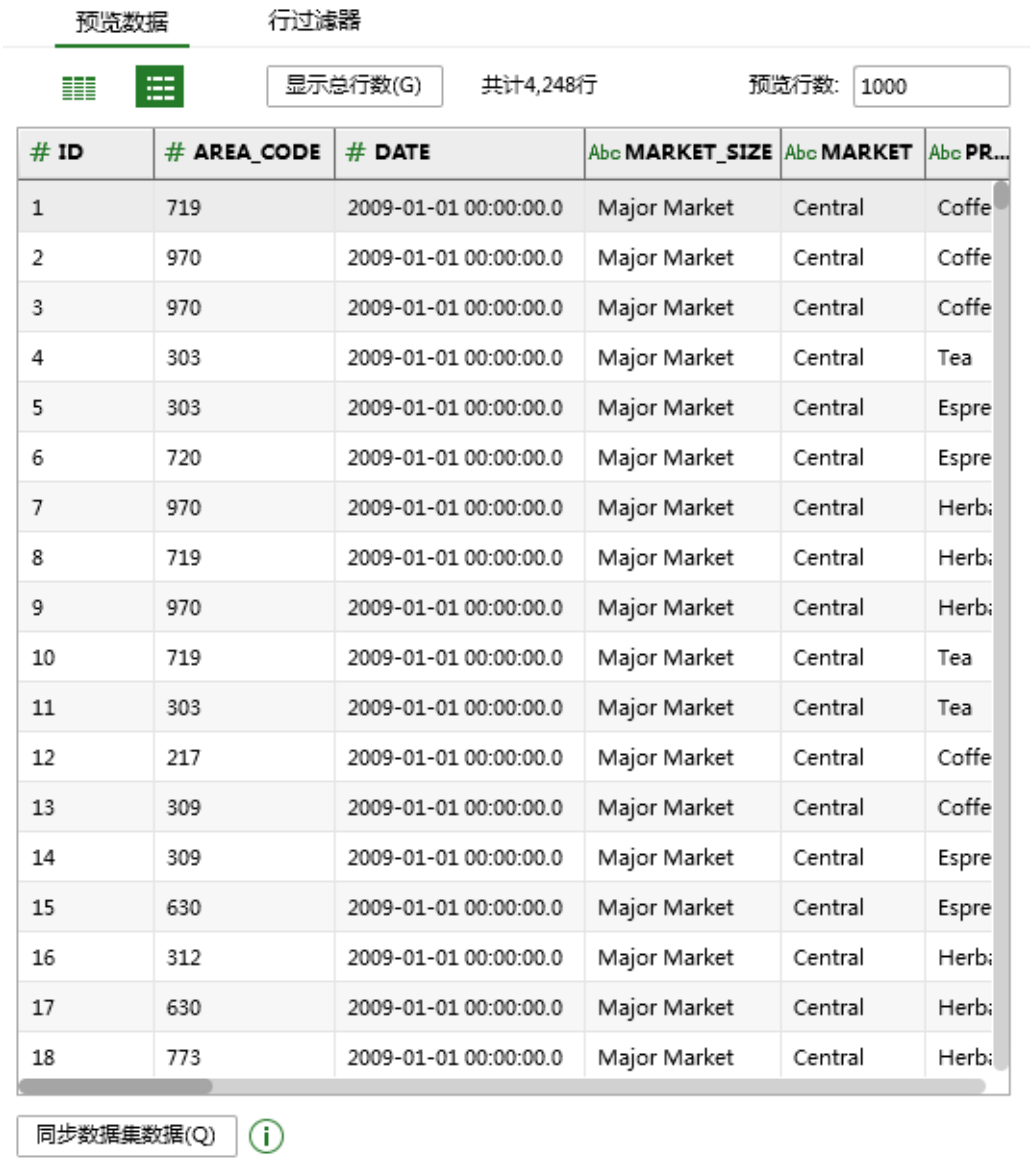
列过滤器，用于设置对某一用户，角色，组的过滤。请参考 VPM 章节中的介绍。

在预览数据的细节数据区用户可以预览当前数据集元数据区中的数据，如下图所示，也可以通过数据集列表中的“预览数据集”按钮打开对应数据集的预览数据。



【预览行数】预览数据集时所显示的行数。默认为 1000 行。预览行数可以修改，修改后，点击 ” 运行 ” 按钮，则可按照用户设置的预览行数进行显示。

【显示总行数】点击”显示总行数”按钮后，会在此按钮的后面显示出所预览数据集的总行数。如下图所示：



# ID	# AREA_CODE	# DATE	Abc MARKET_SIZE	Abc MARKET	Abc PR...
1	719	2009-01-01 00:00:00.0	Major Market	Central	Coffe
2	970	2009-01-01 00:00:00.0	Major Market	Central	Coffe
3	970	2009-01-01 00:00:00.0	Major Market	Central	Coffe
4	303	2009-01-01 00:00:00.0	Major Market	Central	Tea
5	303	2009-01-01 00:00:00.0	Major Market	Central	Espre
6	720	2009-01-01 00:00:00.0	Major Market	Central	Espre
7	970	2009-01-01 00:00:00.0	Major Market	Central	Herb
8	719	2009-01-01 00:00:00.0	Major Market	Central	Herb
9	970	2009-01-01 00:00:00.0	Major Market	Central	Herb
10	719	2009-01-01 00:00:00.0	Major Market	Central	Tea
11	303	2009-01-01 00:00:00.0	Major Market	Central	Tea
12	217	2009-01-01 00:00:00.0	Major Market	Central	Coffe
13	309	2009-01-01 00:00:00.0	Major Market	Central	Coffe
14	309	2009-01-01 00:00:00.0	Major Market	Central	Espre
15	630	2009-01-01 00:00:00.0	Major Market	Central	Espre
16	312	2009-01-01 00:00:00.0	Major Market	Central	Herb
17	630	2009-01-01 00:00:00.0	Major Market	Central	Herb
18	773	2009-01-01 00:00:00.0	Major Market	Central	Herb

倘若用户设定了弹出参数，则在预览细节数据前会弹出参数框，用户可根据需求来设定参数值。

数据集引擎

数据集引擎是 Z-Data Modeler 数据数据集，分析和运行的核心模块。用户通过数据集将数据导入 Yonghong Z-Suite 中，通过可视化的操作界面对数据进行各种操作。而数据集引擎作为后台处理平台，执行这些数据集请求，并将结果返回给前台。

第 2 章：数据集种类

创建数据集模块目前支持九种数据数据集。通过定义的数据集条件，连接不同的数据库，通过各种数据过滤，产生最终的数据集表单。

现支持九种数据集包括：SQL 数据集， Excel 数据集， Script 数据集，定制数据集，数据集市数据集，内嵌数据集，组合数据集， Mongo 数据集，自服务数据集。

数据源可以连接多种数据库：
ORACLE,DB2,SQLSEVER,MYSQL,DERBY,INFORMIX,POSTGRESQL,SYBASE,ACCESS,VERTICA,HIVE,SPARK,PRESTO,IMPALA,HBASE, MONGO 以及本产品的数据库 DATA MART。

以下三种方式进入创建数据集页面：

· 第一种，首页引导区，创建数据集快速入口，可快速进入 SQL 数据集页面、 Excel 数据集页面、自服务数据集页面。

· 第二种，导航栏 -> 创建数据集，打开创建数据集导航页面后，点击相应的数据集进入创建页面。

· 第三种，如果您有已打开的数据集页面，通过菜单栏 -> 创建数据集，点击相应的数据集进入创建页面。。

SQL 数据集

各种实体以及实体之间的各种联系均可用关系模型来表示。关系模型就是指二维表格模型，因而一个关系型数据库就是由二维表及其之间的联系组成的一个数据组织。关系数据库是建立在关系模型基础上的数据库，借助于集合代数等数学概念和方法来处理数据库中的数据，是目前数据库存储类型中的常用类型。当前主流的关系型数据库有 ORACLE, DB2, SQLSERVER， MYSQL, DERBY, INFOMIX, SYBASE, ACCESS, POSTGRESQL, INFOBANK,HIVE,SPARK,PRESTO,IMPALA 等。

标准数据查询语言 SQL 一种介于关系代数与关系演算之间的结构化查询语言，包括查询、操纵、定义和控制，是一个通用的、功能极强的关系性数据库语言，同时又是一种高度非过程化的语言，只要求用户指出做什么而不需要指出怎么做。SQL 集成实现了数据库生命周期中的全部操作，提供了与关系数据库进行交互的方法，它可以与标准的编程语言一起工作。

SQL 数据集主要是通过 JDBC 和 ODBC 与目前主流数据库相联系。 JDBC 和 ODBC 提供了一组对数据库访问的标准 API，建立了一组数据库访问的规范，支持 SQL 语句的执行，同时也是 Yonghong Z-Suite 与数据源之间的主要接口。 Yonghong Z-Suite 对数据库的操作不依赖任何 DBMS，不直接与 DBMS 打交道，所有的数据库操作由对应的 DBMS 的数据库驱动程序完成。

除了一些主流关系型数据库，用户还可以通过选择 GENERIC 类型和输入正确的数据驱动接口，链接到其他类型的数据库，比如一些非关系型的数据库类型 VERTICA, EXADATA, NETEZZA, TERADATA, SYBASE, GREENPLUM 等。

创建 SQL 数据集

1）点击 Yonghong Z-Suite 产品的启动快捷方式。

2）打开浏览器，然后在地址栏中输入 http://hostname:8080/bi/Viewer, 登陆到客户端。这里的 hostname 是你的 IP 地址，如果是本机访问，可以用 localhost。 8080 是默认端口号，如果在安装产品时修改了默认的端口号，请采用正确的端口号。

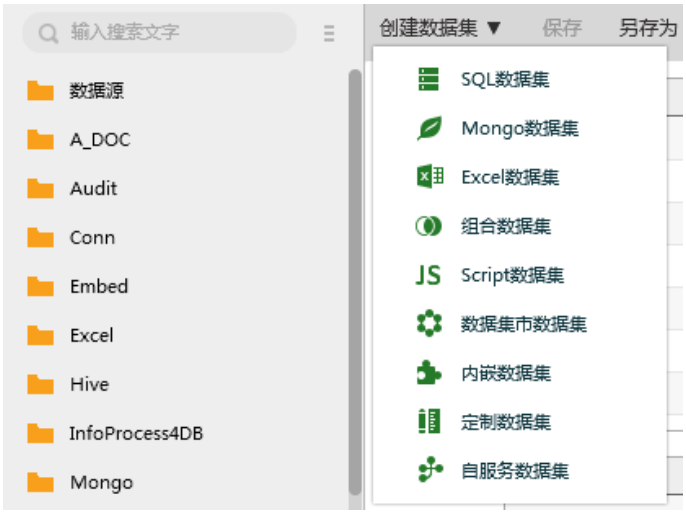
3）输入用户名和密码后登陆到主页面。

4）通过以下两种方式进入创建 SQL 数据集界面：

· 第一种，单击导航栏 -> 创建数据集，在新打开的页面上选择 “SQL 数据集 ”。

· 第二种，如果您有已打开的数据集界面，单击菜单栏 -> 创建数据集 ->SQL 数据集。

5）用户在数据集界面上，点击菜单栏上的新建数据集按钮，选择 SQL 数据集，即可打开 SQL 数据集界面。



也可以点击首页上的 SQL 数据集，进入 SQL 数据集新建界面。

SQL 数据集界面

在打开的界面中，用户可设定数据库类型、驱动、 URL 等，并可编辑使用 SQL 语句，还能够对 SQL 数据集的性能进行检测。

【数据源】当用户选择自定义数据源时，可配置相应的数据源。用户也可以从已有的数据源列表中选择已经创建好的数据源。

【数据库】用户可设定该数据集中使用的数据库的类型，包括 ORACLE、 DB2、 ACCESS、 VERTICA 等多种类型。 Yonghong Z-Suite 产品提供的数据库为 DATA MART。

【URL】所选数据库的 URL 地址。

【驱动】用户可选择需要的驱动类型，例如 ACCESS 的驱动为 sun.jdbc.odbc.JdbcOdbcDriver。

Yonghong Z-Suite 产品提供的驱动类型为 g5.dc.jdbc.GDriver

【需要登录】当数据库设定了访问权限后，用户需要勾选此项，使用用户名和密码来访问当前数据库。

【表】当用户点击刷新按钮后，在列表中列出了当前数据库中存在的所有表。右键刷新后可以对数据源中表、视图、存储过程进行搜索。详细用法见简介中数据源搜索部分。

倘若是使用 Yonghong Z-Suite 提供的驱动和数据库，则在此列表中列出了所有的数据集。

【SQL 语句】用户输入脚本语句，实现对数据库中数据的数据集。

在使用 ACCESS 数据库时，刷出的表名称的后边有 $ 字符，用户在写 SQL 语句时需要使用双引号把该表引起来，如 select Sales from "Coffee_chain$"。在使用 Yonghong Z-Suite 产品提供的驱动及数据库时，在编写 SQL 语句时需要遵循本产品的 SQL 语句法则。

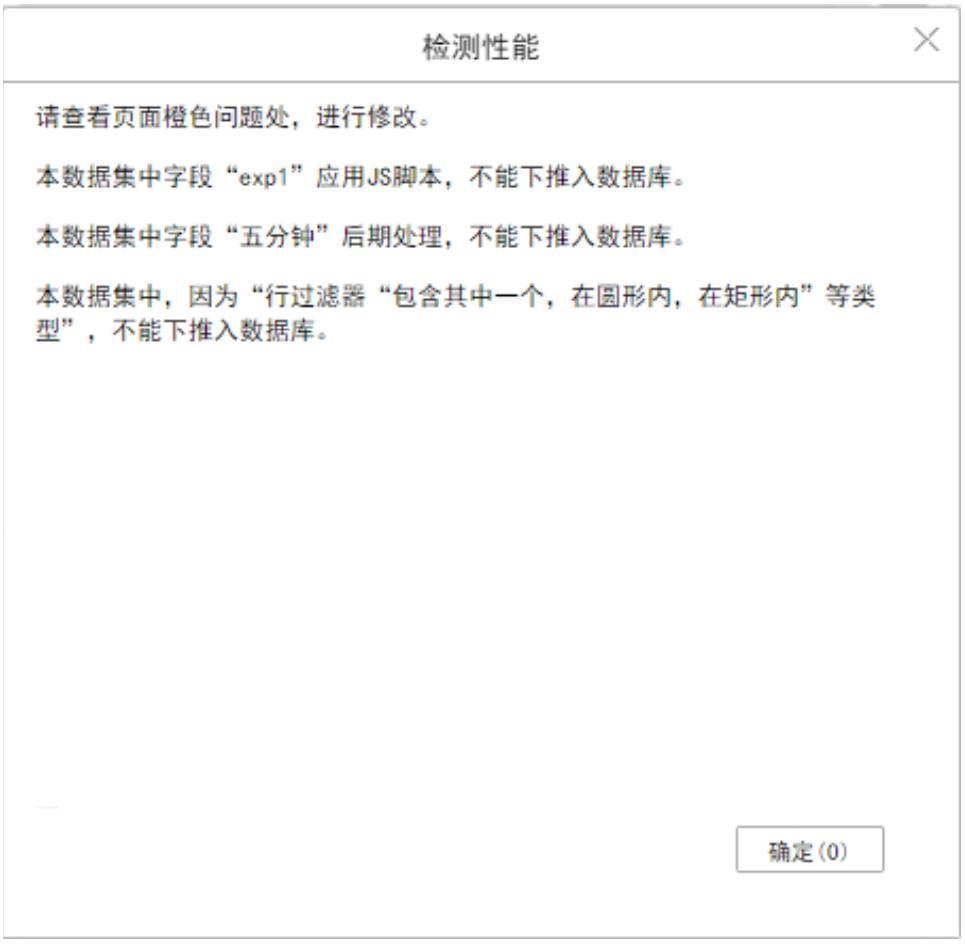
- 引用的数据集存在子级时需要添加双引号，如 select Sales from "cloud/test.clqry"，不存在子级时则不需要添加双引号，如 select Sales from test.clqry
- 引用关键字时需要加引号。如数据集 a.clqry 中存在 Date 字段，由于 Date 字段是数据库中的关键字，则需要被双引号引起来， select "Date" from test.clqry
- 注意赋值类型为字符串类型时，需要用单引号引起来而不是双引号。如 select Nation from test.clqry where Nation='China'

注意事项：在 SYBASE 数据库中不支持 order by 语句。

【性能检测】用户使用 SQL 数据集时，系统会对数据集性能进行实时检测，并对影响性能的地方做出橙色标识，告知用户哪些列操作操作没有下推到数据库执行，如下图所示：

行过滤器处的过滤条件如果没有下推到数据库执行，行过滤器处也会做出橙色标识，如下图所示：

用户也可以通过点击检测性能，查看所有性能问题。



用户点击刷新元数据按钮时，所有数据段将在元数据区域显示。在表达式、过滤器等位置引用的列如果发生变化，刷新元数据时会提示“ 数据集的列如果发生变化可能会导致表达式列、层次或行过滤器不可用，是否清空表达式列、层次和行过滤器？ ”，否则不会提示。本产品默认字符类型、字符串类型、字节类型、布尔类型的数据段存放在维度目录下，其他类型数据段存放在度量目录下。

SQL 数据集的特例

Limit 函数

通过 SQL 语句查询 DATA MART 数据库 (通过永洪的驱动连接)，可以支持 limit n, 来返回前 n 行记录。

例如：

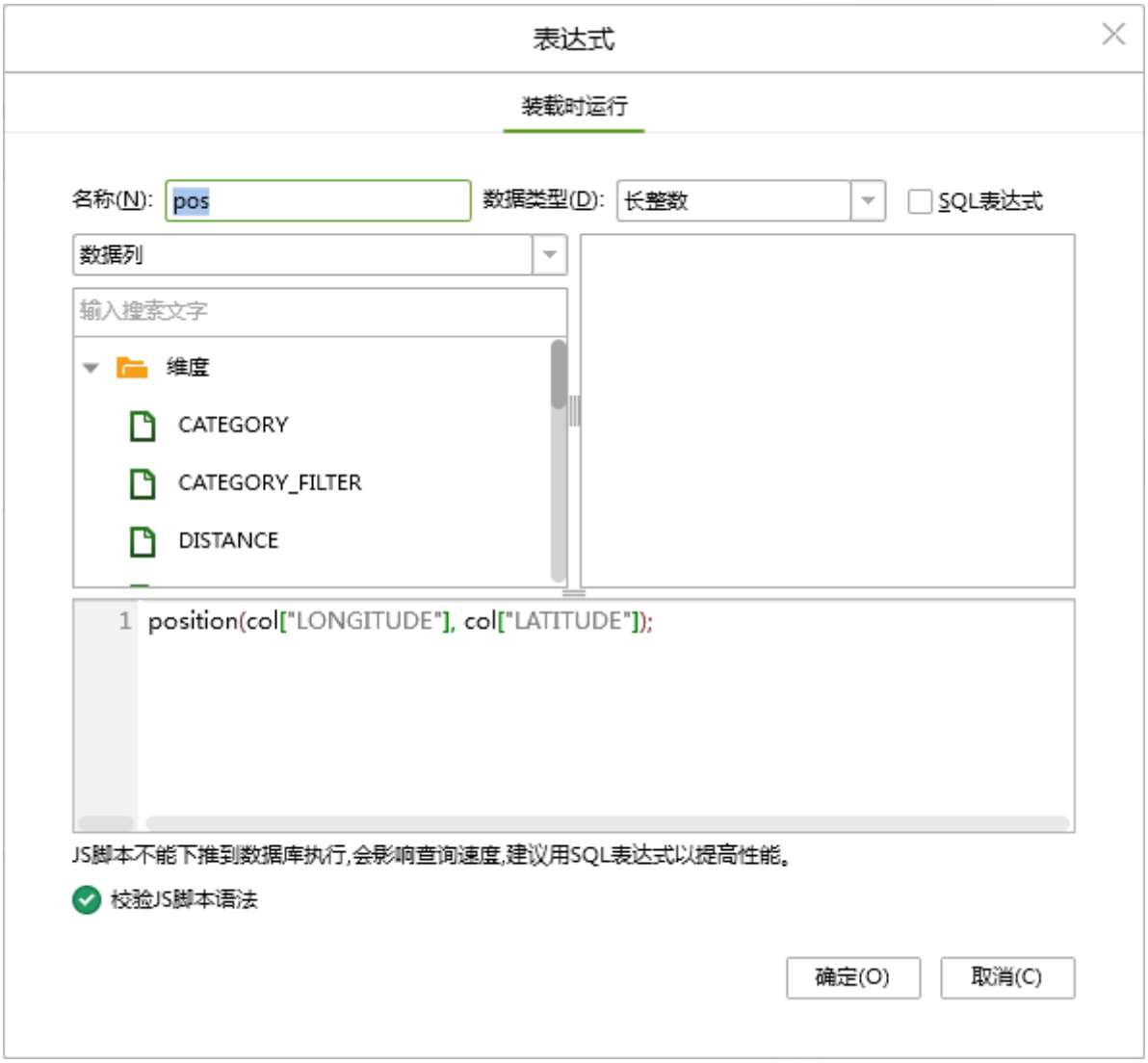
select sum(COGS) COGS , STATE from " 咖啡销售统计 .sqry" group by STATE order by COGS desc limit 3

数据集经纬度数据

SQL 语句支持对经纬度数据的查询。经度的范围是：0-180 度，负数表示西经，正数表示东经；纬度的范围是：0-90 度，负数表示南纬，正数表示北纬。可以通过给定经度值和纬度值查找到给定圆或矩形范围内的数据。

例如：select * from "GIS.sqry" where pos incircle (-83.2215,42.1407, 200) 表示：数据集表 GIS 中，经度值为：-83.2215，纬度值为：42.1407，半径为 200 米的圆形范围内的数据。通过 SQL 语句查询经纬度数据需要使用永洪的驱动。

列 pos 表示中心点的位置。它是通过特定的算法将经度和纬度结合在一起。使用前需要在数据集 GIS 中通过增加表达式的方式来增加 pos 列。创建表达式界面截图如下所示：



参数 incircle 表示在圆形范围内查询。需要输入：经度值、纬度值、半径（米）。

参数 inrect 表示在矩形范围内查询。需要输入：经度值、纬度值、宽（米）和高（米）。

使用 SQL 语句对经纬度数据进行查询的截图如下所示：


数据源(O):
数据源/Conn/YONGHONG
数据库(D):
DATA MART
驱动(I):
g5.dc.jdbc.GDriver
URL:
jdbc:yonghong
服务器地址：端口号
默认数据库(F):
表结构模式(H):
需要登录(T):
最大连接数(M):
用户(E):
admin
密码(P):
**
注释：URL、登录信息等请咨询数据库管理员
表:
点击右键刷新
SQL语句:
select * from "GIS.sqry" where pos incircle (-83.2215,42.1407, 200)
刷新元数据(B)

Excel 数据集

相对于其他数据集，Excel 数据集的使用简单方便。当用户的数据存储在 Excel、CSV、TXT 或 LOG 文件中时，可以通过 Excel 数据集直接上传数据，作为数据集使用。

Excel 数据集目前支持上传 2003 和 2007 类型的 Excel 文件。

创建 Excel 数据集

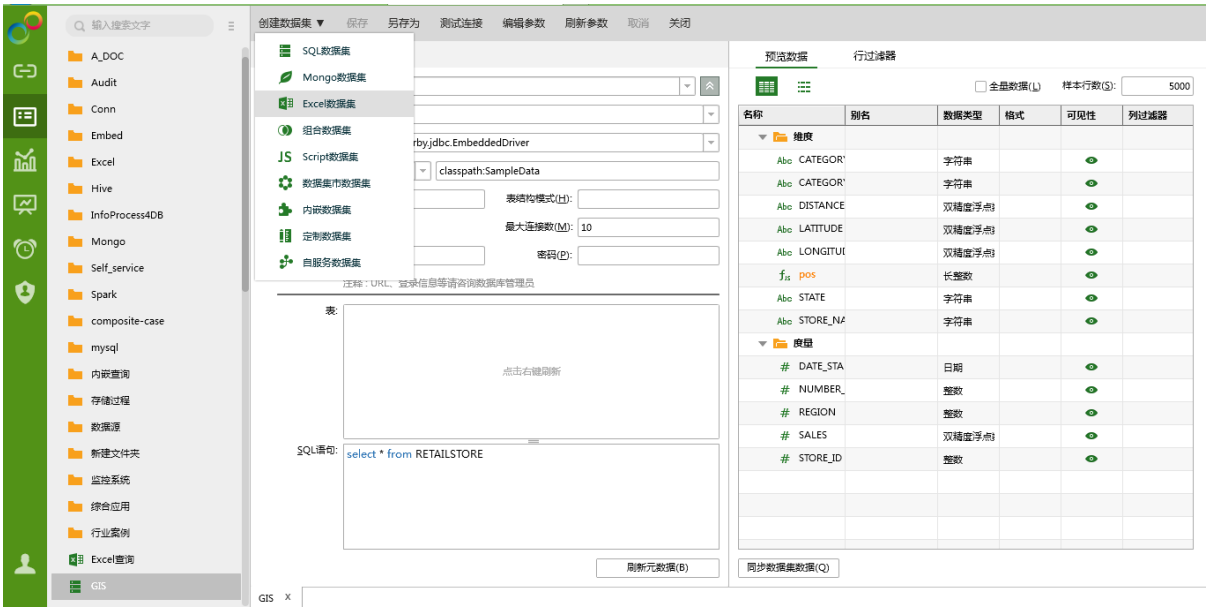
1）点击 Yonghong Z-Suite 产品的启动快捷方式。

2）打开浏览器，然后在地址栏中输入 http://hostname:8080/bi/Viewer, 登陆到客户端。这里的 hostname 是你的 IP 地址，如果是本机访问，可以用 localhost。 8080 是默认端口号，如果在安装产品时修改了默认的端口号，请采用正确的端口号。

3）输入用户名和密码后登陆到主页面。

4）点击连接数据按钮后，进入到连接数据的界面。

5）用户在数据集模块中，点击工具条上的新建数据集按钮，选择 Excel 数据集，即可打开 Excel 数据集界面。



Excel 数据集界面

上传 Excel

【上传】点击上传，弹出对话框，可选择所要上传的文件。

【上传文件】指所上传文件的名称。

【工作表】工作表中列出的是 Excel 中所有 Sheet 表的名称，默认选中第一个 Sheet 表。 Sheet 表可以根据需求单选或多选。多选时，会按照第一个 Sheet 表的列数和列的数据类型进行匹配。当第一个后面的 Sheet 表列数与第一个 Sheet 表列数不匹配时，不能上传。当第一个后面的 Sheet 表列数与第一个 Sheet 表列数据类型不匹配时，会按照第一个 sheet 表的数据类型显示。

【表头】分为自动，第一行，无。自动：表示系统自动判断表头行；第一行：表示将每个 Sheet 表的第一行作为表头行；无：表示各 Sheet 表都没有表头行， Sheet 表中内容都按照数据来处理；

【生成逆透视表】将拥有行表头和列表头的交叉类型的表格生成正常的只有行表头的表格。

行表头层级：所上传表格的行表头数。

列表头层级：所上传表格的列表头数。

【删除无效行】默认上传 excel 时是不删除数据的。如果勾选了删除无效行，系统会自动删除不合理的数据行，例如：数据行中的某一行 10 个数据中有 9 个数据为空，则系统会自动删除这一行。

【刷新元数据】点击刷新元数据，则上传文件的数据被刷新出来，再点击预览数据集即可预览。

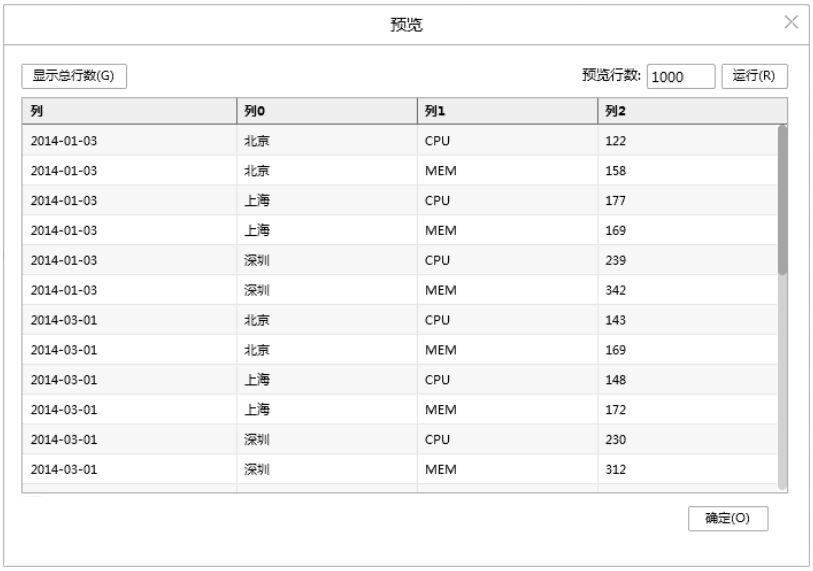
生成逆透视表举例：

1）例如，存在一个交叉类型的 Excel 表格，其中行表头数为 1，列表头数为 2，具体数据如下：

	A	B	C	D	E	F	G	H	I
1		北京		上海		深圳		南京	
2		CPU	MEM	CPU	MEM	CPU	MEM	CPU	MEM
3	2014/1/3	122	158	177	169	239	342		
4	2014/3/1	143	169	148	172	230	312	113	95
5	2015/2/1	178	260	134	173	211	300	119	100
6	2015/3/1	146	164	180	172	213	344	116	101

2）上传，勾选生成逆透视表，并设置行表头层级为：1，列表头层级为：2：

3）刷新元数据，对表格中的数据进行处理，即：1 个行表头数据，2 个列表头数据和数据都作为一列，数据按照从上到下，从左到右并与行列表头数据对应的顺序列出。预览数据集时的数据为：



上传 CSV

【编码类型】当选择上传的文件后，系统会自动加载对应的编码类型，也可以根据需求自定义。

【分隔符】系统会自动检测分隔符。用户也可以通过下拉框，选择分隔符。可选的分隔符类型包括：逗号、分号、竖线、制表符、空格、自定义。选择自定义时，用户可自己输入所需的分隔符

【起始行】读取数据开始的行数，默认是 1。

其余功能与上传 Excel 相同，请参考上传 Excel 界面。

上传 TXT， LOG 的上传界面与 CSV 上传界面相同，不再重复介绍。

内嵌数据集

相对于其他八个数据集功能，内嵌数据集提供的功能比较简单，适用于数量级别要求不高的用户。当用户的数据量不大，参数对象有限，数据关系固定，更新需求不多的时候，内嵌能够很好地满足这种数据固化的需求。通过提供简单的自定义数据集方式，方便用户直接创建数据结构，定义参数类型，给参数赋值，生成所需要的数据集表单。

创建内嵌数据集

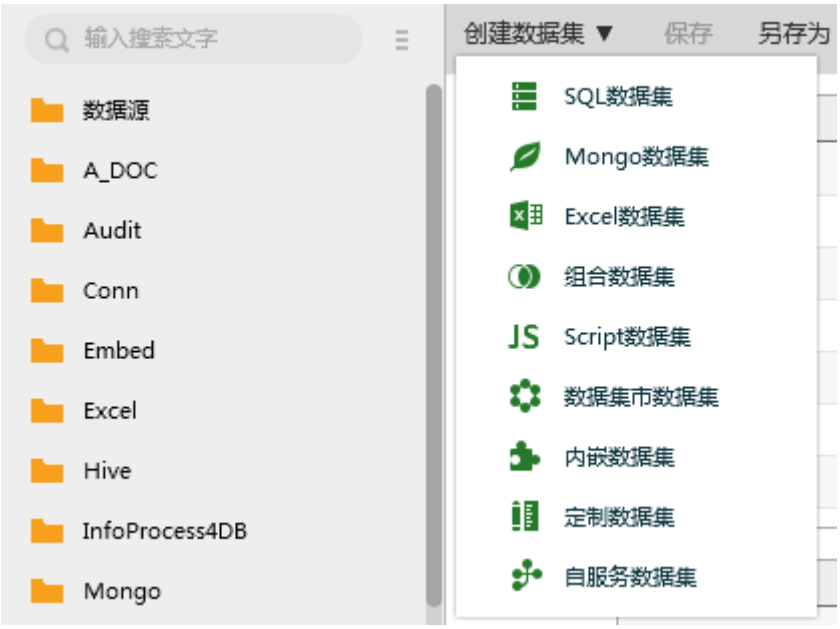
1）点击 Yonghong Z-Suite 产品的启动快捷方式。

2）打开浏览器，然后在地址栏中输入 http://hostname:8080/bi/Viewer, 登陆到客户端。这里的 hostname 是你的 IP 地址，如果是本机访问，可以用 localhost。 8080 是默认端口号，如果在安装产品时修改了默认的端口号，请采用正确的端口号。

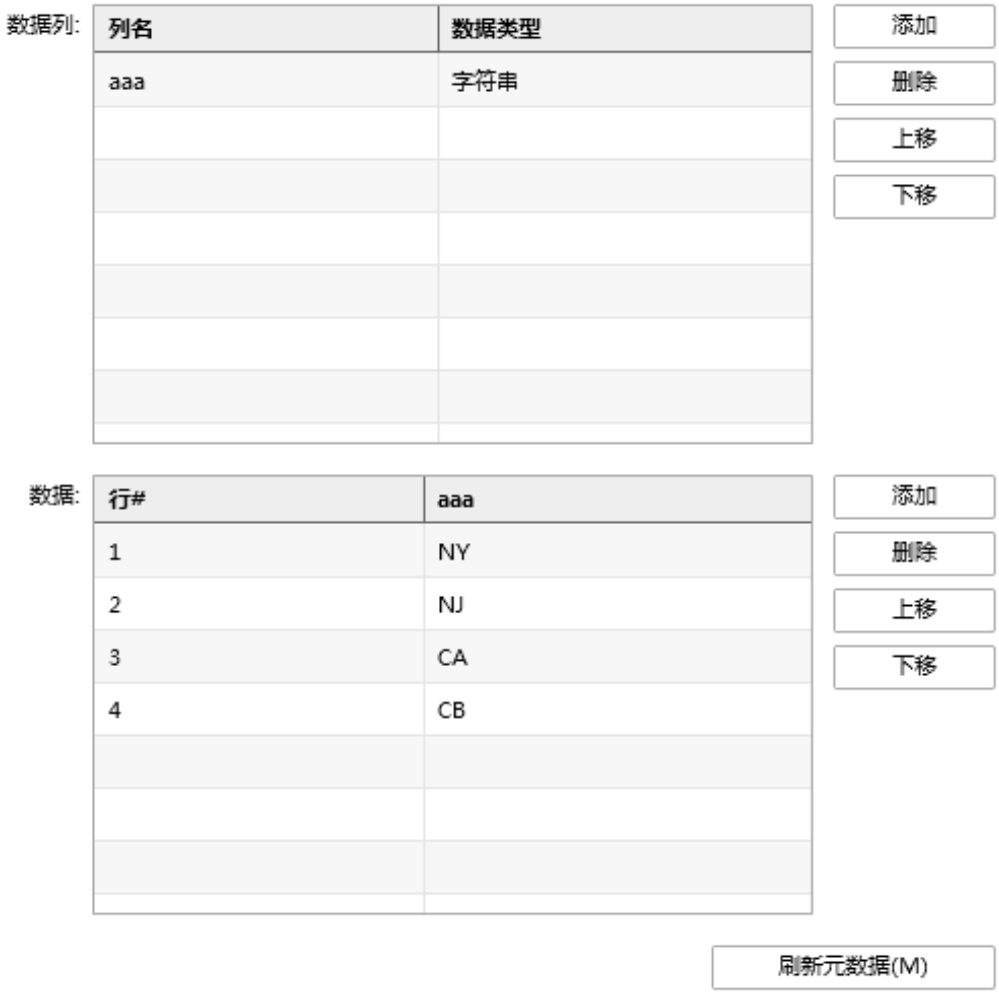
3）输入用户名和密码后登陆到主页面。

4）点击连接数据按钮后，进入到连接数据的界面。

5）用户在数据集模块中，点击菜单栏上的创建数据集按钮，选择内嵌数据集，即可打开内嵌数据集界面。



内嵌数据集界面

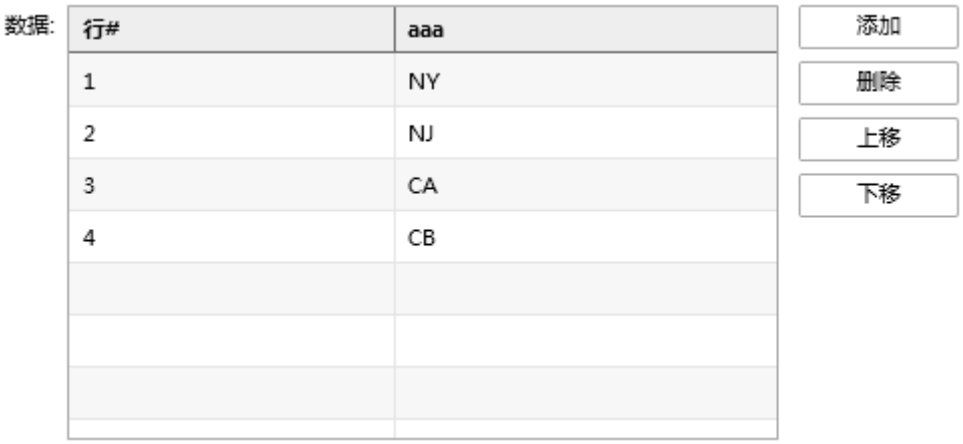


【创建数据段】用户点击下图中红色区域中的添加按钮，即可添加一个数据段。用户可设定该数据段的名称及数据类型。

当用户需要删除该数据段时，首先选中需要删除的数据段，然后点击右侧的删除按钮，即可删除当前数据段。

当用户需要调整已有数据段的顺序时，首先选中需要调整位置的数据段，然后点击右侧的上 / 下按钮，进行位置的调整。

【输入数据】在用户设定好数据段名称后，在数据区域生成相应的数据段，用户可为其添加数据。



当用户需要删除数据时，首先选中需要删除的数据，然后点击右侧的删除按钮，即可删除当前行的所有数据。

当用户需要调整已有数据的顺序时，首先选中需要调整位置的数据行，然后点击右侧的上 / 下按钮，进行位置的调整。

组合数据集

组合数据集提供了一个强大却又简单的方式，通过简单的拖拽操作，实现多个查询，多张表或视图联接（JOIN）或联合（UNION）在一起。这些查询，表，视图均可以来自于不同的数据源。

创建组合数据集

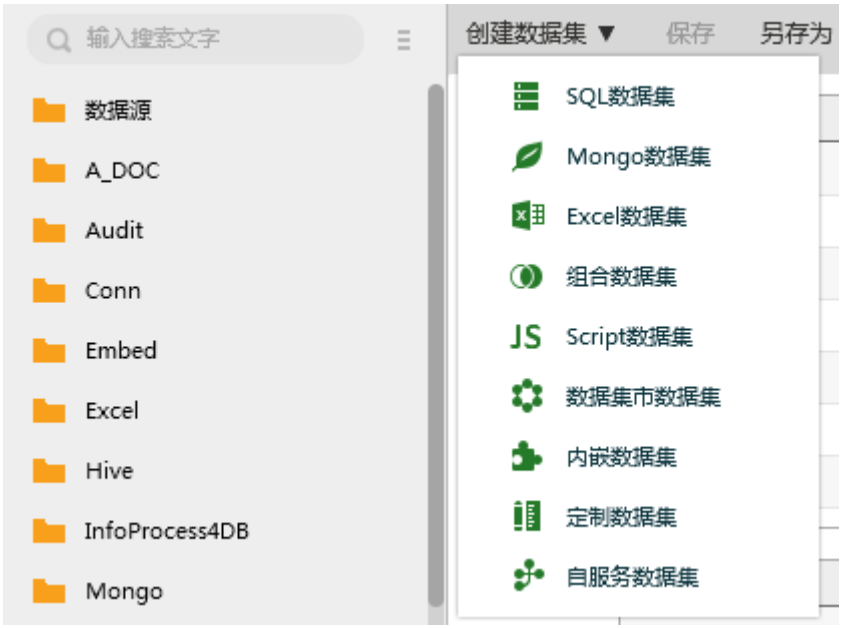
1）点击 Yonghong Z-Suite 产品的启动快捷方式。

2）打开浏览器，然后在地址栏中输入 http://hostname:8080/bi/Viewer, 登陆到客户端。这里的 hostname 是你的 IP 地址，如果是本机访问，可以用 localhost。 8080 是默认端口号，如果在安装产品时修改了默认的端口号，请采用正确的端口号。

3）输入用户名和密码后登陆到主页面。

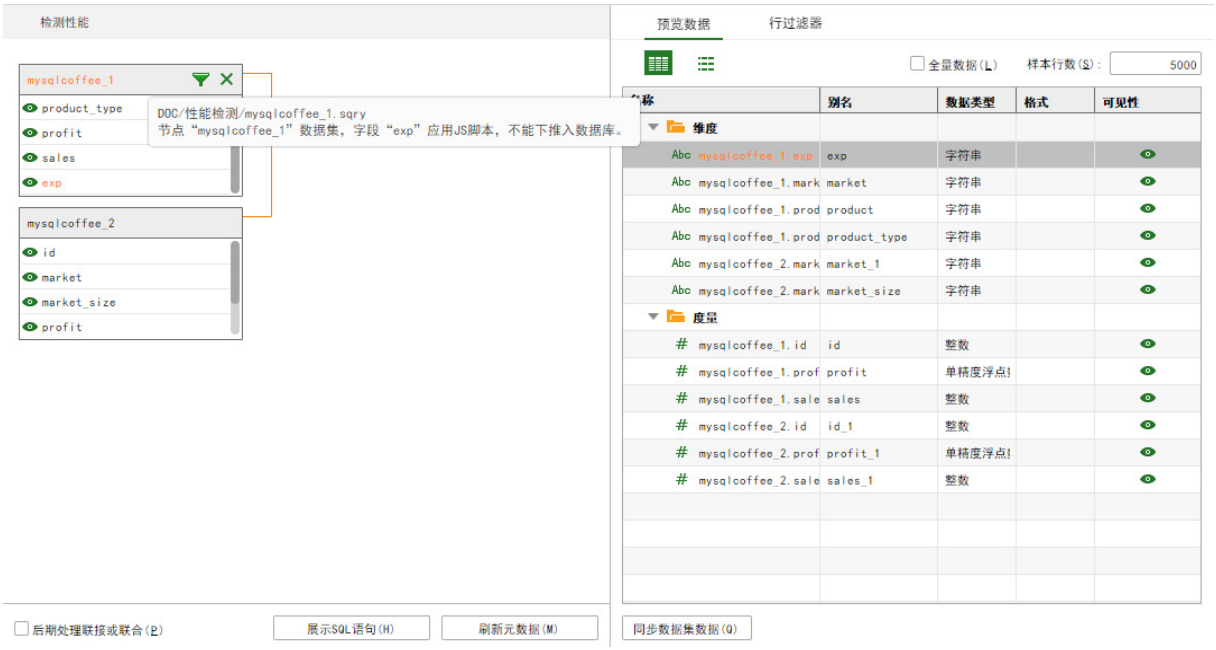
4）点击连接数据按钮后，进入到连接数据的界面。

5）用户在数据集模块中，点击工具条上的新建数据集按钮，选择组合数据集，即可打开组合数据集界面。



组合数据集实时性能检测

用户使用组合数据集时，系统会对数据集性能进行实时检测，并对影响性能的地方做出橙色标识，告知用户哪些操作操作没有下推到数据库执行：



和 SQL 数据集一样，在组合数据集上，用户也可以通过点击检测性能，查看所有性能问题。

添加表

组合数据集可以直接从左边的数据集资源树上拖拽一个数据集到组合数据集编辑区域；也可以拖拽数据源里的表或视图到组合数据集编辑区域。

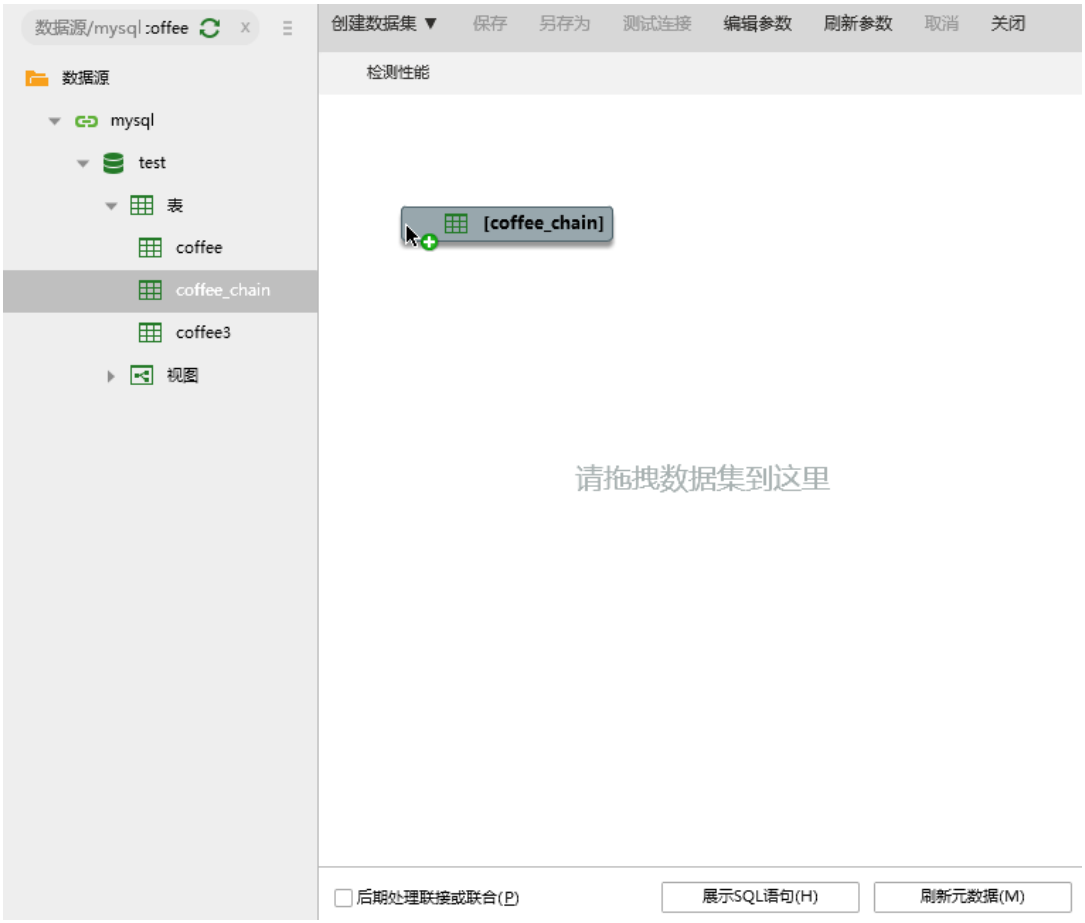
当把数据集、表、视图拖拽到组合数据集里后，在这个文档里，统一叫“表”。

添加数据集

从左边资源树上拖拽一个数据集得到组合数据集编辑区域。任何类型的数据集都可以被添加进来。一个组合数据集里不能仅仅只有一张表，这张表来自于数据集。

添加数据库中的表或视图

展开数据源的表和视图节点，拖拽一张表或一个视图到组合数据集编辑区域。一个组合数据集里可以仅仅只有一张表，这张表来自于表或视图。



添加区域

如果将表插入到一个空白的组合数据集里，可以通过 DND 的方式将这张表放入组合数据集编辑区域的左上角。

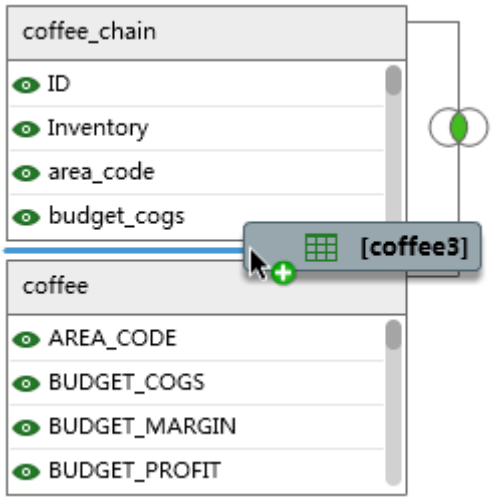
如果组合数据集编辑区域里已经有别的表，可以将新加的表放到：

- 第一张表之前
- 两张表之中
- 最后一张表之后
- 已存在的表上
- 其它空白区域

如果添加到已存在的表上，新添加的表会替换老的表。

如果添加到其它空白区域，新添加的表会插入到最后一张表之后。

下图为添加到两张表之中的截图，用蓝色粗线提示此区域可以插入：



定义表间的联接关系

拖拽数据库中的表，如果两个表之间定义了外连接信息，当两个表拖入组合数据集之后，可以基于这些外连接信息将缺省的连接做好，而无需用户指定。

点击表与表之间连接线上的图标，即可以打开“操作类型”窗口，来定义两张表之间的联接关系。

两张表之间的关系图：

关系	图标
内部联接	
左侧联接	
右侧联接	
外部联接	
联合	
无效连接	

联接

联接类型：

内部联接（Inner Join): 将左右两张表符合联接条件的记录组合在一起。

左侧联接 (Left Join): 左表的记录全部显示，右表只会显示符合联接条件的记录，右表中记录条件不足的地方补空。

右侧联接 (Right Join): 右表的记录全部显示，左表只会显示符合联接条件的记录，左表中记录不足的地方补空。

外部联接 (Outer Join): 左右表的记录都全部显示，左表不符合联接条件的记录对应的右表位置补空，右表不符合条件的记录对应左表位置补空。

联接条件：

在 “ 操作类型 ” 窗口里，点击 “ 添加新的联接列 ” 来定义联接左右两张表的条件。从左右表中选择数据类型匹配的列作为连接条件列。两张表之间可以定义多个条件。

无效联接：当没有定义两张表的联接条件，或联接条件无效时，这种联接关系就是一种无效的联接。

联合

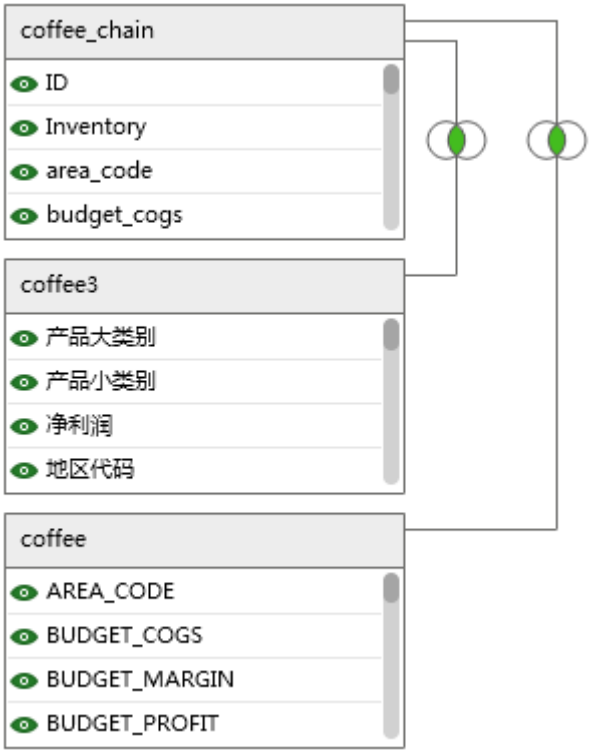
联合 （ Union All): 将两张表的数据 （ 包括重复数据行 ）拼接在一起，左表数据在上，右表数据在下，联合后的表列名为左表的列名。

无效联合：联合的两张表，表的列数必须相同，对应列的数据类型必须匹配，如果不满足上述两个条件，这种联合便是一种无效联合。

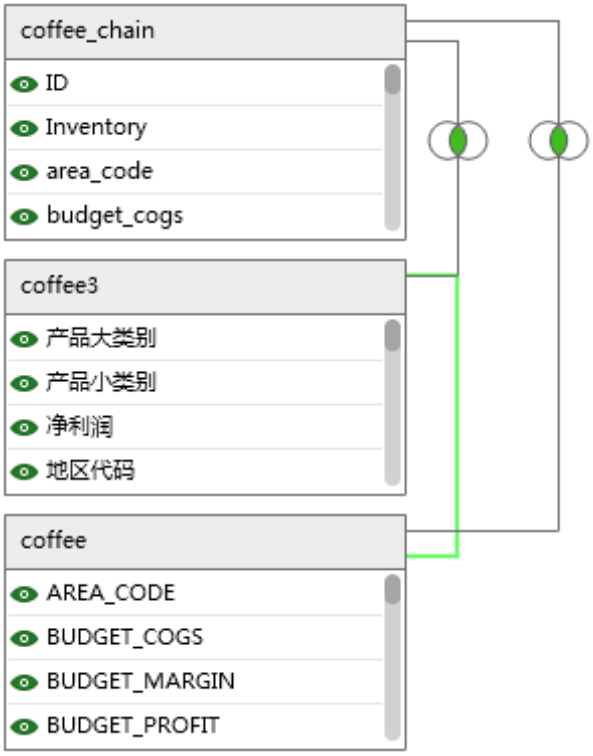
移动连接线

新插入的表，默认会找插入点的上一张表来产生连接。如果不存在上一张表，会找插入点的下一张表产生连接。也可以拖动连接线，来重新选择连接的左右两张表。

移动鼠标到连接线上，当鼠标变为可移动状态时，上 / 下拖动它到另一张表上。



当连接线变为绿色时，即可释放鼠标。如下图：



组合数据集上，任何一张都必须与任意一张别的表产生关系，所有连接线的个数为（表的个数 -1 ）。当拽连接线导致某张表和别的表没有关系时，连接线的颜色会显示为红色，示意非法拖动连接线。

移动表

选择表的表头部分，拖动表到新的位置，可以插入到另一个表之前，或附加到另一个表之后，也可以移动到另一个表上替换这个表。

编辑表

组合数据集里的表，可以定义过滤器，隐藏列，改变列顺序，改变表的尺寸。

定义过滤器：点击表头上的过滤器按钮，打开过滤器窗口，定义过滤。

隐藏列：点击列名前的小图标来显示 / 隐藏列。在表上右键可以显示 / 隐藏所有列。

改变列顺序：拖动列到另一列的前面或后面来改变列的顺序。隐藏列总是排在最后的位置。

改变表的尺寸：拖拽表的底部边线改变这一个表的高度；拖拽表的右边线，改变在同一列上的所有表的宽度。

删除表

点击表头上的 “ 删除数据集 ” 按钮即可删除表。

展示 SQL 语句

可以点击组合数据集编辑区域下方的 “ 展示 SQL 语句 ” 按钮，来查看这个组合数据集生成的 SQL 语句。

Script 数据集

Script 数据集提供脚本语言和数据接口的方式，实现 SQL 数据集所提供的功能。但这种方式比较于 SQL 数据集本身设计上的局限，又能使数据集操作更加灵活自由，为专业人士提供了便利。通过在工具库里提供了一些常用 Javascript 函数接口， Script 数据集支持客户通过编程的方式进行查询。虽然这些接口被限制在设定功能范围内，但是已经能够满足现有日常的工作。

此外 Script 数据集提供了 SQL 数据集不支持的跨数据库联合数据集功能。其类似于 SQL 中的联合查询，根据客户需求，将来自不同数据表单上的数据，通过脚本 JOIN 数据集方法展现在一张表单上。但 Script 数据集提供了比 SQL 的联合查询更为强大的功能。 SQL 联合数据集的前提是所有的数据表单都来自于同一个数据库，而 Script 数据集可以连接不同数据库的数据表单，例如某个公司的经营数据保存在 DB2，而其管理数据保存在 ORACLE，客户通过 Script 数据集可以把他们联合在一起并组成一张新的数据集表用来分析，实现了不同数据库之间的数据合并。

创建 Script 数据集

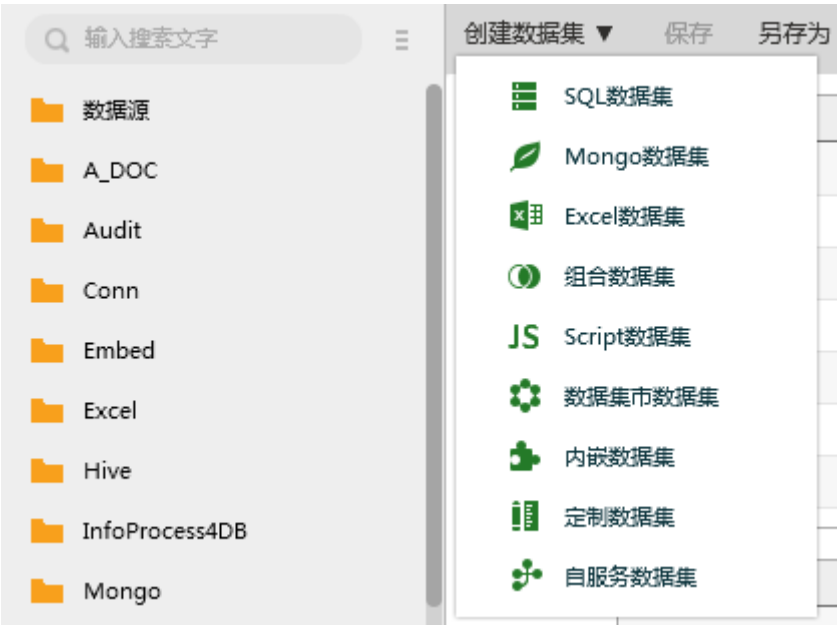
1）点击 Yonghong Z-Suite 产品的启动快捷方式。

2）打开浏览器，然后在地址栏中输入 http://hostname:8080/bi/Viewer, 登陆到客户端。这里的 hostname 是你的 IP 地址，如果是本机访问，可以用 localhost。 8080 是默认端口号，如果在安装产品时修改了默认的端口号，请采用正确的端口号。

3）输入用户名和密码后登陆到主页面。

4）点击连接数据按钮后，进入到连接数据的界面。

5）用户在数据集模块中，点击工具条上的新建数据集按钮，选择 Script 数据集，即可打开 Script 数据集界面。



Script 数据集界面

在打开的界面中，用户可编写脚本语句，来实现数据集的连接。脚本查询语句请参考 Script 文档中的详细介绍。

Script 数据集可实现不同类型数据集的链接。

数据集	对应标识
SQL 数据集	SQL
Excel 数据集	EXCEL
Script 数据集	SCRIPT
定制数据集	CUSTOM
数据集市数据集	CLOUD
内嵌数据集	EMBED
组合数据集	COMPOSITE
Mongo 数据集	MONGO
自服务数据集	DATA_FLOW

Script 数据集举例说明

数据库中的表可通过主键将彼此联系起来。主键 （ Primary Key ）是一个列，在这个列中的每一行的值都是唯一的。在表中，每个主键的值都是唯一的。这样做的目的是在不重复每个表中的所有数据的情况下，把表间的数据交叉捆绑在一起。

在数据集列表中存在两个 SQL 数据集，一个是 Persons, 一个是 Orders。

请看 Persons 数据集中的数据：

Id_P	LastName	FirstName	Adress	City
1	Adams	John	Oxford Street	London
2	Bush	George	Fifth Avenue	New York
3	Carter	Thomas	Changan Street	Beijing

请注意， "Id_P" 列是 Persons 数据集中的主键。这意味着没有两行能够拥有相同的 Id_P。即使两个人的姓名完全相同， Id_P 也可以区分他们。

接下来请看 Orders 数据集中的数据：

Id_O	OrderNo	Id_P
1	77895	3
2	44678	3
3	22456	1
4	24562	1
5	34764	65

请注意， "Id_O" 列是 Orders 数据集中的的主键，同时， "Orders" 表中的 "Id_P" 列用于引用 "Persons" 数据集中的人，而无需使用他们的确切姓名。

"Id_P" 列把上面的两个数据集联系了起来。

编写脚本查询语句，实现以上两个数据集的连接。

```
var lt = execute(this, SQL, "Persons");

var rt = execute(this, SQL, "Orders");

var lkeys = [0];

var rkeys = [2];

var lcols = [0, 1,2];

var rcols = [1];

join(this, FINAL_JOIN | LEFT_MAIN, LEFT_JOIN, lt, rt, lkeys, rkeys, lcols, rcols);
```

以上语句通过 Id_P 实现了 Persons 数据集与 Orders 数据集的连接，以 Persons 数据集为主，并保留 Persons 数据集的第 0,1,2 列， Orders 数据集的第 1 列。

连接结果如下表所示。

Id_P	LastName	FirstName	OrderNo
1	Adams	John	22456
1	Adams	John	24562
3	Carter	Thomas	77895
3	Carter	Thomas	44678
2	Bush	George	33656

数据集市数据集

数据集市数据集作为 Yonghong Z-Suite 云计算的接口为用户提供云数据数据集服务。当用户购买了 Yonghong Z-Suite 云计算模块以后，需要根据自己的需求配置云环境，包括硬件配置和软件配置两部分。通过将后台数据与云计算应用关联起来，云计算模块就可以正常工作了。由于云计算支持的服务器数量众多，需要处理的数据量级别超过普通数据量，必须有特殊的文件存储系统为数据集市数据集服务。

云计算所需要的查询数据都会被单独存放在指定的文件目录下，数据集市数据集创建新的数据集的时候，用户可以直接指定到该目录下，选择已经部署好并可以使用的文件创建数据集市数据集表。一般情况下数据集的数据会在创建数据集之前被定义，并定时的与云端数据互动更新，保证在文件夹数据是最新的，这也是 Yonghong Z-Suite 相对于同类软件的独有的功能。

数据集市数据集市数据集同时支持用户通过调用云计算的 API 接口生成的云数据文件。这种方式相比较通过系统生成的文件更加灵活方便，但是对于用户的技术要求也很高，需要用户自己设计程序和应用。由此产生的数据集表单与其他数据集功能产生的表单是一致的，没有特别的地方。

创建数据集市数据集

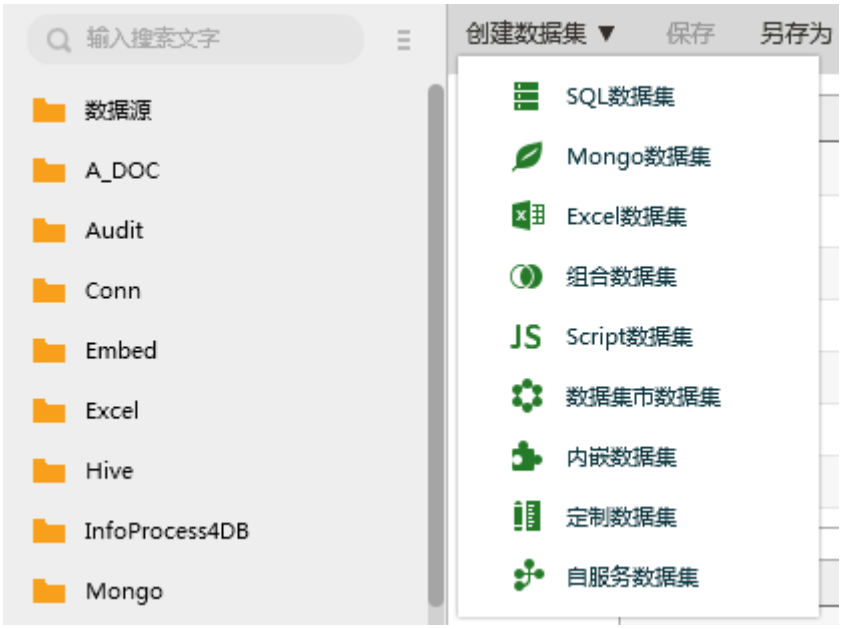
1）点击 Yonghong Z-Suite 产品的启动快捷方式。

2）打开浏览器，然后在地址栏中输入 http://hostname:8080/bi/Viewer, 登陆到客户端。这里的 hostname 是你的 IP 地址，如果是本机访问，可以用 localhost。 8080 是默认端口号，如果在安装产品时修改了默认的端口号，请采用正确的端口号。

3）输入用户名和密码后登陆到主页面。

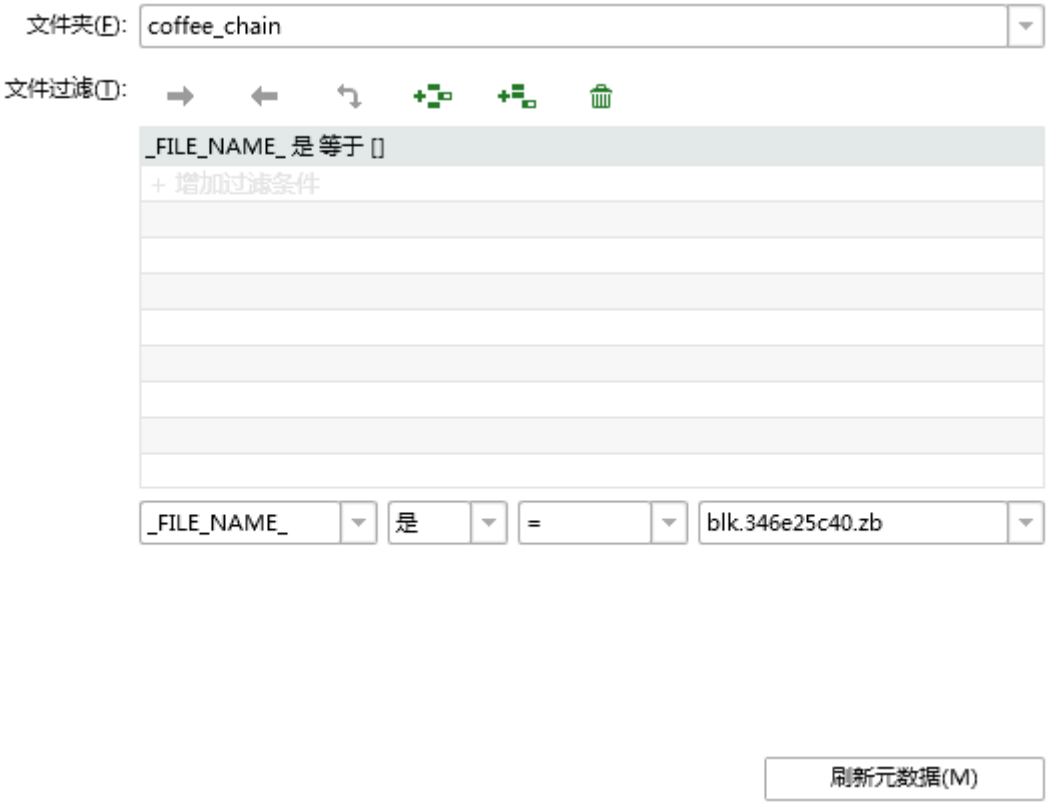
4）点击创建数据集创建数据集创建数据集据按钮后，进入到创建数据集的界面。

5）用户在数据集模块中，点击工具条上的创建数据集按钮，选择数据集市数据集，即可打开数据集市数据集界面。



数据集市数据集界面

文件夹列表中列出的是在定时任务中运行过的增量导入数据任务的文件夹。通过物化数据集按钮物化或通过定时任务物化的数据集不在列表中列出。



定制数据集

定制数据集基于 Script Query 的设计理念，提供给客户更加自由灵活的操作方式，完全由用户自己定义数据集条件和选择数据来源，因此它也是这九种数据集方式里面对于客户技术要求最高的。用户在使用 Yonghong Z-Suite 之前已经建立好自己的数据集系统，并和自己的数据系统深度绑定，因此不需要重新定义数据集机制。定制数据集正好满足此类用户的需求，可以将自有的数据集系统与 Yonghong Z-Suite 快速绑定。

在建立新的定制数据集的时候，提供 g5.qry.GCustomQuery 类接口给用户，帮助他们通过 Java 编程的方式与自己原有数据集系统进行绑定。通过继承 Query 的方式， Yonghong Z-Suite 可以返回用户自定义的数据集结果。另外当用户希望直接从数据库获取原始数据而不使用数据集功能的时候，定制数据集也可以帮助用户完成此类操作。

定制数据集 具有并支持和 Script Query 类似的联合数据集功能，可以将多张表单的数据集结果链接在一起，同时支持不同数据库类型。但是他们之间也有一些不同， Script Query 是基于可视化界面的输出进行数据操作，而定制数据集在技术方面则要求更高，需要用户自定义 java 文件。

创建定制数据集

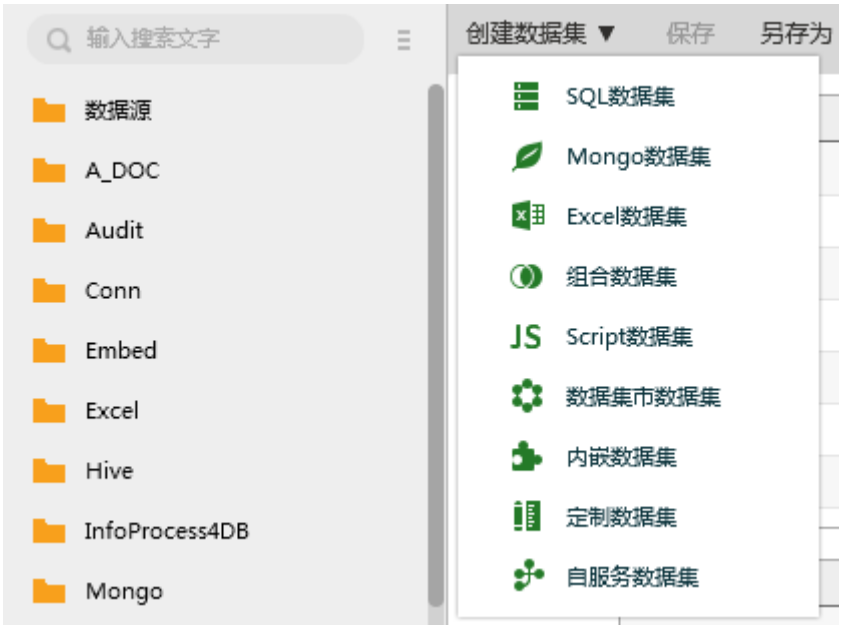
1）点击 Yonghong Z-Suite 产品的启动快捷方式。

2）打开浏览器，然后在地址栏中输入 http://hostname:8080/bi/Viewer, 登陆到客户端。这里的 hostname 是你的 IP 地址，如果是本机访问，可以用 localhost。 8080 是默认端口号，如果在安装产品时修改了默认的端口号，请采用正确的端口号。

3）输入用户名和密码后登陆到主页面。

4）点击创建数据集按钮后，进入到创建数据集的界面。

5）用户在数据集模块中，点击工具条上的新建数据集按钮，选择定制数据集，即可打开定制数据集界面。


输入搜索文字
数据源
A_DOC
Audit
Conn
Embed
Excel
Hive
InfoProcess4DB
Mongo
创建数据集 ▼
保存
另存为
SQL数据集
Mongo数据集
Excel数据集
组合数据集
Script数据集
数据集市数据集
内嵌数据集
定制数据集
自服务数据集

定制数据集界面


类名(N):
DataQry
请输入定制数据集的类名,该类需继承g5.qry.GCustomQuery。
刷新元数据(M)

Mongo 数据集

YonghongBI 支持连接 MongoDB 数据源，进行数据查询、计算和分析。通过输入 URL、用户名和密码成功连接 MongoDB 后，选择数据库中的某个集合，即可以数据集这个集合中的数据。产品支持管道操作（Pipeline）语句对集合进行过滤、分组、聚合和排序等等操作。

创建 Mongo 数据集

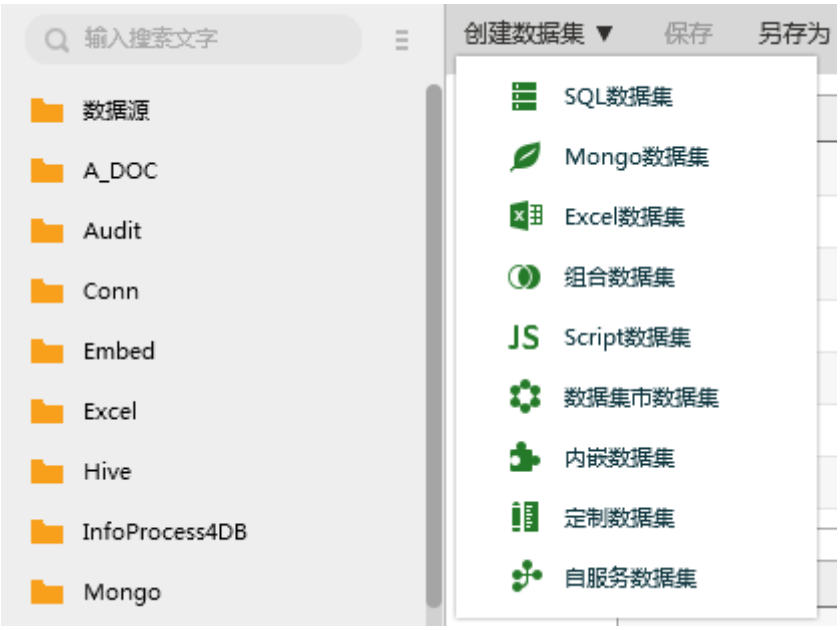
1）点击 Yonghong Z-Suite 产品的启动快捷方式。

2）打开浏览器，然后在地址栏中输入 http://hostname:8080/bi/Viewer, 登陆到客户端。这里的 hostname 是你的 IP 地址，如果是本机访问，可以用 localhost。 8080 是默认端口号，如果在安装产品时修改了默认的端口号，请采用正确的端口号。

3）输入用户名和密码后登陆到主页面。

4）点击创建数据集按钮后，进入到创建数据集的界面。

5）用户在数据集模块中，点击工具条上的新建数据集按钮，选择 Mongo 数据集，即可打开 Mongo 数据集界面。



Mongo 数据集界面

在打开的界面中，用户可从已有的数据源中选择 mongo 数据源，也可以自己设定 URL、集合等，并且可编辑使用管道操作语句。

【数据源】当用户选择自定义数据源时，可配置相应的数据源。用户也可以从已有的数据源列表中选择已经创建好的 Mongo 数据源。

【URL】 Mongo 数据库的 URL 地址。

【选择转换时区】输入数据存入 Mongo 时的时区。如果数据存入 Mongo 时，没有指定时区，即可以不用做选择。如果存入数据时，指定了时区，在这需要选择相应的时区对数据进行转换。

【需要登录】当数据库设定了访问权限后，用户需要勾选此项，使用用户名和密码来访问当前数据库。

【默认数据库】指定默认数据库。

【集合】可以在此输入集合（Collection）名称，也可以从下方选择集合中选择一个集合。

【选择集合】点击右键刷新，此列表中列出了默认数据库下的所有集合。右键刷新后可以对数据库下的集合进行搜索。详细用法可参考简介中数据源搜索部分。

【管道操作语句】用户输入管道操作（Pileline）语句，实现对数据库中数据的数据集。

在选择或输入集合名称之后，点击刷新元数据，既可以按列以 Table 的形式显示集合中的所有数据。

也可以输入管道操作语句，产品可以解析 “{}” 括起来的所有正确的管道操作语句，多个管道操作语句之间用逗号分隔。

如 :

{ $project : { market:{$toUpper:"$market"} , sales:1, _id:0 } },

{ $sort : { market : 1, sales: 1} }

自服务数据集

自服务数据集提供强大，便捷的数据准备和整合方式。无论是 IT 人员，还是业务人员，都可以通过节点之间相连将数据进行整合，清洗，做好完善的数据准备工作。

用户可通过添加数据节点的方式，将来自不同类型的数据集数据作为输入节点，例如 Excel 数据集，内嵌数据集，SQL 数据集，Script 数据集 ,Mongo 等各种任意数据集。在输入节点之后接入各种联接和转换节点，各个节点之间可以任意组合和编辑，最后连线数据集结果节点，就可以完成数据的准备工作。

创建自服务数据集

1）点击 Yonghong Z-Suite 产品的启动快捷方式。

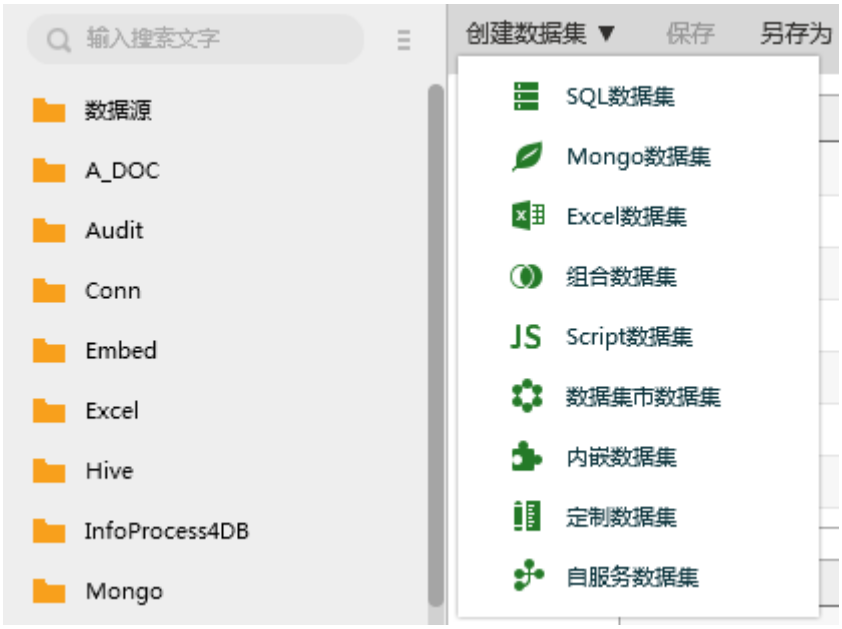
2）打开浏览器，然后在地址栏中输入 http://hostname:8080/bi/Viewer, 登陆到客户端。这里的 hostname 是你的 IP 地址，如果是本机访问，可以用 localhost。 8080 是默认端口号，如果在安装产品时修改了默认的端口号，请采用正确的端口号。

3）输入用户名和密码后登陆到主页面。

4）点击创建数据集按钮后，进入到创建数据集的界面。

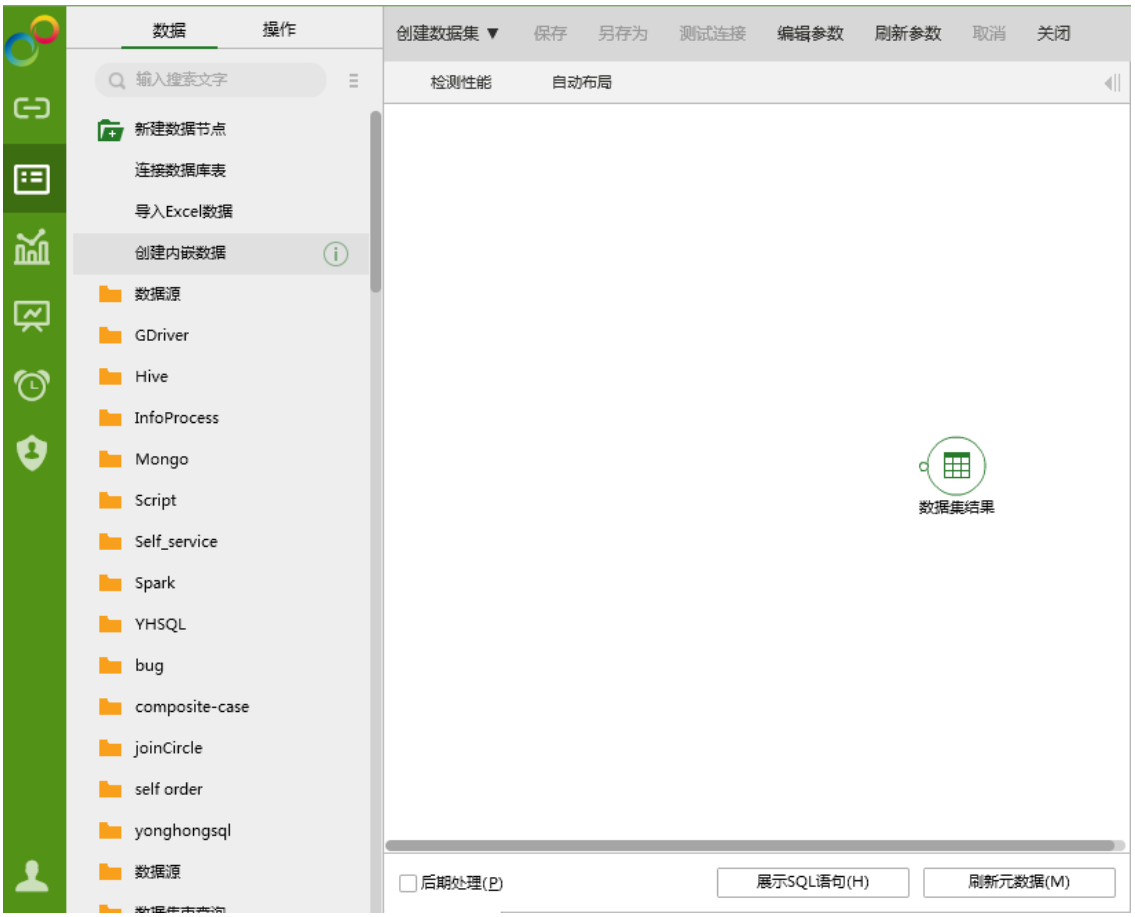
进入自服务数据集的其它方式，可参考 SQL 数据集。

5）用户在数据集模块中，点击工具条上的新建数据集按钮，选择自服务数据数据集，即可打开自服务数据数据集界面。

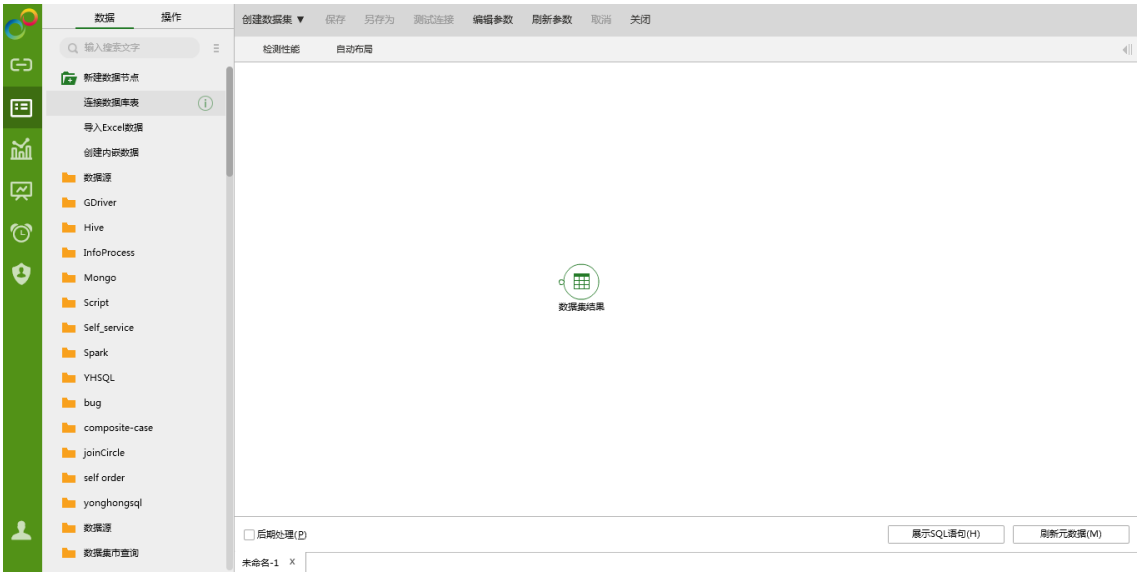


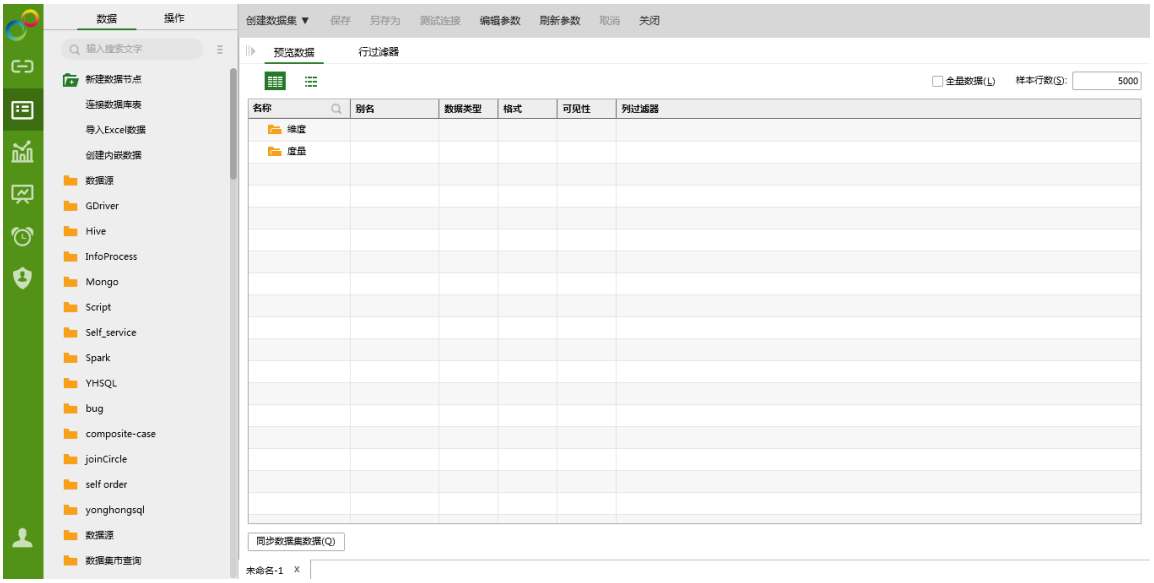
自服务数据集界面

自服务数据集的界面中，会默认带上一个数据集结果的节点。左侧资源数据集树上有数据和操作 2 个部分，两部分可以互相切换。数据区域 可以新建数据节点（连接数据库表、导入 Excel 数据、创建内嵌数据），操作包含关联（包括联接和联合）和转换（包括逆透视表、分组和汇总、自循环列、镜像和去重）；中间空白部分称之为画布部分；自服务还有性能检测和自动布局功能。



自服务数据数据集的界面中，有向右伸展、向左伸展的图标，点击会隐藏数据区或画布，如下图所示。





节点类型

在自服务数据集中，通过添加不同类型的节点，并且添加连线做数据处理。节点主要分为输入节点和中间节点（关联，转换节点），数据集结果节点。可以通过拖拽的方式将节点添加到自服务的空白画布区域。

输入节点

输入节点的右键菜单有：重命名，复制，删除，刷新。节点刷新可以更新数据。

输入节点可以直接从左边的数据集资源树上 “ 数据 ” 区域拖拽，可拖拽的类型主要分为三种：

a ）普通数据集。添加到画布上面的数据集类节点可以通过双击打开节点查看节点详细信息。

b ）数据源里的表或视图。

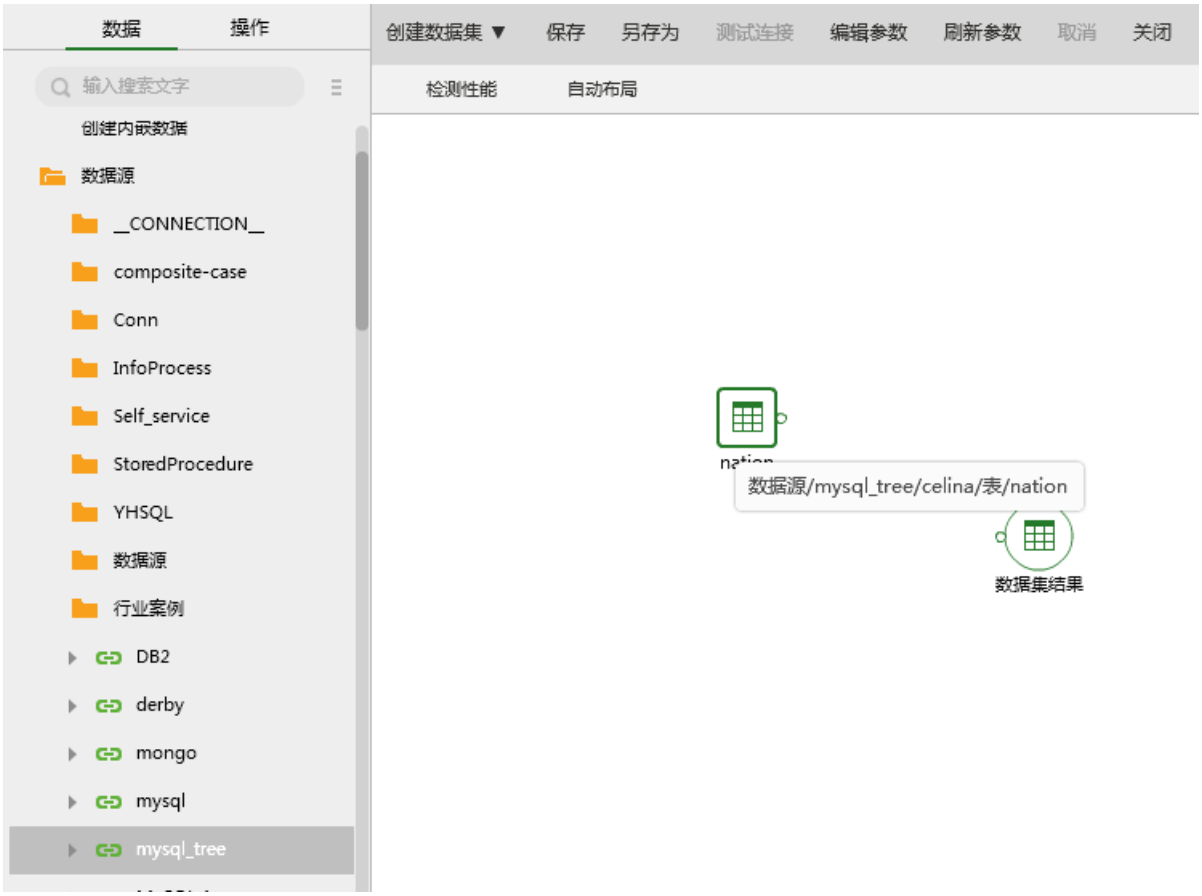
c) 新建数据节点：连接数据库表，导入 Excel 数据，创建内嵌数据，这是自服务独有的创建数据的方式，方便快捷。

输入节点只能连接一个输出节点，但镜像节点除外。

连接数据库表

在左侧的数据集资源树上，选择新建数据节点 > 连接数据库表，拖拽连接数据表到右侧画布空白处，会弹出如同 SQL 数据集一样的界面。如下图所示，选择 MYSQL 数据库，设置好连接属性，在选择 “ 表 ” 的区域，选择一张数据表 或者 视图，例如 表 nation，选择保存并应用到指定位置，保存名称为 mysql_tree, 如下图所示。

在数据集资源树上数据源文件夹下会生成名称为 mysql_tree 数据源，自服务画布上面也会添加一个节点，节点名称与选择的表一致 。


数据
操作
创建数据集 ▼
保存
另存为
测试连接
编辑参数
刷新参数
取消
关闭
检测性能
自动布局
输入搜索文字
创建内嵌数据
数据源
__CONNECTION__
composite-case
Conn
InfoProcess
Self_service
StoredProcedure
YHSQL
数据源
行业案例
DB2
derby
mongo
mysql
mysql_tree
数据源/mysql_tree/celina/表/nation
数据集结果

导入 Excel 数据

在左侧的数据集资源树上，选择新建数据节点 > 拖拽导入 Excel 数据到画布区域，上传 Excel、CSV、TXT 或 LOG 文件，保存并应用。数据集资源树上对应的位置会生成相应的 Excel 数据集，自服务画布上面也会生成 Excel 数据集节点，节点名称与资源树上数据集名称一致。如下图所示 .

创建内嵌数据

在左侧的数据集资源树上，选择新建数据节点 > 拖拽创建内嵌数据到画布区域，设置好数据之后，保存并应用。在左侧的数据集资源树上对应的位置会生成一个对应的内嵌数据集，自服务画布上面会生成对应内嵌数据集节点，节点名称与资源树上数据集名称一致。

关联 & 转换节点

关联节点分为 联接和联合节点。

转换节点分为 逆透视表，分组和汇总，自循环列，镜像，去重节点。

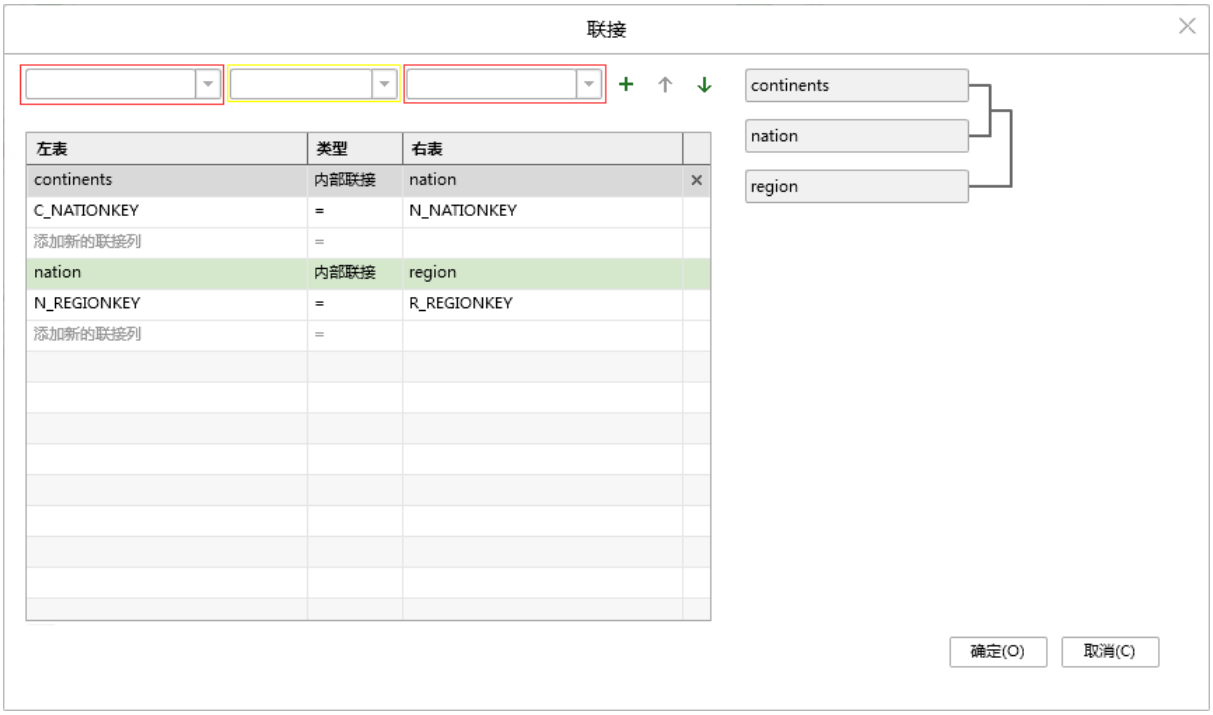
关联节点可以连接多个输入节点。只能有一个输出，镜像节点除外，能连接的镜像节点可以是（n>1）个。

关联和转换节点的右键菜单包含，编辑节点，重命名，删除，刷新（镜像节点除外，镜像节点右键菜单没有编辑节点选项）。关联，转换节点第一次连接其输入节点时编辑都会自动弹框。

联接（Join）

拖拽数据库中的表或视图到自服务数据集。拖拽联接节点到画布时，如果联接节点和输入节点相距75px 之内，联接节点会自动和输入节点连线。联接节点第一次和输入节点自动连线时，会自动弹出编辑节点菜单。如果数据库表之间定义了外联接信息，联接节点会基于这些外联接信息将缺省的联接做好，而无需用户定义，反之，则需要用户自己定义节点之间的联接关系。

双击联接节点或选中联接节点右键编辑节点，打开联接节点对话框进行编辑，通过中间的下拉菜单选择“操作类型”，来定义两个节点之间的联接关系。两个节点之间的关系图：



【输入节点列表】红线标注的下拉列表里罗列的是与联接节点进行连线的所有输入节点。左边的为左表，右边的为右表。

【联接类型】黄线标注的下拉列表可以选择联接类型，有内部联接，左侧联接，右侧联接，外部联接 。

内部联接 （Inner Join): 将左右两张表符合联接条件的记录组合在一起。

左侧联接 (Left Join): 左表的记录全部显示，右表只会显示符合联接条件的记录，右表中记录条件不足的地方补空。

右侧联接 (Right Join): 右表的记录全部显示，左表只会显示符合联接条件的记录，左表中记录不足的地方补空。

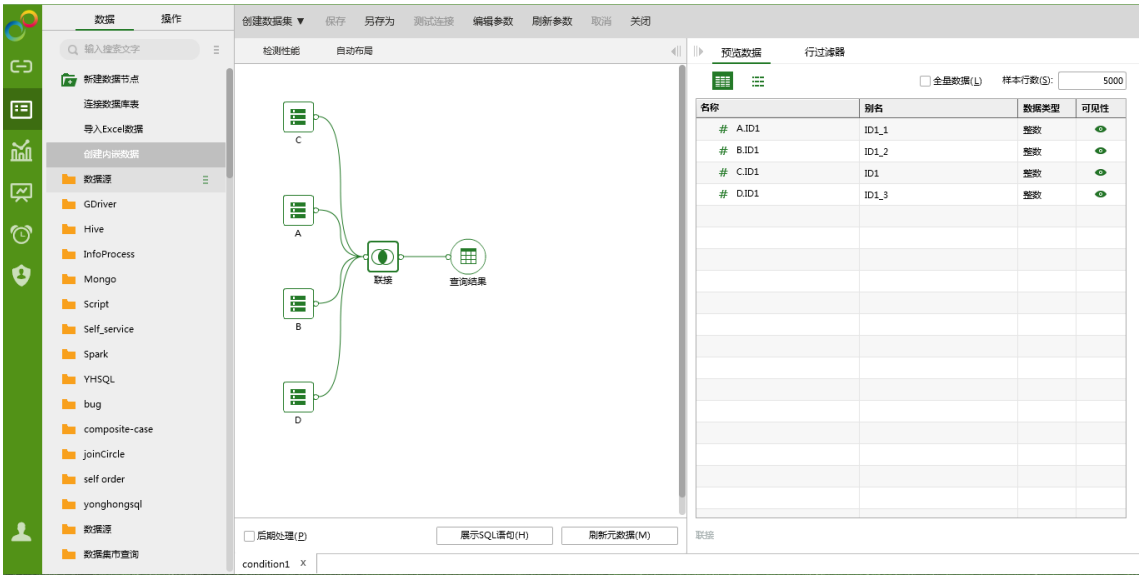
外部联接 (Outer Join): 左右表的记录都全部显示，左表不符合联接条件的记录对应的右表位置补空，右表不符合条件的记录对应左表位置补空。

【添加】选择好左表、右表和联接类型后，点击添加定义的联接列添加到列表中。

【上移 / 下移】列表中选中联接列，可以点击通过上移 / 下移来调整联接列的定义顺序，同时也会对执行顺序产生影响。

【联接执行顺序】右边展示区域显示的是定义的联接关系的执行顺序，点击上移 / 下移，定义顺序也会变化。

联接节点的模型如下：



联合（Union All):

通过连线的方式，将多张表的数据 （包括重复数据行）拼接在一起。

逆透视表（Unpivot Table）

逆透视表节点是将多维转为一维的一种数据处理操作。这是一个多维的数据，如下如所示：

预览数据　行过滤器

显示总行数(G)　预览行数: 1000

Abc ID	# 数学	Abc 班级	# 英语	# 语文
1	100	C1	100	100
2	100	C2	99	100
3	99	C1	99	99
4	99	C1	99	99
5	100	C4	98	98
6	98	C2	99	96

在逆透视节点中，有 2 个列属性设置，保留列和转换列。如下图所示，添加 ID 和班级为保留列，添加数学，语文，英语为转换列。

【数据列】输入节点中所有的可见列。

【保留列】数据保持不变的列。

【转换列】需要从多维转为一维的列。

经过逆透视处理的数据如下所示，转换列数学，语文，英语会从多维数据转为一维显示：

Abc MeaNames	# MeaValues	Abc ID	Abc 班级
数学	100	1	C1
英语	100	1	C1
语文	100	1	C1
数学	100	2	C2
英语	99	2	C2
语文	100	2	C2
数学	99	3	C1
英语	99	3	C1
语文	99	3	C1
数学	99	4	C1
英语	99	4	C1
语文	99	4	C1
数学	100	5	C4
英语	98	5	C4
语文	98	5	C4
数学	98	6	C2
英语	99	6	C2
语文	96	6	C2

分组和汇总（ Aggregate）

分组和汇总是将数据根据分组列做聚合运算。如下图所示，设置分组列为 MARKET，汇总列为 PROFIT。

【数据列】输入节点中所有的可见列。

【分组列】分组的依据列，可以拖拽左侧列到分组列，分组列可以为空。

【汇总列】需要做汇总的列，可以拖拽左侧列到聚合列，聚合列可以为空。

【类型】聚合函数的类型。

计算结果如下图所示：

自循环列（Autoloop）

一个公司部门之间都存在层级关系，每个部门都会有唯一的部门 ID 对应，比如说总部（ID 为 1），技术部（ID 为 11，该层级的第一位为第一层的 ID 值），技术工程师（ID 为 111，该层级的第一位为第一层级的值，第二位为第二层级的值），技术部是总部下面的分属部门，技术工程师是技术部的下属部门，数据库存储部门信息的时候，都是将所有部门 ID 放在一个数据列中，并没有将总部作为一个数据列，技术部作为一个数据列，那么在进行数据分析的时候，如果要显示各个部门的层级关系，并按照层级关系分组显示数据，就没法直接添加维度来实现分组效果。

自循环列就是根据设置自动分出层级关系的列，每个数据集只能创建一个自循环列。一般来说拥有层级关系的 ID 有两种存储方式：ID 长度不一致，ID 长度一致。在本产品中规定，如果想保持 ID 长度一致就必须用 0 补位。

例如：

1. 新建自服务数据集，用普通 query 作为一个输入节点，输入节点如下：

# 员工数	# 部门ID	Abc 部门名称
34	1	总部
3	11	人力资源部
1	111	人力资源经理
2	112	人力资源文员
11	12	开发部
1	121	开发总监
2	122	开发经理
8	123	开发人员
10	13	测试部
1	131	测试总监
9	132	测试人员

2. 在输入节点后面连接一个自循环列节点，在自循环列编辑菜单中，根据一列数据分层，层级长度为 1，设置如下图：

【设置】选择分层列数：根据一列数据分层或者根据两列数据分层。

【层级长度】ID 中多少位代表一个层级。

【ID& 父 ID】父 ID 是 ID 的上一级 ID ；这里只显示度量列。当选择根据一列分层时，就根据层级长度和 ID 对应的列分层；当选择根据两列分层时，就根据 ID 和父 ID 分层。

【分层列】被分层的列。

3. 点击确定，新建出来的自循环列是一个层次列，由于是中间节点，层次列都没有生成文件夹，如图：

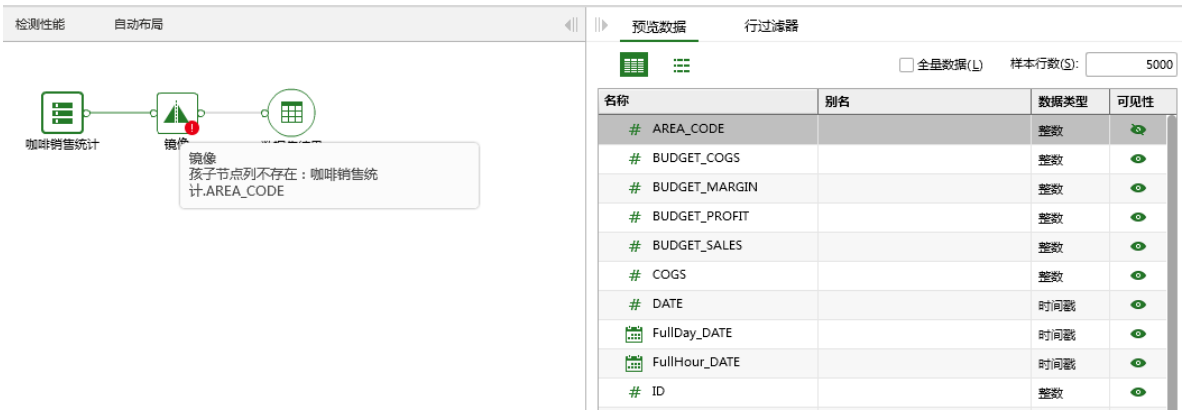
镜像

通过连线镜像节点，任意节点可以被复制一个或多个，源节点改变，镜像跟着改变。

举例说明

1）普通 sql 数据集：咖啡销售统计，后面连接镜像节点，再连上数据集结果节点。

2）在咖啡销售统计 中隐藏其中的一列：AREA_CODE，下一个连接的节点 “ 镜像 ” 会出现检查有效性的红色提示框，改节点后面的连线会变成灰色。鼠标移动到镜像节点时，节点红色图表会提示列不存在的相关信息



3）选中镜像节点，点击鼠标右键 > 刷新

4）刷新之后，镜像节点的元数据和细节数据就会同步，没有列 AREA_CODE。

5）镜像节点传给下一个节点的数据也只剩下 2 列，实现了数据同步。

去重

通过连线的方式，去重节点可以把连线节点的重复记录去掉。

1）新建自服务数据集，添加一个输入节点，输入节点的原始数据如下图所示：

2）连接去重节点之后重复记录被去掉。

数据集结果节点：

节点右键菜单：导入数据库，重命名，显示所有列，隐藏所有列。

数据集结果节点是所有节点数据处理的终结点，数据集结果节点只能有一个输入。

导入数据库

数据集结果可以选择导入数据库，数据集结果节点 > 右键选择导入数据库，如下图所示。

【数据源】选择导出数据的连接数据源。目前支持的数据源为 MYSQL,ORACLE,SQLSERVER,DB2。

【表】输入新表名后，将在数据库中导入新表数据。

【追加】选择追加，导入的数据会添加到该表中而不删除已有的表数据。

节点编辑

【节点的选中】：单击可以选中单个节点，也可以按 Ctrl 键对节点进行多选。或者拖拽鼠标对框选单个或多个节点。

【行过滤器】：选中单个节点，数据区选择行过滤器，打开过滤器编辑窗口，定义过滤器。可以对每个节点单独设置过滤器。

【打开】输入节点是数据集，选中节点，右键菜单选择打开，可以直接打开数据集进行编辑，也可以双击打开数据集进行编辑

【编辑节点】选中联接或转换节点，右键菜单选择编辑节点打开对话框对节点进行编辑，也可以双击打开对话框编辑节点，镜像和去重节点除外。

【重命名】节点右键菜单点击重命名或双击节点的名称，可以对节点进行重命名。

【复制】选中输入节点可以复制。

【粘贴】选择复制后，画布空白处右键可以粘贴，把节点复制一份。

【删除】点击节点右键菜单点击删除，或者使用鼠标框选一个或多个节点点击键盘 delete 键进行删除，能够删除节点以及节点的输入、输出连线，数据集结果节点除外。

【刷新】：点击节点右键刷新，可以更新同步数据。

【隐藏所有列】将元数据中所有的列都隐藏。

【显示所有列】将元数据中所有的列都显示。

节点连线

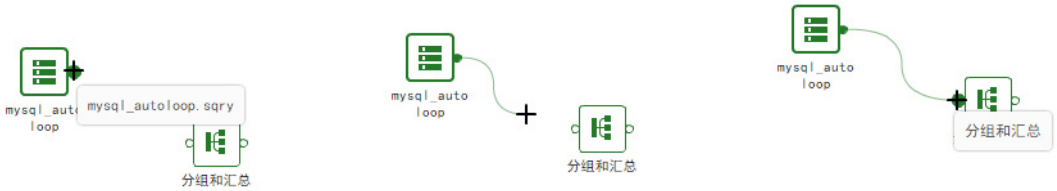
自动连线

以每个节点输入端或输出端所在边框的中点为圆心，在半径为 75px 半圆内区域会触发和其它节点的自动连线。

手动连线

不能自动连线的部分，都可以进行手动连线。

手工连线时，输入输出反馈，当鼠标移至输出端时，空心原点变为实心，鼠标为十字同时会出现提示框；此时点按鼠标并移至下一节点的输入端时，下一节点的输入端原点也会变为实心；如移动到输出端则该原点没有反馈。



删除连线

可以点击连线进行删除；删除节点也会自动删除左右的连线。

性能检测

实时性能检测

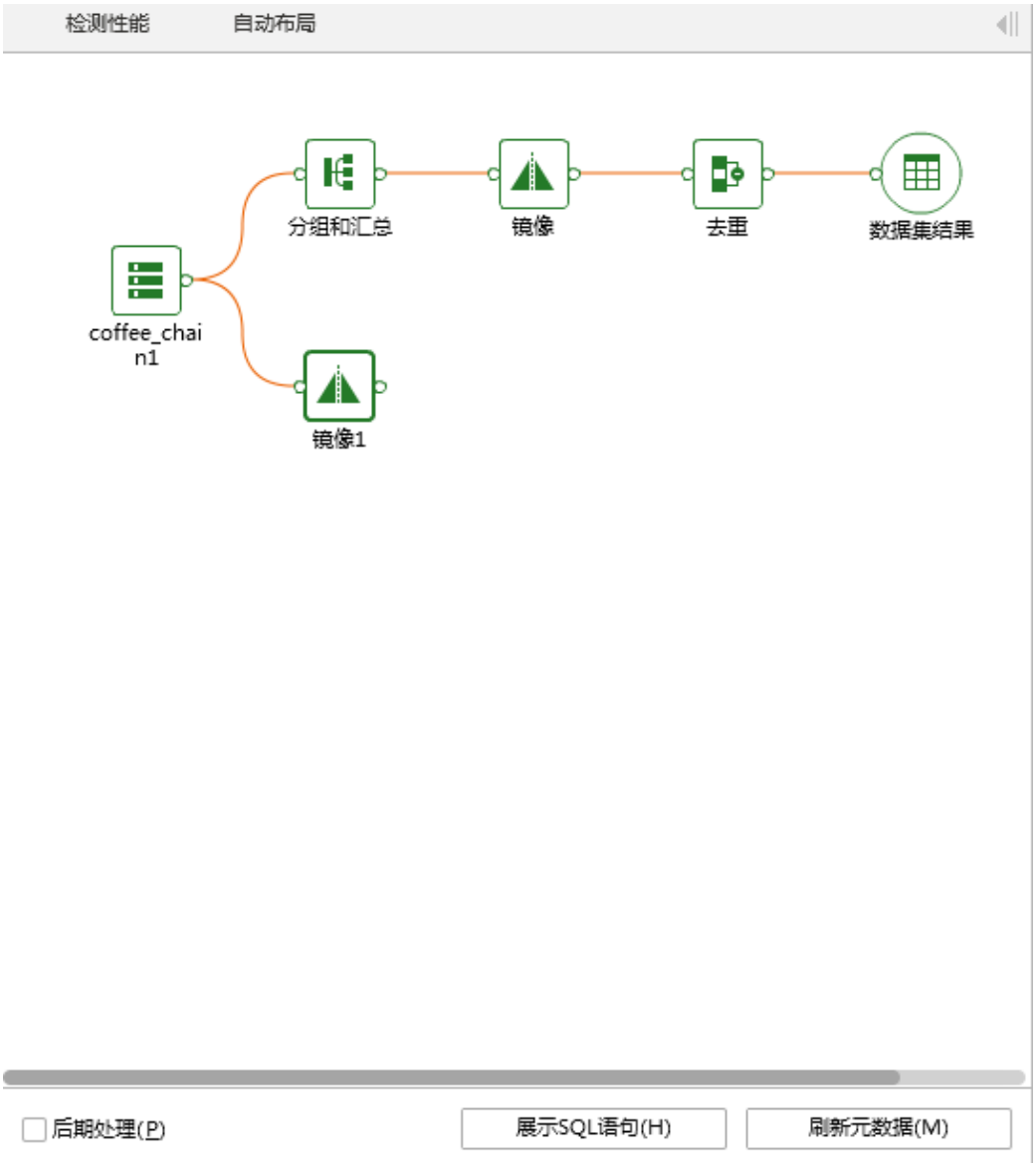
自服务数据准备中，数据执行的绝对快慢很难得知，可以通过连线的颜色进行实时性能检测，连线颜色为绿色时表示性能快，连线为黄色时表示性能慢，用户可以根据性能快慢进行性能调优。

举例说明

1）新建自服务数据集添加 SQL 数据集作为输入节点，在输入节点处添加 script 表达式，内容是 col['marketmysql']+'_ 市场 '。

2）添加一个分组和汇总节点，与输入节点做连线。编辑节点，设置分组列时添加表达式列，确定。

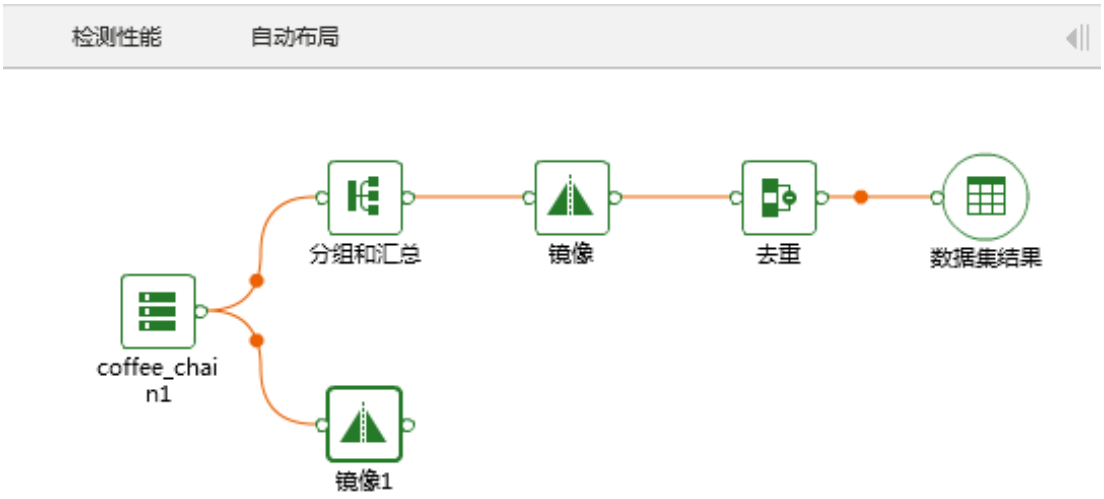
3）因为引用了表达式列为分组列，从引用数据的输入开始，整个链式连线的颜色都会变成黄色。



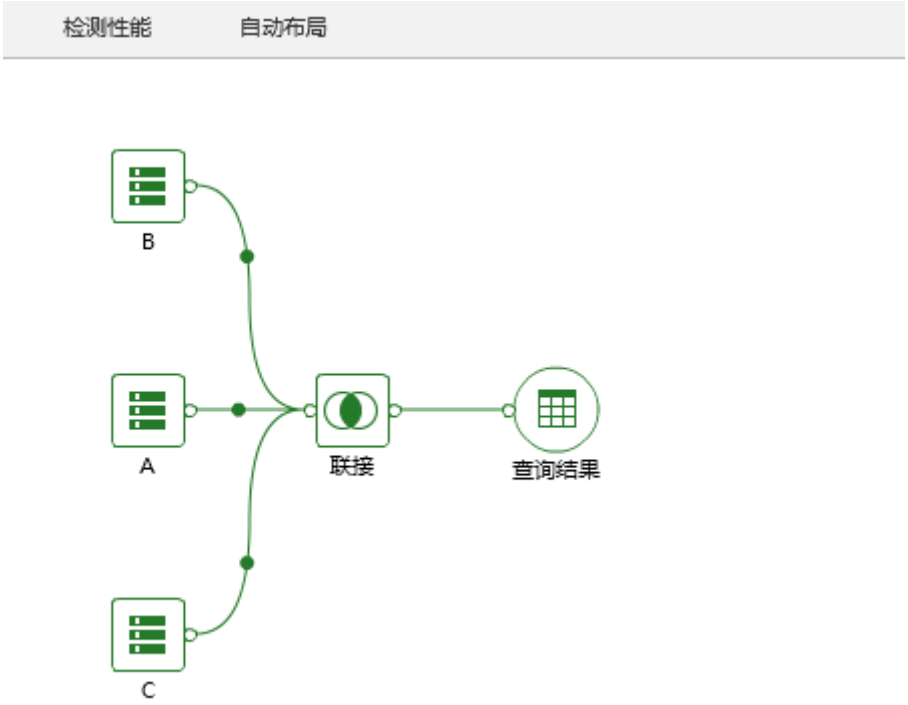
性能检测按钮

点击画布左上角性能检测，小球会沿着输入节点的轨迹，开始滚动，在数据集结果节点停止；小球的颜色跟连线的颜色相同。

多分支时小球会沿着每个分滚动，如下图所示：



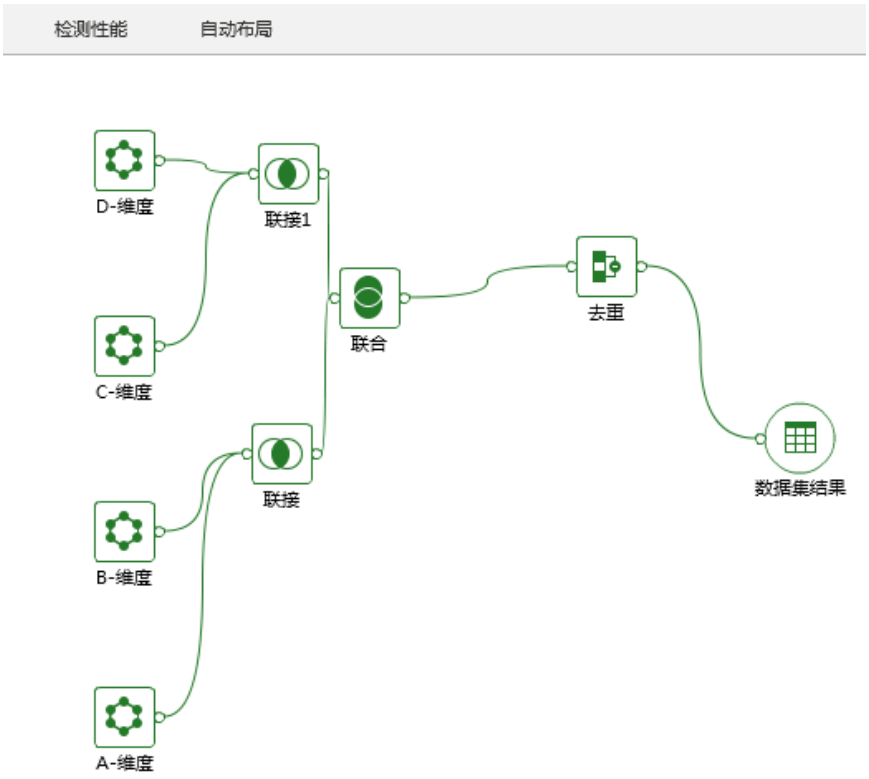
多分支性能检测小球会汇集到一点，如下图所示：



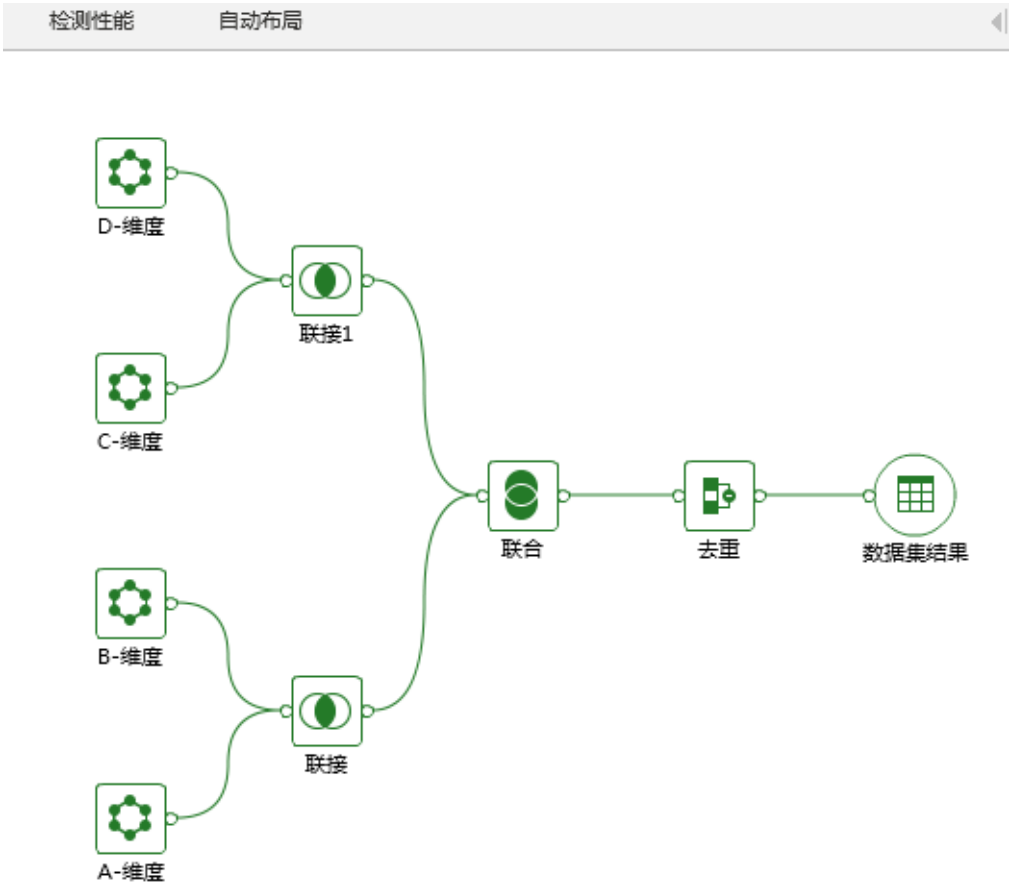
自动布局

点击自动布局，节点和节点之间的布局会根据既定算法优化，布局会更加合理美观。

布局前：


检测性能
自动布局
D-维度
联接1
C-维度
联合
去重
联接
B-维度
数据集结果
A-维度

布局后，格式如下：


检测性能
自动布局
D-维度
联接1
C-维度
联合
去重
数据集结果
B-维度
联接
A-维度

综合应用

@query 的实现

@query 的功能介绍

通过 @query 我们可以实现的功能如下：

1. 如果用户想去动态的查询一些信息，比如我想查询浏览过京东和淘宝的用户的信息，或者我想查询既购买牙刷又购买牙膏的用户的信息的时候，我就可以通过 @query 的方式进行动态的查询数据。

2.@query 可以减少数据集的新建数量，比如想实现 20 个不同数据集的关联，我可能需要创建 20 个数据集，但是通过使用 @query 就可以大大减少数据集的数量，我们可以通过给 @query 动态的传递不同的值就能实现不同的数据集关联的问题。

@query 的定义

@query 是一个动态的数据集，定义它是只需要定义数据库的连接信息就可以，只需要在 SQL 语句中写 ?{@sql} 就可以动态的接收不同的 SQL 语句。

举例说明

举例 1：

1) 定义一个 @query 进行接收 SQL 语句，名称为 aa。

2）创建新的数据集 query，用来进行参数的传递，名称为 query2。

select ID,MARKET,STATE from COFFEE_CHAIN where ID in ?{a} OR ID in ?{b}

如下图：

3）创建 Script 数据集

// 需要制定为 true，这样我们才不会去使用 Meta Columns, 因为 Meta Columns 是静态的，而不是动态的。

param["_DYNAMIC_SQL_"]=true;

// 条件 1

param["sql"] = "select * from COFFEE_CHAIN where STATE in ?{state}";

param["state"] = ["Ohio","Florida","Texas"];

var grid = execute(this, SQL, "aa");

// 将其转化为数组的形式

var size = grid.size(-1); //size 函数是计算行数；

var arr = grid!=null ? toArray(grid,0) : []; // 转化为数组

// 条件 2

```
param["sql"] = "select * from ORDERS where STATE in ?{state1}";
param["state1"] = ["CA","NJ"];
var grid1=execute(this, SQL, "aa");
var size1 = grid1.size(-1);
var arr1 = grid1!=null ? toArray(grid1,0) : []; // 转化为数组
// 将两个不同条件的参数进行传递过去
param["a"]=arr;
param["b"]=arr1;
var grid2=execute(this, SQL, "query2");
// 返回 datagrid 的数据
grid2;
```

4. 运行刷新数据预览就能看到我们想要的数据。

实现数据集市数据集的 JOIN 和 SUB QUERY

在 CLOUD 数据集中不支持 JOIN 和 SUB QEURY，我们需要将数据集进行拆分出来，然后再通过 SCRIPT 数据集进行连接起来。

JOIN 的实现方式：

有两个不同的 query1 和 query2，然后通过他们之间的共同的字段进行 join, 并且写出保留字段。

SUB QUERY 的实现方式：

新建一个 query1，然后在新建一个 query2，通过 query2 去访问 query1 进行实现 SUB QUERY 的功能。

举例说明：

例子 2：

所有网站的 UV 指标和日均 UV 指标。这个例子就包括了数据集市数据集如何实现的 JOIN 和 SUB QUERY。

具体的 SQL 语句如下：

```
Select  ta.Site_ID, ta.UV, tb.AvgUV
From (
  Select Site_ID, count(distinct panel_id) as UV
  From webdata
  group by site_id
)ta inner join
(
  Select Site_ID, Avg(UV)as AvgUV
  From (
     Select Site_ID, Dwd_date, count(distinct panel_id) as UV
     From webdata
  Group  By Site_ID, dwd_date
)tb on ta.Site_ID  = tb.Site_ID
```

在我们产品中的运行方式如下：

1. 新建一个 query， Select Site_ID, Dwd_date, count(distinct panel_id) as UV from webdata group by Site_ID, Dwd_date

2. 再新建一个 query1，

Select Site_ID, Avg(UV)as AvgUV from query1 group by Site_ID

3. 新建一个 query2，

Select Site_ID, count(distinct panel_id) as UV From webdata group by Site_id

4.script query 把 query 和 query2 合并。

var lt = execute(this, SQL, "query");

var rt = execute(this, SQL, "query2");

var lkeys = [0];

var rkeys = [0];

var lcols = [0,1];

var rcols = [1];

join(this, FINAL_JOIN | LEFT_MAIN, LEFT_JOIN, lt, rt, lkeys, rkeys, lcols, rcols);

刷新预览就可以得到想要的数据。

第 3 章：数据类型和字段类型

数据源的数据有数据类型的概念。根据数据类型不同，数据的展现，对话框的输入，格式的使用，都不同。 Yonghong Z-Suite 中支持的数据类型是可以在创建数据集模块上定义或修改的。数据源的数据可能是数字类型，但数据只有 0 和 1，可从业务逻辑上表明此字段是布尔类型，就可以在界面上改成布尔类型。

数据类型分好了，还需要把其分成两类：维度和度量。一般维度用来给数据分组的，是表示一个方面；而度量是用来做汇总统计的数值。通过不同的方面，来分析一些数值的指标，这样数据就被组合成立方体。对立方体的数据过滤，可以切成小立方，和一个平面片。对应 BI 的 OLAP 概念里得切片和切面。

维度是可以被划分成层次。根据层次的高低，用户可以向上、向下钻取。另外日期类型的层次较特殊，界面上创建日期类型的层次来实现。数字范围也是维度的一种字段，用户自定义出一种划分数据范围的字段，把数据范围作为维度来分析数据也是很常用的使用。另外，用户还可以自定义出各种字段来达到分析需求，包括新建的表达式字段，日期层次的字段，数字范围的字段。还可以在制作报告界面定义维度指标计算器，细节指标计算器，聚合指标计算器，动态计算器。

现支持的数据类型包括以下几种：

数据类型	说明	范围
String	字符串	多个任意字符
Char	单个字符	单个任意字符
Boolean	布尔	True，false
Long	长整数	64 位，范围：从 -2^{63} 到 $2^{63}-1$
Short	短整数	16 位，范围：从 -2^{15} 到 $2^{15}-1$
Integer	整数	32 位，范围：从 -2^{31} 到 $2^{31}-1$
Byte	字节	8 位，范围：从 -2^{7} 到 $2^{7}-1$
Float	单精度浮点数	32 位
Double	双精度浮点数	64 位
Date	日期	日期：YYYY-MM-DD
Time	时间	时间：HH:MM:SS
Timestamp	时间戳	日期 + 时间：YYYY-MM-DD HH:MM:SS

各数据类型除了有数据存储和展现上的区别外，在使用的时候还有一些区别。例如根据类型不同，默认是维度还是度量，画图表时轴的类型，能支持聚合函数的类型，等方面也有不同。

维度

维度表示数据分类的角度或方面。城市是一个维度，月份是一个维度，数据范围是一个维度。多维度思考，符合人类习惯。常用的是 3 维立体思维。 3 维可以构成一个立方体。 Slice 是一个立方体的切片。例如一月的所有城市的数据，构成一个面。 Dice 是一个立方体的切块。例如一月份的北京的数据，构成一个小立方体。也可以把维度理解成组的概念，把这些方面分组，把数值类型做统计汇总。常用来做维度的数据类型包括：

数据类型 / 字段类型	说明
String	字符串
Char	单个字符
Boolean	布尔
Date Hierarchy	日期的所有层次
Numeric Range	数据范围
Other	其他非数字和非日期的类型

维度的排序功能更有意义，特别是高级排序可以支持基于别的字段的聚合后的排序，还可以进行排名。细节数据不具备此功能，只有聚合数据才能排名。

维度可以为转变成度量，连接数据模块的界面可以修改，只是整个 data set 级别的改动，所有使用报表的地方，都按照数据集里的划分来判断维度还是度量。还有一个地方可以在维度和度量之间转换，就是在对象的绑定界面上，这个转换只是对象本身起作用。但维度字段切成度量后，支持的统计函数只有求最大值，最小值，统计个数，统计不同值个数，和近似不同值的个数。因为对非数值类型字段做求和等统计没有意义。但是取最大，最小，统计个数，统计精确不同值计数，和不同值计数还是有意义的。

函数名	用途
Count	返回数据集中的数据个数
Accurate Distinct Count	返回数据集中不同值的数目
Distinct Count	大数据的基数估计算法
Max	返回数据集中的最大数值
Min	返回数据集中的最小值

度量

度量表示可被用于衡量和统计的数值，销售，利润，成本等都是度量。以此，数值类型的数据都被划分为度量了。另外把日期，时间也划分为度量。常用来做度量的数据类型包括：

数据类型	说明
Long	长整数
Short	短整数
Integer	整数
Byte	字节
Float	单精度浮点数
Double	双精度浮点数
Date	日期
Time	时间
Timestamp	日期 + 时间

度量同样也可以变为维度，转变规则如上。

度量支持的统计函数较多，支持所有本产品提供的统计函数。

函数名	用途
Sum	返回数据集中所有数据之和
Count	返回数据集中的数据个数
AccurateDistinct Count	返回数据集中不同值的数目
Distinct Count	大数据的基数估计算法
Max	返回数据集中的最大数值
Min	返回数据集中的最小值
Range	返回数据集的范围
Average	返回数据集中的平均值
Product	返回一组数据的乘积
Median	返回给定数值集合的中位数
Quartile	返回一组数据的四分位点
Mode	返回在某一数组或数据区域中的众数
Sum Square	返回一组数据的平方和
Pth Percentile	返回数值区域的 P 百分比数值点
Variance	返回一组数据的方差
Population Variance	返回一组数据的总体方差
Standard Deviation	返回一组数据的标准差

Standard Error	返回一组数据的标准误差
Population Standard Deviation	返回一组数据的总体标准差
Sum Weight	返回一组数据的权重之和
Weight Average	返回一组数据的权重的均值
Covariance	返回一组数据的协方差
Correlation	返回一组数据的相关系数

需要自动把数据类型分成维度还是度量分类的地方包括：

1）刷新元数据。在数据集建立好后，刷新元数据的时候，需要把数据集里的所有字段自动分成维度和度量，分配规则如上。分好后，会分别在元数据界面的两个节点上列出来。

2）新建表达式。在元数据界面上新建表达式，即用脚本生成字段，此时需要选择数据类型。会自动列到对应的节点上。

3）创建了层次，日期层次，数字范围，新建分组，值映射、去空格，拆分列会自动列到维度上；缺失值填充根据原字段数据类型来判断，如果是字符串会自动列在维度上，如果是数值类型会自动列在度量上。

4）在制作报告的时候，创建表达式字段。在报表的绑定界面上，不会根据数据类型来分，而是根据选择的表达式字段的类型来分。细节维度字段是维度。细节度量字段和聚合度量字段是度量。

文件夹

用户可以在数据集编辑界面创建文件夹，按照需求将字段拖拽到文件夹中，便于对字段进行分类。通过维度和度量字段创建的文件夹分别对应存放在维度区域和度量区域。当字段较多时，通过创建文件夹可以使界面看起来更有层次感，展示起来更清晰。

举例说明

某一 SQL 数据集中存在 BUDGET_COGS,BUDGET_MARGIN,BUDGET_PROFIT,BUDGET_SALES 这四个与预算相关的字段，如下图所示：

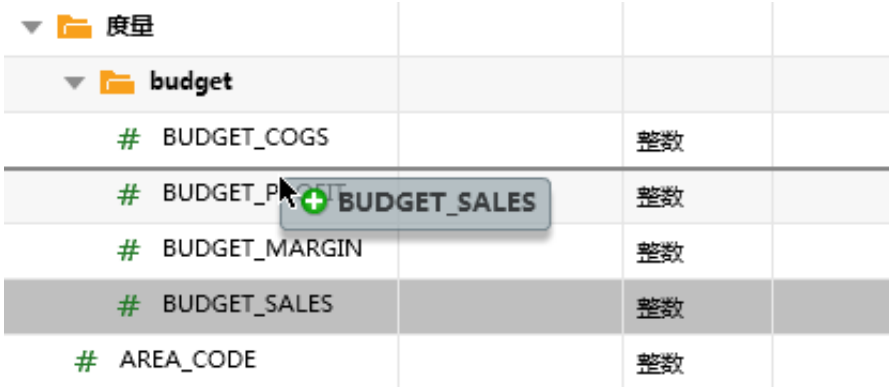
用户在元数据区域右键选择新建文件夹，如下图所示：

在弹出的对话框中，中文环境下会默认名称是“文件夹”，修改文件夹名称，点击确定按钮则在元数据区域生成文件夹。

拖拽鼠标把相应字段放到文件夹中，如下图所示：

被拖拽到文件夹中的字段不可以通过鼠标的拖拽来调节位置。

在 bi.properties 中配置属性 manual.sort.repository=true，可以通过拖拽鼠标调整维度、度量、各个文件夹下列的排序，如下图所示



层次

维度的范围有大小的概念，例如国家的范围大，省的范围次之，城市的范围更小。可以把范围的大小的概念称之为层次。在维度节点下建立层次目录，把有范围大小的维度通过拖拽放入层次中。范围大的放在最上面，范围小的维度放在下面。维度的顺序，决定了钻取的顺序。当需要上钻时，会找到与当前维度最近的上一个维度（即范围大点的维度）。当需要下钻时，会找到与当前维度最近的下一个维度（即范围小点的维度）。

举例说明

某一 SQL 数据集中存在年、季度、月份字段，三个字段之间有范围大小的概念，年大于季度，季度大于月份，如下图所示。

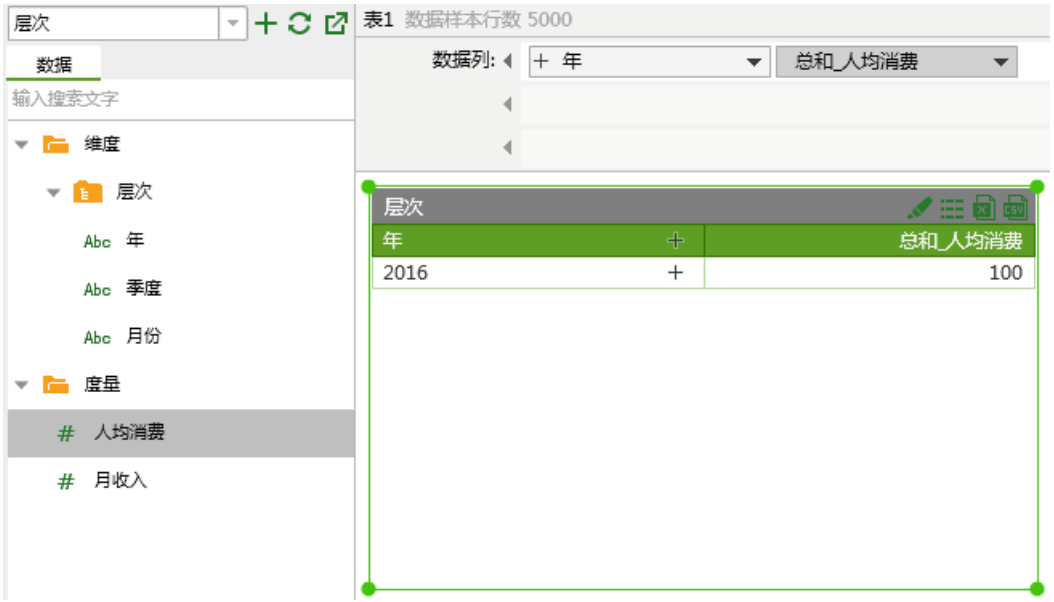
用户在元数据区域右键选择增加层次，如下图所示。

在弹出的对话框中，，中文环境下会默认名称是“ 层次 ”，可以修改层次名称，点击确定按钮则在元数据区域生成层次。

用户通过鼠标拖拽来把具有层次的字段放到层次文件夹中，如下图所示。

被拖拽到层次文件夹中的字段仍可通过鼠标的拖拽来调节位置。在上的字段范围较大，如下图所示。

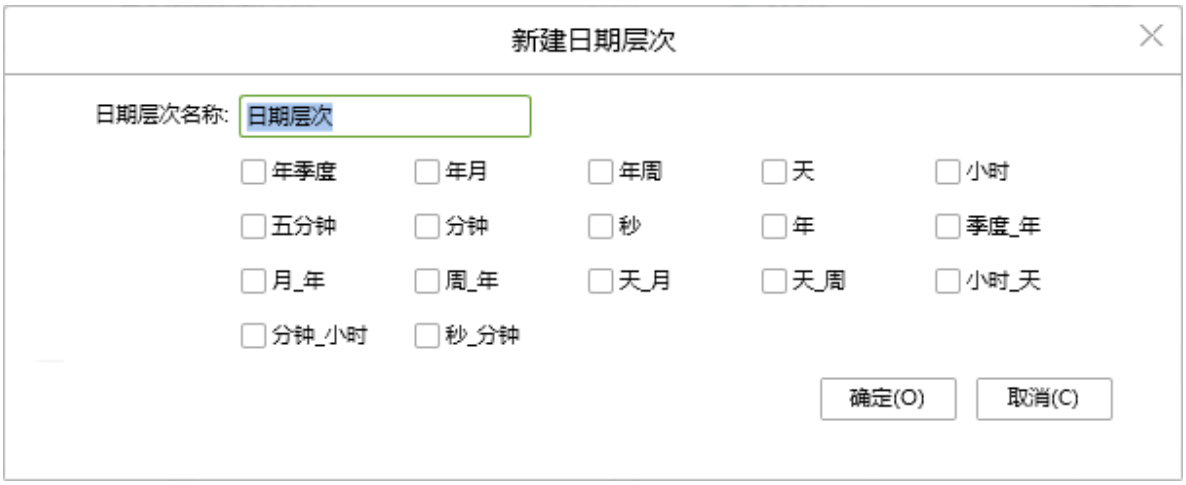
当用户在报告编辑区中绑定该层次文件夹中的字段时，会在范围较大的数据段的左侧生成加号，当用户点击此加号时，比当前数据段的范围小一级的数据段将被自动绑定，同时加号变成减号。


层次
数据
输入搜索文字
维度
层次
Abc 年
Abc 季度
Abc 月份
度量
人均消费
月收入
表1 数据样本行数 5000
数据列:
年
总和_人均消费
层次
年
总和_人均消费
2016
100

日期层次

用户可在日期、时间、时间戳类型的数据段上右键，选择增加日期型层次，来创建用户数据段。

打开的日期型层次对话框如下图所示。会默认名称是“日期层次”，用户根据需要修改日期型层次的名称，以及勾选需要创建的日期型列。



日期是一个典型的层次结构。当选择了某一日期字段时，可以给该字段建立一个层次，并选择需要建立的日期维度。支持的日期维度如下表所示。

日期维度	数据类型	说明
年季度	Timestamp	季度，把所有本季度的，映射到本季度的第一天
年月	Timestamp	月份，把所有本月的，映射到本月第一天

年周	Timestamp	周，把所有本周的，映射到本周第一天
天	Timestamp	日，把所有本天的，映射到本天零点
小时	Timestamp	小时，把所有本小时的，映射到该小时的起点
五分钟	Timestamp	五分钟，把所有本五分钟的，映射到该五分钟的起点
分钟	Timestamp	分，把所有本分钟的，映射到该分钟的起点
秒	Timestamp	秒，把所有本秒的，映射到该秒的起点
年	Integer	年份
季度 _ 年	Integer	季度 ,1-4
月 _ 年	Integer	月份 ,1-12
周 _ 年	Integer	每年得第几个星期，1-52 周
天 _ 年	Integer	日，每月的第几日，1-31
天 _ 周	Integer	每个星期的第几天
小时 _ 天	Integer	小时，0-23
分钟 _ 小时	Integer	分 ,0-59
秒 _ 分钟	Integer	秒 ,0-59

年季度、年月、年周、天、小时、五分钟、分钟、秒，均把相应的日期、时间、时间戳数据段映射成时间戳类型。年、季度 _ 年、月 _ 年、周 _ 年、天 _ 年、天 _ 周、小时 _ 天、分钟 _ 小时、秒 _ 分钟则把日期、时间、时间戳数据段映射成整数类型。

例如一时间戳类型数据为 2012-11-02 11:34:25，进行映射后的数据如下表所示。

映射前的数据	映射后的数据
年季度	2012-10-01 00:00:00
年月	2012-11-01 00:00:00
年周	2012-10-28 00:00:00
天	2012-11-02 00:00:00
小时	2012-11-02 11:00:00
五分钟	2012-11-02 11:30:00
分钟	2012-11-02 11:34:00
秒	2012-11-02 11:34:25
年	2012
季度 _ 年	4
月 _ 年	11
周 _ 年	44
天 _ 年	2

映射前的数据	映射后的数据
天 _ 周	6
小时 _ 天	11
分钟 _ 小时	34
秒 _ 分钟	25

在已创建的日期型层次上右键，可对其进行重命名、删除。

数据范围

数据范围是给一个数字类型字段创建一个划分范围的维度字段。因此，此字段会自动列入维度的节点下。可以在元数据界面上，选择一个数字类型字段，右键选择新建数据范围，如下图所示：

也可以在细节数据界面，选择一个数字类型字段列头，右键选择新建数据范围

在打开的对话框中，用户可修改数据范围的名称、类型、边界、最小值、最大值、步长。

【原始字段】创建数据范围所使用的字段。

【名称】数据范围的列名，默认名称是“ 数据范围 ”。

类型为范围：

【最小值】设定数据范围的最小值。

【最大值】设定数据范围的最大值。

【步长】设定数据范围的步长值。

【包含小于最小值的范围】当用户不勾选时，则小于最小值的值将被映射成空。当勾选上时，小于最小值的值将被映射成最小值减去步长值。

【包含大于最大值的范围】当用户不勾选时，则大于最大值的值将被映射成空。当勾选上时，大于最大值的值将被映射成最大值加上步长值。

【包含范围的左边界不包含右边界】数据范围包含左边界但不包含右边界。

【不包含范围的左边界包含右边界】数据范围包含右边界但不包含左边界。

类型为分组：

【定义刻度】定义刻度值。

【添加】将定义的刻度添加进去。

【删除】将已添加的刻度删除掉。

【标签】可以给添加的刻度范围设置别名。

举例说明

假设对成绩列增加数字范围，该数字范围的最小值为 60，步长值为 10，最大值为 100，包含小于最小值的范围，不包含范围的左边界包含右边界，如下图所示。

在表上的映射结果如下图所示。因为勾选了“ 包含小于最小值的范围 ”所以小于 60 的成绩被映射成 50。勾选了“ 不包含范围的左边界包含右边界 ”，所以 70 映射成了 60，否则被映射成 70。

成绩	数字范围
55	50
60	50
65	60
70	60
78	70
80	70
82	80
90	80
95	90
100	90

新建生成数据范围列后，会自动选中该列，如果有纵向滚动条会滑动到新列的位置并选中新列。

在已创建的数据范围数据段上右键，可对其进行编辑、删除。

表达式

表达式是指在数据集编辑界面，创建新的表达式字段。作用域是当前数据集，所有使用该数据集的报表都能使用此字段。在元数据界面上选中列或者在细节数据界面上选中字段的列头，右键选择新建表达式，并输入表达式名称，选择数据类型，并编辑脚本语言来返回结果。脚本中不能使用聚合函数。创建后，可以通过拖拽，放到维度结点下或者度量结点下。

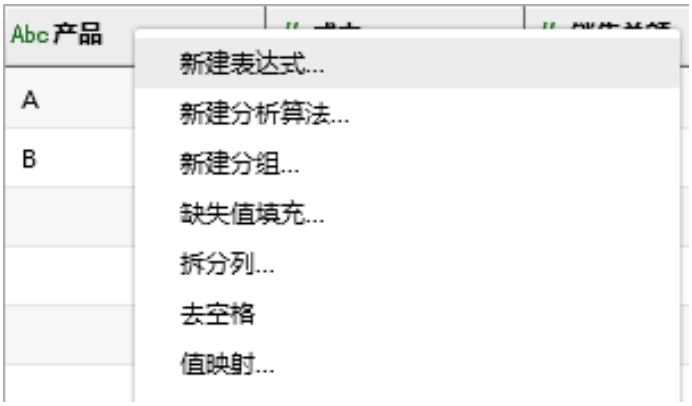
举例说明

1.SQL&JS 表达式

在元数据区中存在下图中的字段，用户需要创建一个利润字段，该字段的计算方式为销售总额字段减去成本。

在元数据区域右键选择新建表达式，如下图所示。

在细节数据界面，选中列的列头，右键选择新建表达式。



在弹出的表达式对话框中默认名称是“ 表达式 ”，修改名称，数据类型以及输入脚本进行计算，可通过 SQL 或者 JS 两种方式运行，以 JS 为例。

点击确定按钮后，将在元数据区生成利润字段。

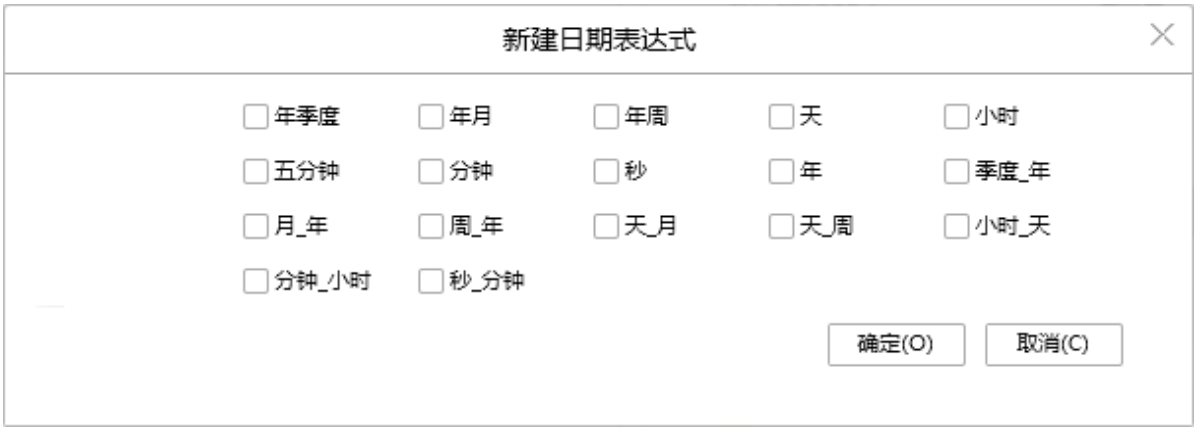
用户在编辑数据库数据集时，如果使用 JS 脚本，会影响数据集性能。产品的实时性能检测机制，会标识出性能不佳的地方。用户也可以通过 【检测性能】按钮，查看所有性能问题。

新建生成表达式列后，会自动选中该列，如果有纵向滚动条会滑动到新列的位置并选中新列。

在已创建的表达式上右键，可对其进行编辑、删除。

2. 日期表达式

元数据中在日期，时间，时间戳类型的字段上右键，或者在细节数据界面选中日期，时间，时间戳类型字段的列头可以新建日期表达式。表达式对话框如图：



新建生成日期表达式后，会自动选中该列（如果生成多个则只选中第一个），如果有纵向滚动条会滑动到新列的位置并选中新列。

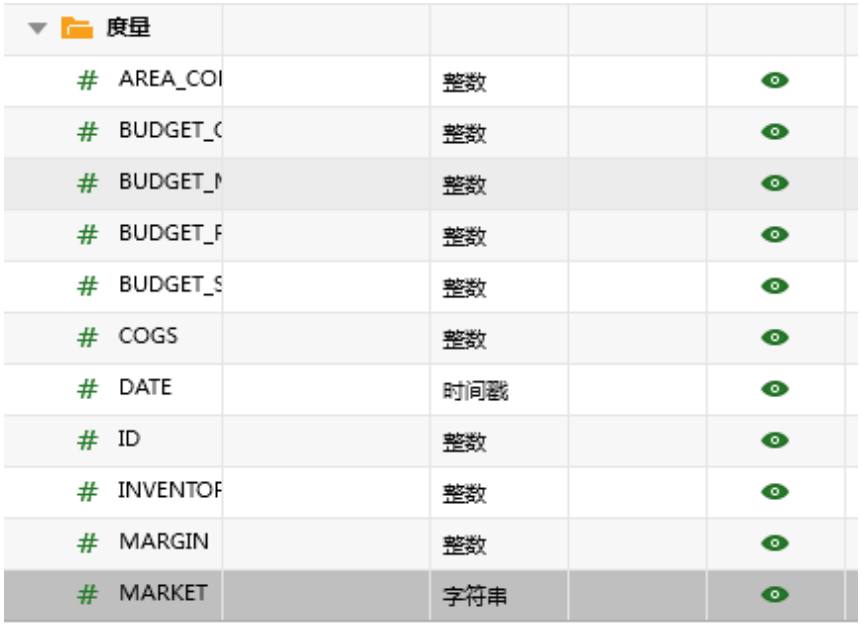
在已创建的日期表达式上右键，可对其进行删除。

这里不再赘述，详细信息请参看本章的日期层次。

转换为度量列

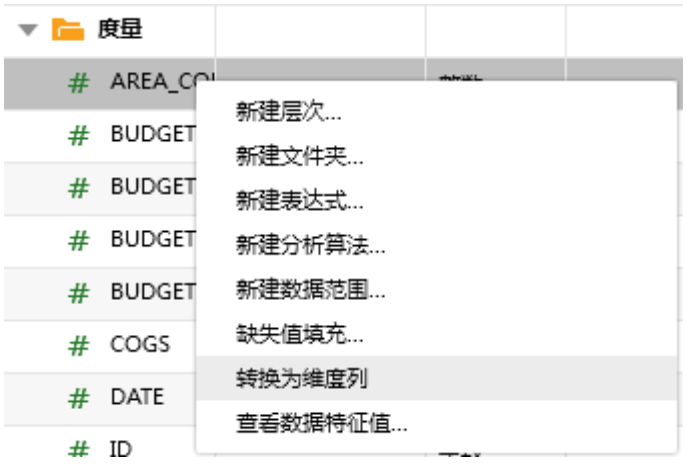
有两种方式可以将维度字段转为度量字段，一种是通过拖拽的方式，即选中维度字段拖动鼠标将字段从维度区域拖拽到度量区域，另一种是在元数据区选中维度字段右键选择转换为度量列，如下图所示。

转换后该列存放在度量区域。



转换为维度列

有两种方式可以将度量字段转为维度字段，一种是通过拖拽的方式，即选中度量字段拖动鼠标将字段从度量区域拖拽到维度区域，另一种是选中度量字段右键选择转换为维度列，如下图所示。



转换后该列存放在维度区域。

转换为日期列

可以将字符串和长整数两种类型的字段转换为日期列。在元数据上选中字段或者在细节数据选中字段的列头右键菜单中可以选择转换为日期列选项，同样在元数据或细节数据中右键可以编辑或者删除新转化的列。

1. 在字符串类型的字段上右键，打开转换为日期列对话框，如图：

【原始字段】需要转换为日期列的字段。

【列名】日期列名称，默认名称是“ 日期列 ”，可以对名称进行修改。

【日期】根据列数据选择相应的格式。例如数据为 2016-01-23，选择的格式应该为 yyyy-MM-dd。

2. 在长整数类型的列上右键，打开转换为日期列对话框，如图：

【原始字段】需要转换为日期列的字段。

【名称】日期列名称，默认名称是“ 日期列 ”。修改列名，点击确定后，根据相应的公式，把数据转化为日期。

举例说明

原始数据如图：

将 “ 字符串 ” 列转换为日期列 “ 日期列 1”，格式为 “yyyy.MM.dd” ；将 “ 长整数 ” 转换为日期列 “ 日期列 2”。转化后数据如下：

新建生成日期列后，会自动选中该列，如果有纵向滚动条会滑动到新列的位置并选中新列。

在已创建的日期列上右键，可对其进行编辑、删除。

转换为数字列

可以将字符串类型的字段转换为数字列，在元数据上选中字段或者在细节数据选中字段的列头右键菜单中可以选择转换为数字列选项，同样在元数据或细节数据中右键可以编辑或者删除新转化的列。

在字符串类型的字段上右键，打开转换为数字列对话框，如图：

【原始字段】需要转换为数字列的字段。

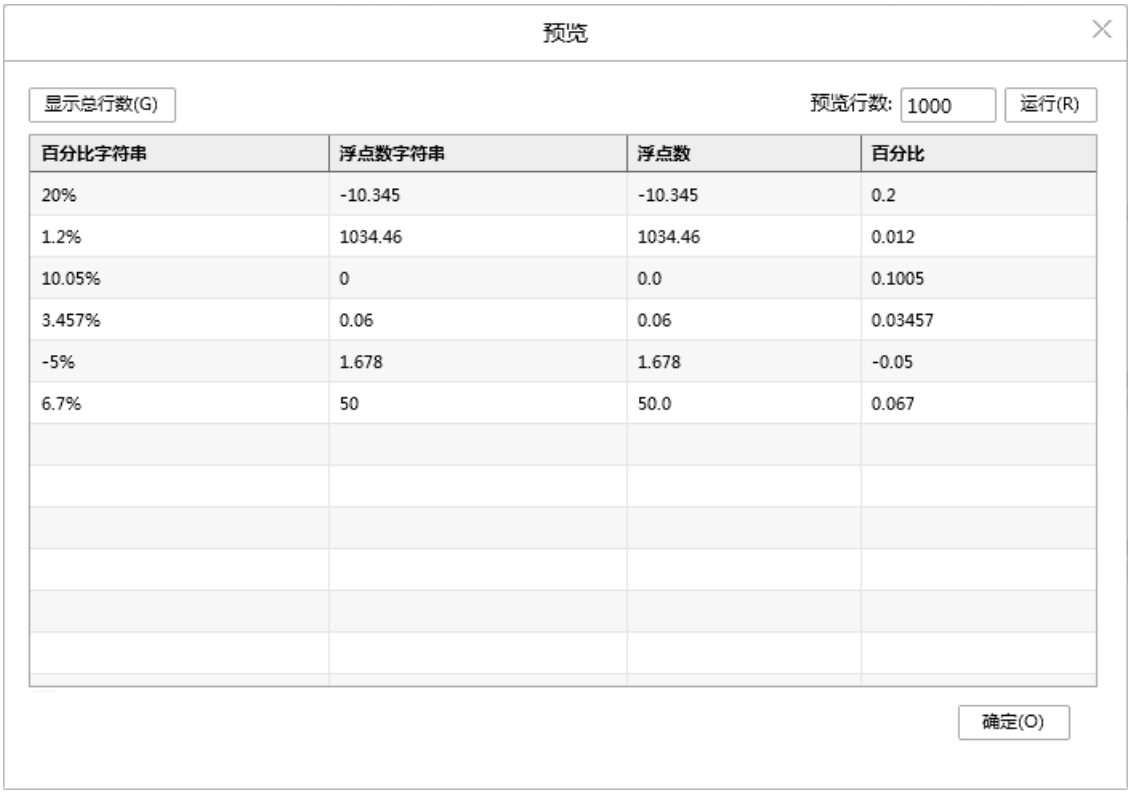
【名称】数字列名称，默认名称是“数字列”，可以对名称进行修改。

【数字 & 货币 & 百分比】根据列数据选择相应的格式。例如数据为 20%，选择的格式应该为百分比。

举例说明

原始数据如图：

将 “百分比字符串” 转化为数字列 “百分比”，格式选择 “百分比” ；将 “浮点数字符串” 转换为 “浮点数”，格式选择 “#,##0.##”。预览数据如图：



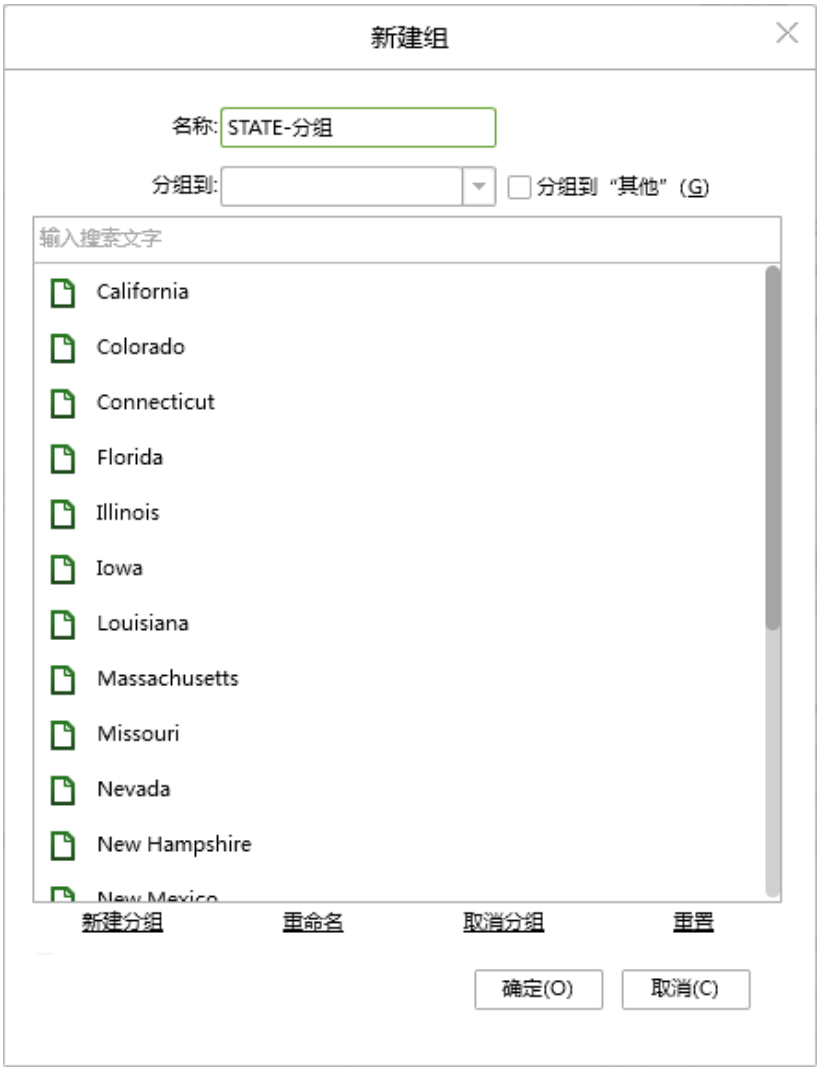
新建生成日期列后，会自动选中该列，如果有纵向滚动条会滑动到新列的位置并选中新列。

在已创建的日期列上右键，可对其进行编辑、删除。

新建分组

选择数据类型是布尔、字符串、时间戳、日期或时间的字段，在元数据上选中字段或者在细节数据选中字段的列头右键菜单中可以选择新建分组，对字段的数据值进行分组，数据列表中空数据和空字符串会被过滤掉。

在字符串类型的字段上右键，打开新建分组对话框，如图：



【名称】新建分组列的名称，默认是 " 原列名 - 分组 ", 可以对名称进行修改。

【分组到】选中一个或多个数据，选择列表下的组名，将数据分组到该组下。

【分组到 " 其他 " 】勾选此选项会生成名称为 " 其他 " 的分组，如果有的数据没有进行分组，则会放到 " 其他 " 分组下面。

【新建分组】点击后会生成分组，默认名称是 分组 1, 分组 2……，新建后光标会选中并能修改分组名称。

【重命名】选中某个分组，点击重命名能够对分组修改名称。

【取消分组】选中某个分组，点击取消分组，当前分组被清除。

【重置】清除所有的分组。

举例说明

原始数据如图：

# date	# 列1
2015-11-01	6
2015-10-05	8
2014-11-06	9
2014-10-06	3
2014-12-06	4
2015-12-05	5
2010-10-01	

对 date 列新建分组，2014 的相关数据分组到 " 2014 年 " 下，2015 的相关数据分组到 " 2015 年 " 下，其余到 " 其他 " 分组下。预览数据如图：

# date	# 列1	f_x date-分组
2015-11-01	6	2015年
2015-10-05	8	2015年
2014-11-06	9	2014年
2014-10-06	3	2014年
2014-12-06	4	2014年
2015-12-05	5	2015年
2010-10-01		其他

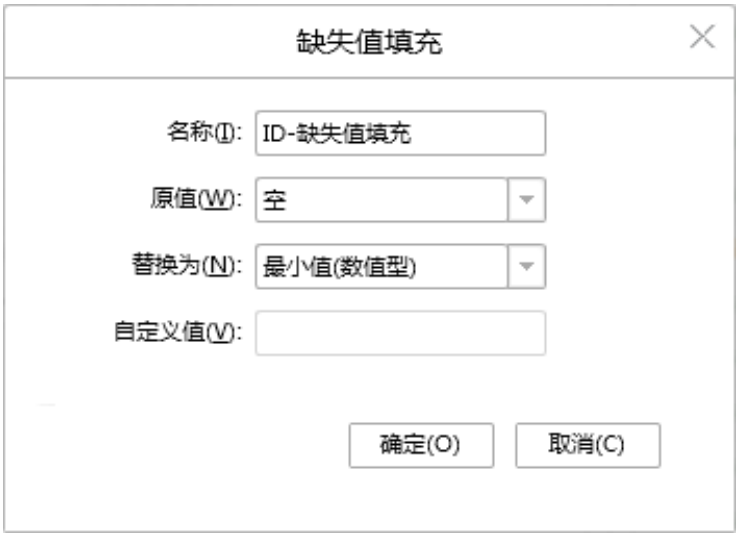
生成分组列后，会自动选中该列，如果有纵向滚动条会滑动到新列的位置并选中新列。

在已创建的分组列上右键，可对其进行编辑、删除。

缺失值填充

选择类型是字符串、数值类型（双精度、单精度、整数、长整数等）的字段，在元数据上选中字段或者在细节数据选中字段的列头右键菜单中可以选择缺失值填充，对字段数据进行填充。

在整数类型的字段上右键，打开缺失值填充对话框，如图：



【名称】缺失值填充生成列的名称，默认是“ 原列名 - 缺失值填充 ”, 可以对名称进行修改。

【原值】要被填充的数据原值。被填充列是字符串类型时，原值包括空、空字符串；被填充列是数值类型时，原值包括空。

【替换为】需要替换的值。被填充列是字符串类型时，替换值包括 -、NULL、自定义；被填充列是数值类型时，替换值包括最大值、最小值、平均值、0、自定义。

【自定义值】替换值选择自定义时，自定义值可编辑，将原值替换为自定义输入的值。

举例说明

原始数据如图：

Abc str	# ID
	1
	2
22	3
	4
	5
66	
77	
soso	33
1212	

将 str 中的空字符串填充为 " 空 " ，ID 中的空值填充为平均值。预览数据如图：

Abc str	# ID	fx ID-avg	fx str-缺失值填充
	1	1.0	空
	2	2.0	空
22	3	3.0	22
	4	4.0	
	5	5.0	
66		8.0	66
77		8.0	77
soso	33	33.0	soso
		8.0	空
1212		8.0	1212

生成填充列后，会自动选中该列，如果有纵向滚动条会滑动到新列的位置并选中新列。

在已创建的填充列上右键，可对其进行编辑、删除。

拆分列

选择类型是字符串的字段，在元数据上选中字段或在细节数据选中字段的列头右键选择拆分列，可以根据分隔符对数据进行全部拆分或部分拆分。

在字符串类型的字段上右键，打开拆分列对话框，如图：

【分隔符】拆分列时所依据的分隔符。

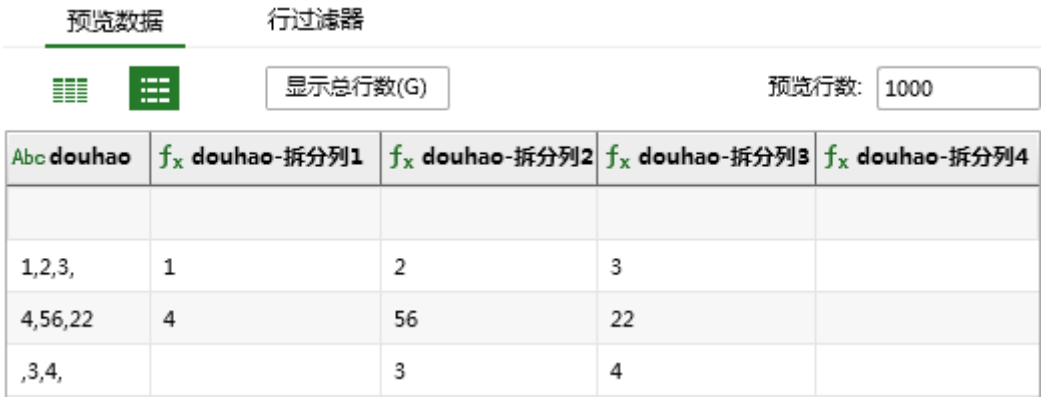
【拆分开始位置】拆分的起始位置，从第一个开始，还是从最后一个，或者是拆分全部

【拆分列数】根据输入值确定拆分后生成的列数，如果拆分开始位置是全部，则不需要输入拆分列数。

举例说明

原始数据如图：

将 douhao 进行拆分，分隔符是逗号，开始位置是全部。预览数据如图：



Abc douhao	fx douhao-拆分列1	fx douhao-拆分列2	fx douhao-拆分列3	fx douhao-拆分列4
1,2,3,	1	2	3	
4,56,22	4	56	22	
,3,4,		3	4	

生成拆列后，会自动选中第一个列，如果有纵向滚动条会滑动到新列的位置并选中新列。

在已创建的拆分列上右键，可对其进行删除。

去空格

选择类型是字符串的字段，在元数据上选中字段或者在细节数据选中字段的列头右键菜单中可以选择去空格，可以去除数据中前、后空格。

在整数类型的字段上右键，打开缺失值填充对话框，如图：

举例说明

原始数据如图：

对去空格列执行去空格操作。预览数据如图：

Abc 去空格	fx 去空格-去空格
a	a
a	a
a	a
a	a
a b	a b
a b	a b
a b	a b
a	a
a	a
ab bb	ab bb
c v	c v
v c	v c
v c b	v c b
vvv	vvv
y uu	y uu

生成去空格列后，会自动选中该列，如果有纵向滚动条会滑动到新列的位置并选中新列。

在已创建的去空格列上右键，可对其进行删除。

值映射

选择类型是字符串、布尔、字符、时间、日期或时间戳的字段，在元数据上选中字段或者在细节数据选中字段的列头右键菜单中可以选择值映射，对字段数据起别名，值列表中会列出空数据和空字符串。

在字符串类型的字段上右键，打开值映射对话框，如图：

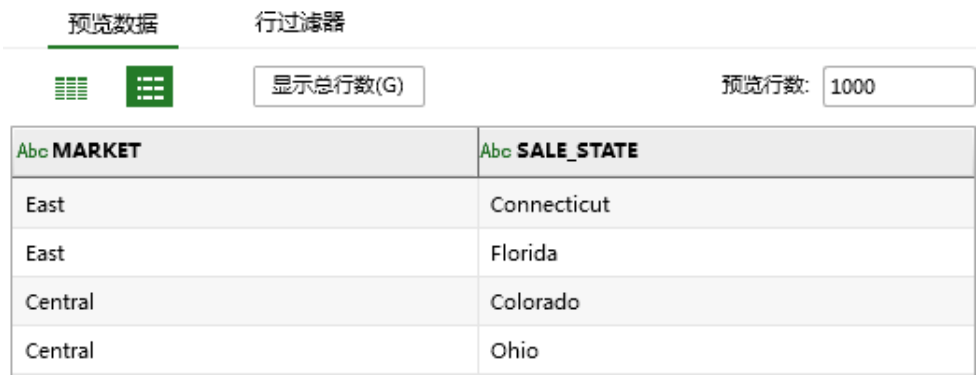
【名称】值映射生成列的名称，默认是 “ 原列名 - 值映射 ”, 可以对名称进行修改。

【值 (别名)】双击输入框后可编辑，能够对字段的值设置别名。

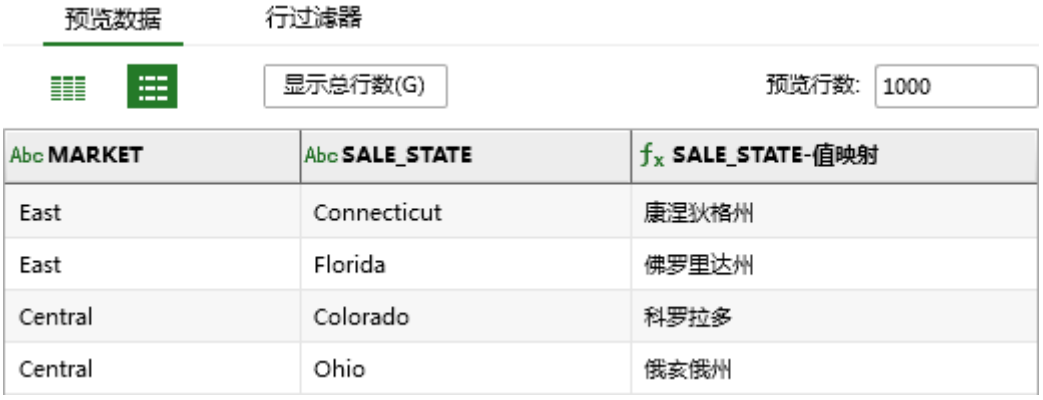
【清空别名】清空给值设置的别名。

举例说明

原始数据如图：



对字符串中的值进行映射，取别名。预览数据如图：



生成值映射列后，会自动选中该列，如果有纵向滚动条会滑动到新列的位置并选中新列。

在已创建的值映射列上右键，可对其进行编辑、删除。

查看数据特征值

选择数值类型（双精度浮点数、单精度浮点数、长整数、整数、短整数、字节）的字段，在元数据上选中字段或者在细节数据选中字段的列头右键选择查看数据特征值，可以计算出字段的最大值、最小值、平均值、中位数。

举例说明

原始数据如图：

选中列 datadata，右键查看数据特征值

点击 计算 按钮，对中位数进行计算

第 4 章：虚拟权限控制（VPM）

Yonghong Z-Suite 支持权限控制模块，在系统中分为两个模块，分别是提供管理用户信息和管理访问权限的功能。用户可以选择权限控制的类型，一种是不需要权限控制，一种是选择产品自带的文件系统权限控制，还有一种是用户自定义得权限控制系统，只要注册两个类（分别实现定义好的 API）就可以。

权限控制模块可以实现控制对资源的认证访问。除了资源级别的权限控制，Yonghong Z-Suite 还支持在数据级别的访问权限控制，用户能配置一个数据集的列对哪些用户可见，一个数据集的行对哪些用户可见。对于一个省的经营分析系统而言，不同地区的用户只能看到本地区的数据，用这个功能很方便的就可以支持。

在制作报表的时候，会遇到不同的用户，根据自己角色的不同来取到和自己相应的数据。例如一般员工不能查看经理能查看的信息。 Yonghong Z-Suite 提供了一种称为虚拟权限控制（VPM）的功能来控制信息的安全。

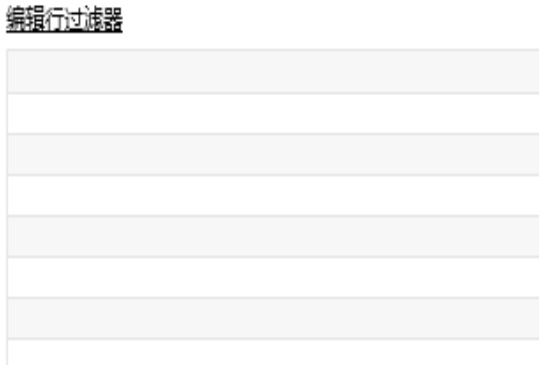
安全

VPM 是一种根据用户及角色来提取数据的控制，这是一种出于安全信息的考虑。那用户和角色信息如何建立呢？ Yonghong Z-Suite 的安全模块提供两种方式来支持管理用户信息：文件权限管理系统和定制权限管理系统。

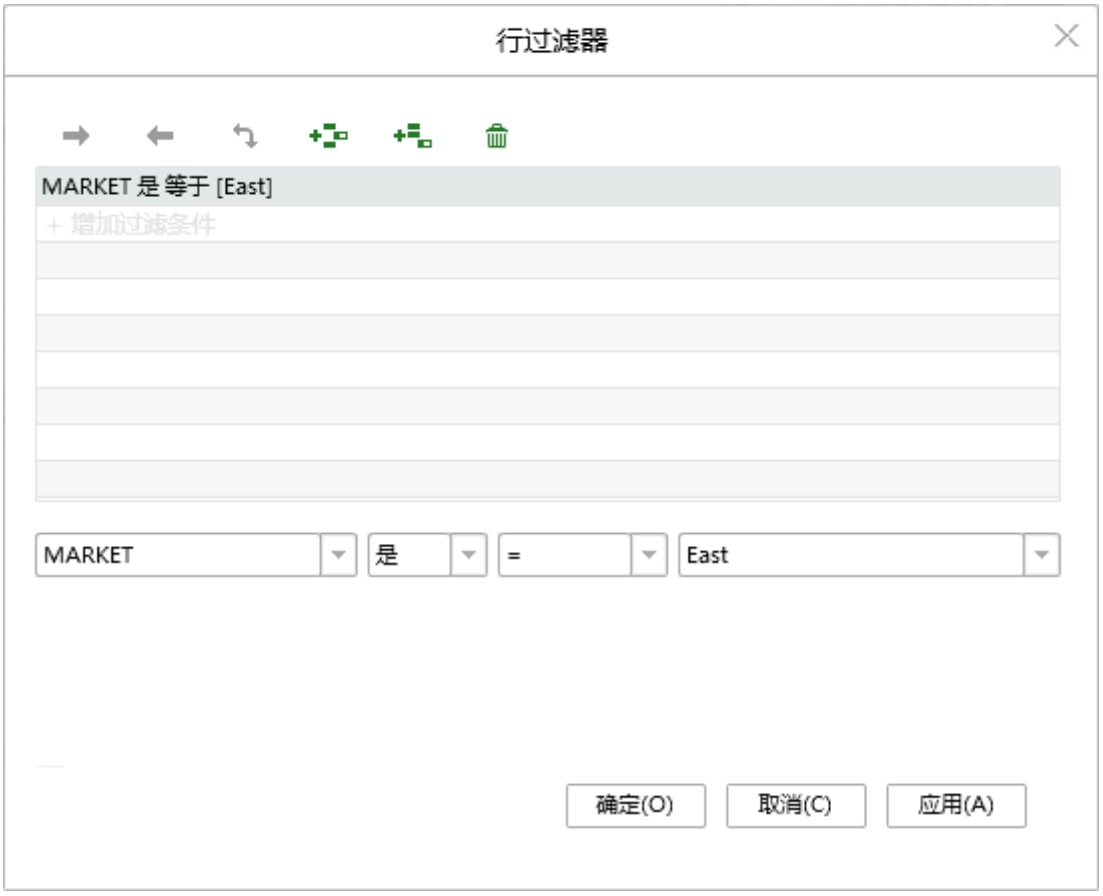
选择好管理系统，并创建好了用户信息，用户所在的组和角色信息后，就可以给需要控制权限的数据集部署 VPM。主要是通过添加行过滤和列过滤来控制权限。一旦部署好了 VPM，就可以放心的把数据集提供给制作报告和查看报告的用户，同时得到完全的数据安全的控制。

行控制

点击编辑行过滤器按钮，打开行过滤器对话框。



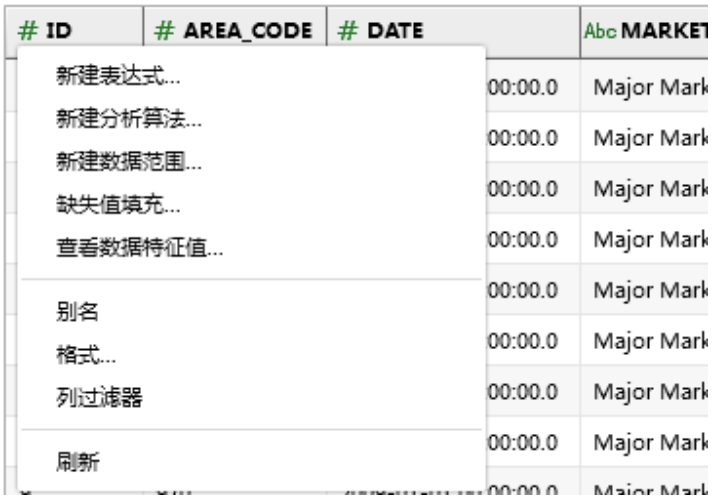
用户在行过滤器对话框中设定过滤条件，如下图所示。



用户可在编辑器中对筛选出符合条件的数据进行编辑。

列控制

在数据集的元数据和细节数据区域，可以对数据集中列的可见和列过滤器进行设置，如下图所示：



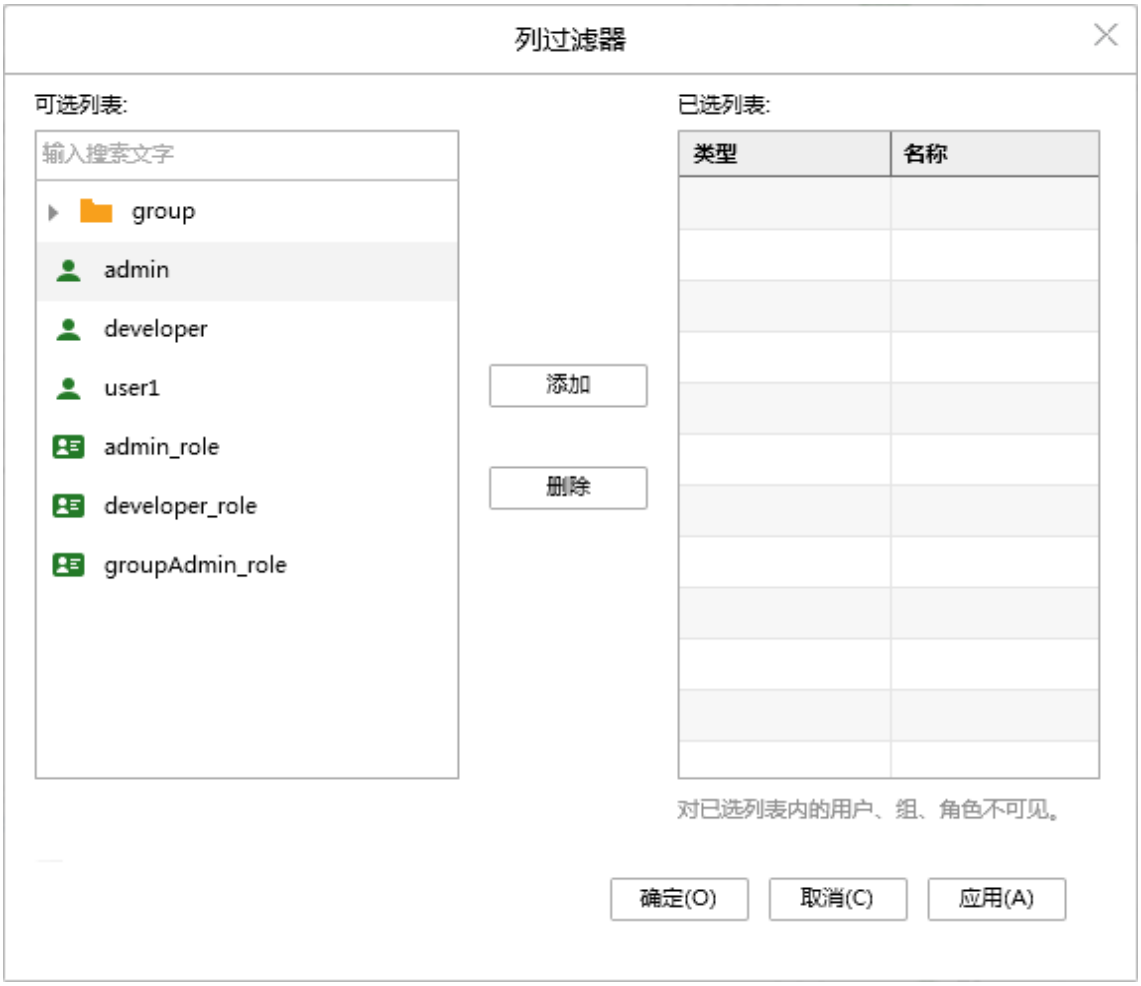
可见性

【可见性】数据集中的列默认为可见状态的，当点击某一列的 “可见” 按钮后，则这列的可见状态被修改为不可见状态，同时，“可见” 按钮上会显示一条向右的斜线，表示此列为不可见。

列过滤器

【列过滤器】在管理系统 - 认证授权 - 安全管理下设置文件权限后，在数据集的元数据区域会显示列过滤器的操作项。列过滤器可以对用户，角色和组设置不可见的权限。

当鼠标点击数据列与列过滤器的交叉处时，将会显示提示文字：编辑，点击编辑，则会弹出列过滤器对话框，如下图所示。



【可选列表】列出了所有可以被设置权限的用户，角色和组。只有具备 admin_role 和 groupAdmin_role 的用户才可以设置列权限。

对于 admin_role 的用户，可用列表会列出所有的用户，组和角色。

对于 groupAdmin_role 的用户，只列出 groupAdmin_role 下的所有用户和组。

【已选列表】添加到已选列表中用户，组或角色在预览数据集或查看报告中不能看到所编辑的列。比如：对 product 列进行列过滤器编辑，将 user1 添加到已选列表中，应用并确定。再用 user1 登录后，在预览数据集和编辑报告中都不能看到 product 列。

【添加】将可用列表中的用户添加到已选列表中。

【移除】将可用列表中的用户移除。

可见与列过滤器的关系

在可见的状态下，可以对用户，组和角色设置列过滤器。设置后，所设置的用户，角色和组对设置列不可见而其他的用户，组和角色不受影响。

在不可见的状态下，列过滤器的对话框为置灰状态的。不能对列过滤器进行编辑。对度量列设置不可见后如果在组件中已绑定该列，打开组件后不会对绑定产生影响，但组件中不会显示该列。

第 5 章：同步数据集数据

同步数据集数据并不是真正意义上的数据查询。它不与链接数据库链接，不进行实际的数据查询，而是将数据库里的数据保存到本地，让保存在文件里的数据通过可操作而变得真实 。它的主要功能是使数据展示和数据操作与其他数据集模块产生的结果一致。

同步数据集数据所使用的数据来源于已经建立好的文件，而不是直接从数据库里读取数据。因此用户会困惑并质疑文件里的数据到底是否能按照数据库的操作方式进行真正的操作。当数据文件通过特别的存储技术被读取并加入到数据集结果的时候，用户可以通过同步数据集数据对数据进行各种操作。此时用户的感觉就像操作真正的数据库一样，这就是同步数据集数据的最重要的作用。

通过任务计划模块制定一个任务，在自定义的时间去执行数据集并把结果存储到部署的硬盘中，基于该数据集定制的报表会优先采用同步数据集数据的结果来展现数据。如果数据库中的实际数据会变化，可以制定一个循环执行的任务来定时更新同步数据集数据的结果。

同步数据集数据可以显著提高数据集的性能，尤其是提高复杂数据集的性能。

目前可通过两种方式对数据集进行同步，一种是直接在数据集界面上进行同步。另一种是通过调度任务中的同步数据集数据进行同步，可定时执行。

直接在数据集界面进行同步

除了数据集市数据集类型不可以同步外，其他数据集皆可同步。当用户需要同步数据集数据时，打开此数据集，在元数据区域点击“同步数据集数据”按钮即可同步该数据集。界面如下图所示：

点击“同步数据集数据”按钮后，弹出同步数据集数据对话框界面，如下图所示：白色的圆圈表示执行过的程序，灰色的圆圈表示还未执行完成的程序。

【后台运行】点击后台运行按钮后，同步数据集数据的对话框关闭，同步数据集数据继续在后台运行。

【取消同步数据】点击取消同步数据后，同步数据集数据的对话框关闭，正在同步的数据集终止。

点击后台运行，鼠标悬停在同步数据集数据按钮后面的小图标上时，可通过 tooltip 查看同步的实时信息

运行完成后，后台运行和取消同步数据按钮分别变为了确定和取消按钮。如下图所示：

同步完成后，原来的“ 同步数据集数据 ”按钮变为“ 释放 ”按钮，当鼠标悬停在释放按钮后面的小图标上时，可通过 tooltip 查看同步后的结果信息，如下图所示：

点击“ 释放 ”按钮，可释放已同步的数据集数据。

通过调度任务界面进行同步

调度任务界面介绍

在调度任务模块中，用户可针对报表和数据集指定不同的任务，然后定期执行此任务。

如用户需要在每月月底检查员工工资情况，则可设定任务，在每月月底把员工工资表以 PDF 格式或 Excel 格式导出到指定地址，或以邮件的形式发送到指定人的邮箱中。

为了提高数据集的性能，用户可在计划任务模块中制定同步数据集数据任务。

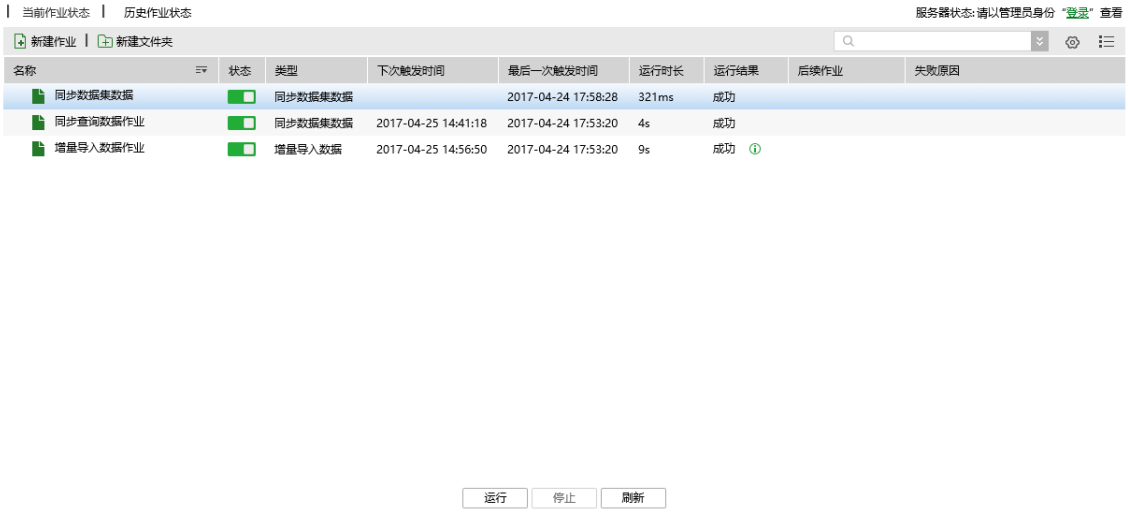
同步数据集数据任务

在调度任务界面同步数据集数据，需要用户先制定同步数据集数据任务，再将任务加入作业中执行。同步数据集数据任务步骤如下：

1）首先用户需要建立任务。比如，任务的名称为：同步数据集数据，任务的类型选择：同步数据集数据，并选择需要进行同步的数据集，界面如下图所示：

2）将任务加到作业中。设置作业的名称为：同步数据集数据，在任务类别中选择：单任务，任务选择：同步数据集数据

3）在作业中执行。选择新建的作业点击“运行”按钮，即可同步数据集中的数据。同步后不能再对数据集中的数据进行修改。



4）查看同步数据集数据的结果。打开已同步的数据集，在数据集界面的元数据区域，显示释放，当鼠标悬停在“释放”按钮后面的小图标上时，可通过 tooltip 查看同步后的结果信息，即：同步数据集数据的作业名称，下次触发时间，最后一次触发时间，运行结果。如下图所示：

对于数据集市数据集，虽然不能同步数据集数据，但增量导入数据任务也可以通过这个小图标来查看运行后的结果信息，如下图所示：

释放同步数据集数据

当用户需要释放掉已经同步的数据集时，在数据集的元数据区，点击“ 释放 ”按钮，即可释放掉已同步的数据集。释放后状态小图标也随着消失。

第 6 章：存储过程

SQL 存储过程（Stored Procedure）是一组为了完成特定功能的 SQL 语句集，经编译后存储在数据库中。用户通过指定存储过程的名字并给出参数（如果该存储过程带有参数）来执行它。存储过程是数据库中的一个重要对象，任何一个设计良好的数据库应用程序都应该用到存储过程。

存储过程的使用

使用产品的同时就是在参数的对话框中进行参数的配置，可以对参数进行配置为不同方向的参数，方向分为 IN、 OUT 和 IN_OUT 三种。参数的方向需要和数据库一致。

DB2 数据库

1）点击 Yonghong Z-Suite 产品的启动快捷方式。

2）打开浏览器，然后在地址栏中输入 http://hostname:8080/bi/Viewer, 登陆到客户端。这里的 hostname 是你的机器名，如果是本机访问，可以用 localhost。 8080 是默认端口号，如果在安装产品时修改了默认的端口号，请采用正确的端口号。

3）输入用户名和密码后登陆到主页面。

4）点击连接数据按钮后，进入到连接数据的界面。

5）如果说在后台创建的语法如下 :

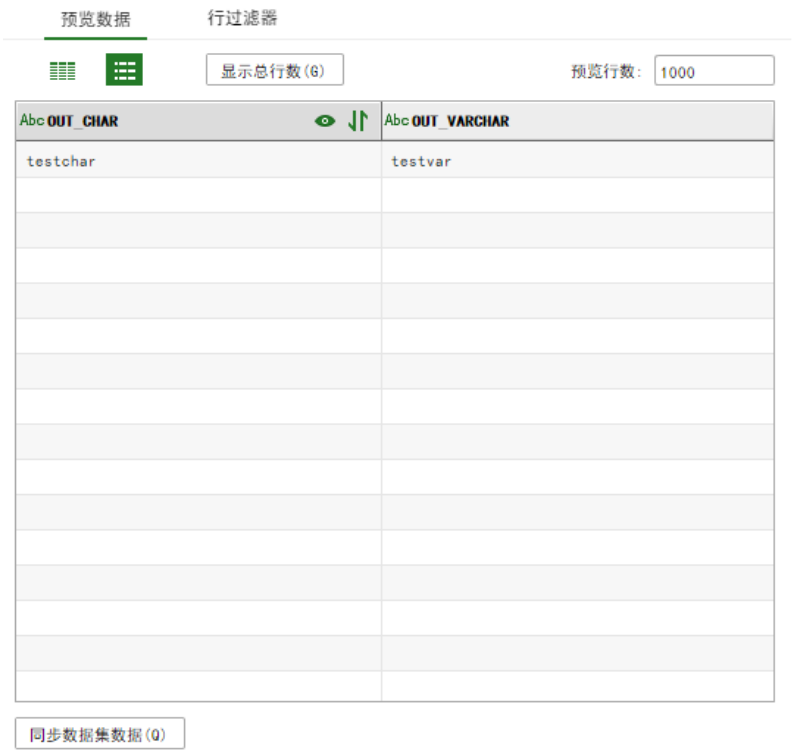
```
create procedure pro_char1
(
in in_char char(10),
out out_char char(10),
in in_varchar varchar(10),
out out_varchar varchar(10)
)
dynamic result sets 1
language SQL

P1: begin
set out_char = in_char;
set out_varchar = in_varchar;
end P1
```

6) 那么在 query 中表的选项中会显示出所有的存储过程，然后在 SQL 语句中填写如下的语句：

```
Call PRO_CHAR1(?{IN_CHAR},?{OUT_CHAR},?{IN_VARCHAR},?{OUT_VARCHAR})
```

7) 然后再点击参数按钮，给每一个参数设定数据类型和方向，每个参数的方向需要和数据库参数的方向一致。

8) 刷新参数，然后进行预览数据



带参数的 SQL SERVER 数据库

对于 SQL SERVER 数据库来讲，进入模式同 DB2，如果在数据库中创建的存储过程的 SQL 语句是：

create procedure testsp

@jiu int

AS

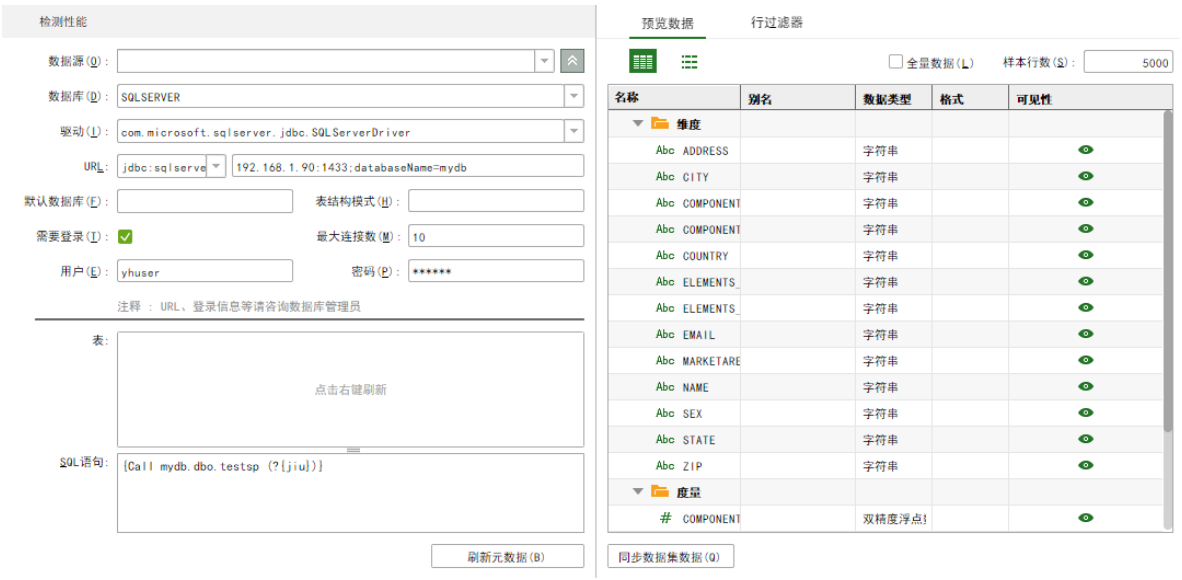
select * from student where consume_number=@jiu

Go

在 SQL SERVER 数据库中写 SQL 语句的时候外面的大括号是必须的，在 oracle 和 DB2 中有没有大括号都不影响运行的，那么在产品中运行该存储过程的 SQL 语句就为：

{Call mydb.dbo.testsp (?{jiu})}

如下图所示



并且在传递参数的同时也需要在参数对话框中对参数 @jiu 进行设置一下，如下图：

刷新元数据就能预览运行出数据了。

不带参数的 SQL SERVER 数据库

对于不带参数的数据库中存储过程的语法如下：

```
create procedure  uui
as
select * from student
go
```

然后在产品的 SQL 语句中写的语句为：

```
{Call mydb.dbo.uui ()}
```

此时没有任何的参数需要配置，刷新元数据就能预览出数据了。

ORACLE 数据库返回数据集（游标）

对于 ORACLE 数据库，进入的模式同 DB2，如果在数据库中创建的存储过程的语句是：

CREATE PROCEDURE SCOTT.SP_OUT_JOBS(ret_cursor1 OUT sys_refcursor)

IS

BEGIN

OPEN ret_cursor1 for select * from HR.JOBS;

END;

然后在产品的 sql 语句处写的调用语句是

Call "CELINA"."SP_OUT_JOBS" (?{out_cursor})，情况如下图：

同时要对参数进行配置，包括参数的类型，方向和默认值等，配置如下图：

刷新出来的数据如下图：

Abc ADDRESS	Abc CITY	Abc COMPONENTS_...	Abc COMPONENTS_DES...
301 Withers Street	Huntington	ML-034	17" Filter Screen
96 Margulies	Drogenbos	PT-099	hp scanjet 5550c s
23rd Street Suite 201	Manchester	MB-041	Gigabyte GA-7DPXDW
44 Parker Road	Rio Rancho	CP-019	AMD XP 2100+ Box w
660 Tempest Drive	Reading	SW-011	MicroSoft Windows
66 Wagner Ave	Koeln	ABB-078	AMD XP 1700+ & Gig
44 Barrymore Blvd	Okayama	YS-006	AMD 1.1Ghz Starter
69 H?ssleholm Blvd	Kristiansand	CS-031	313N-B ATX case 30
290 Nelson Ave	Huntsville	CMI-084	Combo Intel P4 1.8
72nd Street	Calcutta	CP-013	AMD Duron 1.1Ghz
85 Tottori Ave	Grapevine	CS-024	6072-1 case with s
20 Duschel Street	Douala	ML-017	Cooler Master Roun
73 Harrison Drive	Holliston	CP-045	AMD Duron 1.3 Ghz
43 Lewin Blvd	Lodi	MB-035	Gigabyte GA-8IE800
30 Richter Drive	Macclesfield	CS-051	Allied 400watt Pow

第 7 章 : 附录

附录中列出了永洪产品支持的 SQL 函数。其中非聚合函数还可以支持用户自定义的函数。另外 ,script.functions.path 文件中的内容支持热装载，修改内容不需要重新启动。附录如下 :

常用查询语句

基础语句

例子 :

All

SELECT * from test_cloud.clqry

Where

SELECT Market_Size, Sales from test_cloud.clqry where Market_Size ="Major Market"

Group by

SELECT Market, Sum(Sales) as Sum_Sales from test_cloud.clqry group by Market

Order by

SELECT Market, Sales from test_cloud.clqry order by Sales DESC

SubQuery

SELECT Id_O as M_ID FROM Orders.eqry WHERE M_ID = (SELECT domestic_price FROM CustomOilQry.cqry WHERE domestic_price=3

SELECT ID FROM DerbyCoffeeQryScript.scqry where ID > (SELECT ID as S_ID FROM DerbyCoffeeQrySQL.sqry WHERE S_ID = 4245

Alias

SELECT Market as "Market1", Sum(Sales) as Sum_Sales from test_cloud.clqry group by Market SELECT Market as shichang from test_cloud.clqry

过滤条件

Like

SELECT Market from test_cloud.clqry where Market like '%a%'

Sub Query not in

SELECT domestic_price as M_ID ,global_price FROM CustomOilQry.cqry where M_ID NOT IN (SELECT ID as S_ID FROM DerbyCoffeeQrySQL.sqry WHERE S_ID>3

Between

SELECT ID from test_cloud.clqry where ID between 2 and 10

ContainsIn

“=”, "!=", "<>", ">", ">=", "<", "<="

Join

SELECT * from "a.eqry" A inner join "b.eqry" B on A.key=B.key

SELECT * from "a.eqry" A full join "b.eqry" B on A.key=B.key

SELECT * from "a.eqry" A right join "b.eqry" B on A.key=B.key

SELECT * from "a.eqry" A left join "b.eqry" B on A.key=B.key

Union

SELECT ID1 from "a.eqry" A union select ID2 from "b.eqry"

算数计算

例子 :

SELECT State as state_a, Product, sum(ID) + 2 as tp from coffee.sqry group by state_a, Product order by tp

SELECT Sales * 2 + 3 from test_cloud.clqry

日期函数

Year, Quarter, month, week, day, hour, minute, second

SELECT "Date", year("Date"), quarter("Date"), month("Date"), week("Date"), day("Date"), hour("Date"), minute("Date"), second("Date") from test_cloud.clqry

Datepart: quarterpart, monthpart, weekpart, daypart, hourpart, minutepart, secondpart

SELECT "Date", quarterpart("Date"), monthpart("Date"), weekpart("Date"), daypart("Date"), hourpart("Date"), minutepart("Date"), secondpart("Date") from test_cloud.clqry

GIS 函数：incircle, inrect

InCircle 用来返回指定经纬度，半径（单位是米）以内满足条件的点

例子 :

SELECT longitude,latitude from "gis/gis_cloud.clqry" where pos1 incircle(6.0, 5.1, 1000000) order by longitude, latitude asc

InRect 用来返回指定中心点，宽和高（单位是米）以内满足条件的点

例子：

SELECT longitude,latitude from "gis/gis_cloud.clqry" where pos1 inrect(-15, 4.1, 2226388, 2226388) order by longitude, latitude asc

聚合函数

Max

SELECT Max(Sales) as Max_Sales from test_cloud.clqry

SELECT Product, Max(Sales) as Max_Sales from test_cloud.clqry group by Product

Min

SELECT Product, Min(Sales) as Min_Sales from test_cloud.clqry group by Product

Sum

SELECT Sum(Sales) as Sum_Sales from test_cloud.clqry

Count

SELECT Count(Sales) as Count_Sales from test_cloud.clqry

Count Distinct

SELECT Count(Distinct Market) as DistinctCount_Market from test_cloud.clqry

DistinctCount

SELECT Product, DistinctCount(Sales) as DistinctCount_Market from test_cloud.clqry group by Product

Avg

SELECT Product, Avg(Sales) as Avg_Sales from test_cloud.clqry group by Product

Correlation

SELECT Product, Correlation(Sales, Profit) as Correlation_Sales_Profit from test_cloud.clqry group by Product

Covariance

SELECT Product, Covariance(Sales, Profit) as Covariance_Sales_Profit from test_cloud.clqry group by Product

Median

SELECT Product, Median(Sales) as Median_Sales from test_cloud.clqry group by Product

PopulationStandardDeviation

SELECT Product, PopulationStandardDeviation(Sales) as PopulationStandardDeviation_Sales from test_cloud.clqry group by Product

PopulationVariance

SELECT Product, PopulationVariance(Sales) as PopulationVariance_Sales from test_cloud.clqry group by Product

Product

SELECT Product, Product(Sales) as Product_Sales from test_cloud.clqry where ID <= 100 group by Product

PthPercentile

SELECT Product, PthPercentile(Sales, 20) as PthPercentile_Sales from test_cloud.clqry group by Product

Quartile

SELECT Product, Quartile(Sales, 1) as Quartile_Sales from test_cloud.clqry group by Product

Range

SELECT Product, Min(Sales) as Min_Sales, Max(Sales) as Max_Sales, Range(Sales) as Range_Sales from test_cloud.clqry group by Product

StandardDeviation

SELECT Product, StandardDeviation(Sales) as StandardDeviation_Sales from test_cloud.clqry group by Product

StandardError

SELECT Product,StandardError(Sales) as StandardError_Sales from test_cloud.clqry group by Product

SumSQ

SELECT Product,SumSQ(Sales) as SumSQ_Sales from test_cloud.clqry group by Product

SumWT

SELECT Product,SumWT(Sales,Profit) as SumWT_Sales_Profit from test_cloud.clqry group by Product

Variance

SELECT Product, Variance(Sales) as Variance_Sales from test_cloud.clqry group by Product

WeightAVG

SELECT Product, WeightAVG(Sales, Profit) as WeightAvg_Sales_Profit from test_cloud.clqry group by Product

Mode

SELECT Product, Mode_(Sales) as Mode_Sales from test_cloud.clqry group by Product

非聚合函数

1	isNumber(Object val)	判断给定的参数是不是数字
2	isDate(Object val)	判断给定的参数是不是日期
3	position(经度 :int, 纬度 :int)	把经度和纬度转化为一个 long 来存储 , 可以用来做 GIS inrect 和 inCircle 的输入参数
4	formatDate(Object val, String pattern)	把日期按照指定的格式转化为字符串
5	parseDate(Object val, String pattern)	把字符串按照指定的格式转化为日期
6	dateAdd(Object date, String field, int interval)	按照指定的 Field 修改日期，合理的 field 参数是 ” year”，“quarter”，“month”，“weekofyear”，“dayofyear”，” dayofmonth”，“dayofweek”，“hour”，“minute”，“second”
7	dateGap(Object date1, Object date2, String field)	计算日期按照指定的 Field 的间隔，合理的 field 参数是 ” year”，“quarter”，“month”，“weekofyear”，“dayofyear”，” dayofmonth”，“dayofweek”，“hour”，“minute”，“second”
8	datePart(Object date, String field)	返回日期指定的 Field 的值，合理的 field 参数是 ” year”，“quarter”，“month”，“weekofyear”，“dayofyear”，” dayofmonth”，“dayofweek”，“hour”，“minute”，“second”
9	trim (String str)	去掉字符串前后的空格
10	substring(String str，Integer idx, Object end))	返回第一个参数中从第二个参数指定的位置开始到第三个参数指定的长度的子字符串
11	formatNumber (double num, String pattern)	把数字按照指定的格式转化为字符串
12	indexOf (String src, String key, Object fromIdx)	再制定的字符串中查找子串的位置
13	sqr (Object val)	返回平方
14	sqrt (Object val)	返回平方根
15	abs(Object val)	返回绝对值

提示：非聚合函数支持用户自定义函数

Excel 函数

除了以上常见的函数 , 还支持 Excel 中的各种计算函数 , 具体如下 :

1、数学函数 :

abs(double)

acos(double)

acosh(double)

asin(double)

asinh(double)

atan(double)

atan2(double, double)

atanh(double)

ceiling(double, double)

combin(double, double)

cos(double)

cosh(double)

degrees(double)

even(double)

exp(double)

fact(double)

factdouble(double)

floor(double, double)

gcd(Object)

integer(double)

lcm(Object)

ln(double)

log(double, double)

log10(double)

mathProduct(Object)

mathSum(Object)

mathSumsq(Object)

mdeterm(Object)

mod(double, double)

mround(double, double)

multinomial(Object)

odd(double)

pi()

power(double, double)

quotient(double, double)

radians(double)

rand()

randbetween(int, int)

round(double, int)

rounddown(double, int)

roundup(double, int)

seriessum(double, double, double, Object)

sign(double)

sin(double)

sinh(double)

sqrt(double)

sqrtpi(double)

subtotal(int, Object)

sumif(Object, String, Object)

sumproduct(Object)

sumx2my2(Object, Object)

sumx2py2(Object, Object)

sumxmy2(Object, Object)

tan(double)

tanh(double)

trunc(double, int)

2、日期函数 :

date(int, int, int)

datevalue(Object)

day(Object)

dayofyear(Object)

days360(Object, Object, Object)

edate(Object, int)

eomonth(Object, int)

hour(Object)

minute(Object)

month(Object)

monthname(Object)

networkdays(Object, Object, Object)

now()

quarter(Object)

second(Object)

time(int, int, int)

timevalue(Object)

today()

weekday(Object, Object)

weekdayname(Object)

weeknum(Object, Object)

workday(Object, int, Object)

year(Object)

yearfrac(Object, Object, Object)

3、财务函数 :

accrint(Object, Object, Object, double, double, double, double)

accrintm(Object, Object, double, double, double)

amordegrc(double, Object, Object, double, int, double, int)

amorlinc(double, Object, Object, double, int, double, int)

coupdaybs(Object, Object, double, int)

coupdays(Object, Object, double, double)

coupdaysnc(Object, Object, double, double)

coupncd(Object, Object, double, double)

coupnum(Object, Object, double, double)

couppcd(Object, Object, double, double)

cumipmt(double, int, double, double, double, int)

cumprinc(double, int, double, double, double, int)

ddb(double, double, double, double, double)

disc(Object, Object, double, double, double)

duration(Object, Object, double, double, double, double)

effect(double, double)

financialDb(double, double, double, double, double)

fv(double, int, double, double, int)

fvschedule(double, Object)

intrate(Object, Object, double, double, double)

ipmt(double, int, int, double, double, int)

ispmt(double, int, int, double)

mduration(Object, Object, double, double, double, double)

mirr(Object, double, double)

nominal(double, double)

nper(double, double, double, double, int)

npv(double, Object)

pmt(double, int, double, double, int)

ppmt(double, int, int, double, double, int)

price(Object, Object, double, double, double, double, int)

pricedisc(Object, Object, double, double, double)

pricemat(Object, Object, Object, double, double, double)

pv(double, int, double, double, int)

received(Object, Object, double, double, double)

sln(double, double, int)

syd(double, double, double, double)

tbilleq(Object, Object, double)

tbillprice(Object, Object, double)

tbillyield(Object, Object, double)

vdb(double, double, double, double, double, double, boolean)

xnpv(double, Object, Object)

yielddisc(Object, Object, double, double, double)

yieldmat(Object, Object, Object, double, double, double)

4、逻辑函数 :

and(Object, Object, Object, Object)

iif(Object, Object, Object)

not(Object)

or(Object, Object, Object, Object)

5、统计函数 :

avedev(Object)

average(Object)

averagea(Object)

averageif(Object, String, Object)

binomdist(double, double, double, boolean)

correl(Object, Object)

counta(Object)

countblank(Object)

countdistinct(Object)

countif(Object, String)

countn(Object)

covariance(Object, Object)

devsq(Object)

expondist(double, double, boolean)

fisher(double)

fisherinv(double)

forecast(double, Object, Object)

frequency(Object, Object)

geomean(Object)

harmean(Object)

hypgeomdist(double, double, double, double, boolean)

intercept(Object, Object)

kurt(Object)

large(Object, int)

maxa(Object)

mina(Object)

negbinomdist(double, double, double)

pearson(Object, Object)

percentile(Object, double)

percentrank(Object, double, double)

permut(double, double)

poisson(double, double, boolean)

prob(Object, Object, double, double)

rank(double, Object, double)

rsq(Object, Object)

skew(Object)

slope(Object, Object)

small(Object, int)

standardize(double, double, double)

statisticCount(Object)

statisticMax(Object)

statisticMedian(Object)

statisticMin(Object)

statisticMode(Object)

statisticQuartile(Object, double)

stdev(Object)

stdeva(Object)

stdevp(Object)

stdevpa(Object)

steyx(Object, Object)

trimmean(Object, double)

vara(Object)

varn(Object)

varp(Object)

varpa(Object)

weibull(double, double, double, boolean)

weightedavg(Object, Object)

6、文本函数 :

character(int)

code(String)

concatenate(Object)

currency(double, int)

dollar(double, int)

exact(String, String)

find(String, String, int)

fixed(double, int, boolean)

left(String, int)

len(String)

lower(String)

mid(String, int, int)

proper(String)

replace(String, int, int, String)

rept(String, int)

right(String, int)

search(String, String, int)