阿里云
MaxCompute

使用教程

文档版本：20190716

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的”现状“、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

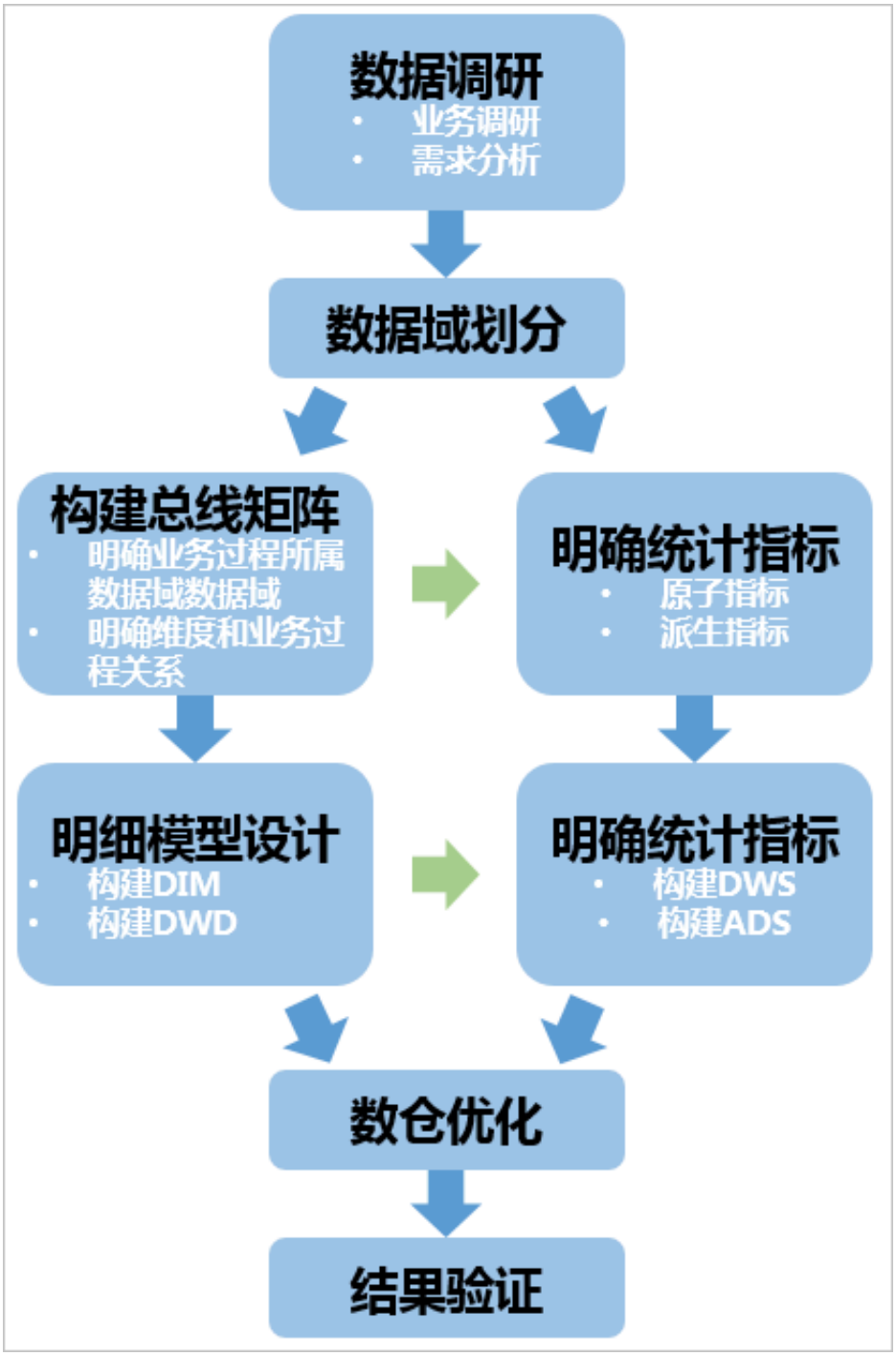
格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **禁止:** 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **警告:** 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 **说明:** 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 确定。
`courier` 字体	命令。	执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。
##	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[]或者[a|b]	表示可选项，至多选择一个。	`ipconfig` *[-all|-t]*
{}或者{a|b}	表示必选项，至多选择一个。	`swich` *{stand | slave}*

目录

1 构建与优化数据仓库

1.1 数仓构建流程

下图为MaxCompute数据仓库构建的整体流程。

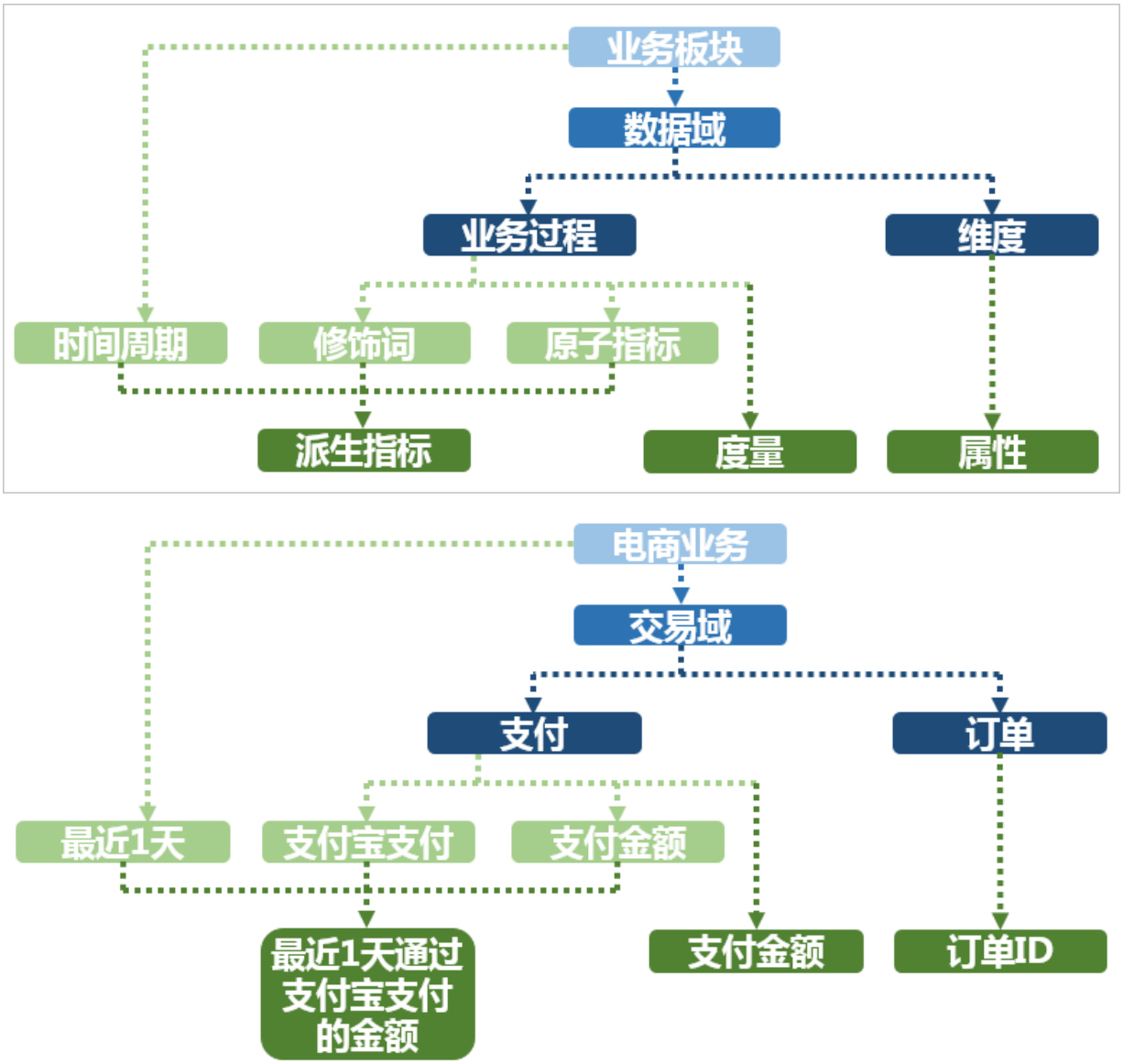


基本概念

在正式学习本教程之前，您需要首先理解以下基本概念：

- 业务板块：比数据域更高维度的业务划分方法，适用于特别庞大的业务系统。
- 维度：维度建模由Ralph Kimball提出。维度模型主张从分析决策的需求出发构建模型，为分析需求服务。维度是度量的环境，是我们观察业务的角度，用来反映业务的一类属性 。属性的集合构成维度 ，也可以称为实体对象。例如， 在分析交易过程时，可以通过买家、卖家、商品和时间等维度描述交易发生的环境。
- 属性（维度属性）：维度所包含的表示维度的列称为维度属性。维度属性是查询约束条件、分组和报表标签生成的基本来源，是数据易用性的关键。
- 度量：在维度建模中，将度量称为事实， 将环境描述为维度，维度是用于分析事实所需要的多样环境。度量通常为数值型数据，作为事实逻辑表的事实。
- 指标：指标分为原子指标和派生指标。原子指标是基于某一业务事件行为下的度量，是业务定义中不可再拆分的指标，是具有明确业务含义的名词 ，体现明确的业务统计口径和计算逻辑，例如如支付金额。
  - 原子指标=业务过程+度量
  - 派生指标=时间周期+修饰词+原子指标，派生指标可以理解为对原子指标业务统计范围的圈定。
- 业务限定：统计的业务范围，筛选出符合业务规则的记录（类似于SQL中`where`后的条件，不包括时间区间）。
- 统计周期：统计的时间范围，例如最近一天，最近30天等（类似于SQL中`where`后的时间条件）。
- 统计粒度：统计分析的对象或视角，定义数据需要汇总的程度，可理解为聚合运算时的分组条件（类似于SQL中的`group by`的对象）。粒度是维度的一个组合，指明您的统计范围。例如某个指标是某个卖家在某个省份的成交额，则粒度就是卖家、地区这两个维度的组合。如果您需要统计全表的数据，则粒度为全表。在指定粒度时，您需要充分考虑到业务和维度的关系。统计粒度常用语作为派生指标的修饰词而存在。

基本概念之间的关系和举例如下图所示。



1.2 业务调研

1.2.1 确定需求

在进行数据仓库构建之前，首先需要确定数仓构建的目标与需求，进行全面的业务调研。您需要了解真实的业务需求是什么，以及确定整个业务系统能解决什么问题。

业务调研

充分的业务调研和需求分析是数据仓库建设的基石，直接决定数据仓库能否建设成功。在数仓建设项目启动前，您需要请相关的业务人员介绍具体的业务，以便明确各个团队的分析员、运营人员的需求，沉淀出相关文档。

您可以通过调查表、访谈等形式详细了解以下信息：

1. 用户的组织架构和分工界面。例如，用户可能分为数据分析、运营、维护部门，各个部门对数仓的需求不同，您需要对不同部门分别进行调研。
2. 用户的整体业务架构，各个业务模块之间的联系与信息流动的流程。梳理出整体的业务数据框架。
3. 各个已有的业务系统的主要功能及获取的数据。

本教程中以A公司的电商业务为例，梳理出业务数据框架如下图所示。A公司的电商业务板块分为招商、供应链、营销、服务四个板块，每个板块的需求和数据应用都不同。在您构建数仓之前，首先需要明确构建数仓服务的业务的板块和需要具体满足的业务需求。

A公司电商	招商	供应链	营销	服务
商业目标/业务需求	寻找优质商家并帮助快速入驻	优化进、销、存链路，降低成本	商家成长、行业增长、精准营销	提升用户体验和留存
数据需求	市场评估、商家成交分析、品牌成交分析	仓库选址、货品规划、货单跟踪	用户运营、营销分析、成交驱动	客户体验、服务质量、完美订单
核心数据	品牌分析、行业趋势、商家流量、商家成交	供应商分层、库存周转、财务结算、库存管理、物流时效	行业用户、行业流量、竞品监控、订单成交	退款纠纷、用户评价、投诉率
数据应用	销售预测、商家分层、生意参谋	物流时效、货品汰换、智能补货	用户画像、成交预测、品类分析、人群投放	假货感知、服务跟踪

此外，您还需要进一步了解各业务板块中已有的各数据功能模块。功能模块通常和业务板块紧耦合，对应一个或多个表，可以作为构建数仓的数据源。下表展现的是一个营销业务板块的数据功能模块。

功能模块	A公司电商营销管理
商品管理	Y
用户管理	Y
购买流程	Y
交易订单	Y
用户反馈	Y

说明:

Y代表包含该功能模块，N代表不包含。

本教程中，假设用户是电商营销部门的营销数据分析师。数据需求为最近一天某个类目（例如：厨具）商品在各省的销售总额、该类目Top10销售额商品名称、各省用户购买力分布（人均消费额）等，用于营销分析。最终的业务需求是通过营销分析完成该类目的精准营销，提升销售总额。通过业务调研，我们将着力分析营销业务板块的交易订单功能模块。

需求分析

在未考虑数据分析师、业务运营人员的数据需求的情况下，单纯根据业务调研建设的数据仓库可用性差。完成业务调研后，您需要进一步收集数据使用者的需求，进而对需求进行深度的思考和分析。

需求分析的途径有两种：

- 根据与分析师、业务运营人员的沟通获知需求。
- 对报表系统中现有的报表进行研究分析。

在进行需求分析阶段，您需要沉淀出业务分析或报表中的指标，以及指标的定义和粒度。粒度可以作为维度的输入。建议您思考下列问题，对后续的数据建模将有巨大的帮助：

- 业务数据是根据什么（维度、粒度）汇总的，衡量标准是什么？例如，成交量是维度，订单数是度量。
- 明细数据层和汇总数据层应该如何设计？公共维度层该如何设计？是否有公共的指标？
- 数据是否需要冗余、沉淀到汇总数据层中？

举例： 数据分析师需要了解A公司电商业务中厨具类目的成交金额。当获知这个需求后，您需要分析：根据什么（维度）汇总、汇总什么（度量）以及汇总的范围多大（粒度）。例如，类目是维度，金额是度量，范围是全表。此外，还需要思考明细数据和汇总数据应该如何设计、是否是公共层的报表、数据是否需要沉淀到汇总表中等因素。

需求调研的分析产出通常是记录原子与派生指标的文档。

1.2.2 分析业务过程

业务过程可以概括为一个个不可拆分的行为事件。用户的业务系统中，通过埋点或日常积累，通常已经获取了充足的业务数据。为理清数据之间的逻辑关系和流向，首先需要理解用户的业务过程，了解过程中涉及到的数据系统。

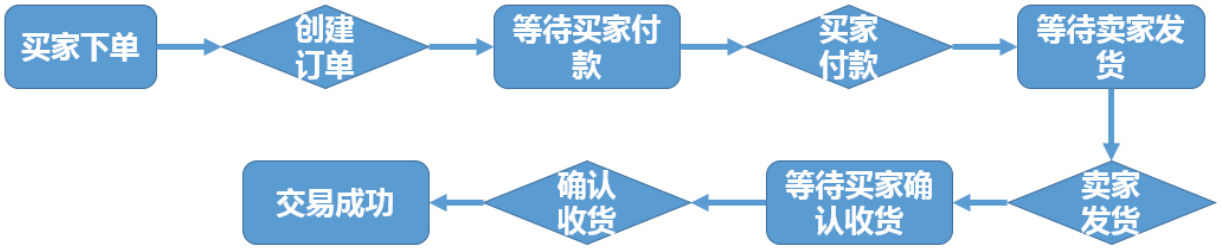
您可以采用过程分析法，将整个业务过程涉及的每个环节一一列清楚，包括技术、数据、系统环境等。在分析企业的工作职责范围（部门）后，您也可以借助工具通过逆向工程抽取业务系统的真实模型。您可以参考业务规划设计文档以及业务运行（开发、设计、变更等）相关文档，全面分析数据仓库涉及的源系统及业务管理系统：

- 每个业务会生成哪些数据，存在于什么数据库中。
- 对业务过程进行分解，了解过程中的每一个环节会产生哪些数据，数据的内容是什么。
- 数据在什么情况下会更新，更新的逻辑是什么。

业务过程可以是单个业务事件，例如交易的支付、退款等；也可以是某个事件的状态，例如当前的账户余额等；还可以是一系列相关业务事件组成的业务流程。具体取决于您分析的是某些事件过去发生情况、当前状态，或是事件流转效率。

选择粒度：在业务过程事件分析中，您需要预判所有分析需要细分的程度和范围，从而决定选择的粒度。 识别维表、选择好粒度之后，您需要基于此粒度设计维表，包括维度属性等，用于分析时进行分组和筛选。最后，您需要确定衡量的指标。

本教程中，经过业务过程调研，我们了解到用户电商营销业务的交易订单功能模块的业务流程如下。



这是一个非常典型的电商交易业务流程图。在该业务流程图中，有创建订单、买家付款、卖家发货、确认收货四个核心业务步骤。由于确认收货代表交易成功，我们重点分析确认收货或交易成功步骤即可。

在明确用户的业务过程之后，您可以根据需要进行分析决策的业务划分数据域。

1.2.3 划分数据域

数据仓库是面向主题（数据综合、归类并进行分析利用的抽象）的应用。数据仓库模型设计除横向的分层外，通常也需要根据业务情况进行纵向划分数据域。数据域是联系较为紧密的数据主题的集合，是业务对象高度概括的概念层次归类，目的是便于数据的管理和应用。

划分数据域

通常，您需要阅读各源系统的设计文档、数据字典和数据模型设计文档，研究逆向导出的物理数据模型。进而，可以进行跨源的主题域合并，跨源梳理出整个企业的数据域。

数据域是指面向业务分析，将业务过程或者维度进行抽象的集合。为保障整个体系的生命力，数据域需要抽象提炼，并长期维护更新。在划分数据域时，既能涵盖当前所有的业务需求，又能让新业务在进入时可以被包含进已有的数据域或扩展新的数据域。数据域的划分工作可以在业务调研之后进行，需要分析各个业务模块中有哪些业务活动。

数据域可以按照用户企业的部门划分，也可以按照业务过程或者业务板块中的功能模块进行划分。例如A公司电商营销业务板块可以划分为如下数据域，数据域中每一部分都是实际业务过程经过归纳抽象之后得出的。

数据域	业务过程
会员店铺域	注册、登录、装修、开店、关店
商品域	发布、上架、下架、重发
日志域	曝光、浏览、点击
交易域	下单、支付、发货、确认收货
服务域	商品收藏、拜访、培训、优惠券领用
采购域	商品采购、供应链管理

1.2.4 定义维度与构建总线矩阵

明确每个数据域下有哪些业务过程后，您需要开始定义维度，并基于维度构建总线矩阵。

定义维度

在划分数据域、构建总线矩阵时，需要结合对业务过程的分析定义维度。以本教程中A电商公司的营销业务板块为例，在交易数据域中，我们重点考察确认收货（交易成功）的业务过程。

在确认收货的业务过程中，主要有商品和收货地点（本教程中，假设收货和购买是同一个地点）两个维度所依赖的业务角度。从商品角度我们可以定义出以下维度：

- 商品ID（主键）
- 商品名称
- 商品交易价格
- 商品新旧程度： 1 全新 2 闲置 3 二手
- 商品类目ID
- 商品类目名称
- 品类ID
- 品类名称
- 买家ID
- 商品状态： 0 正常 1户删除 2下架 3
- 商品所在城市
- 商品所在省份

从地域角度，我们可以定义出以下维度：

- 城市code
- 城市名称
- 省份code
- 省份名称

作为维度建模的核心，在企业级数据仓库中必须保证维度的唯一性。以A公司的商品维度为例，有且只允许有一种维度定义。例如，省份code这个维度，对于任何业务过程所传达的信息都是一致的。

构建总线矩阵

明确每个数据域下有哪些业务过程后，即可构建总线矩阵。您需要明确业务过程与哪些维度相关，并定义每个数据域下的业务过程和维度。如下所示是A公司电商板块交易功能的总线矩阵，我们定义了购买省份、购买城市、类目名称、类目ID、品牌名称、品牌ID、商品名称、商品ID、成交金额等维度。

数据域/过程		一致性维度								
		购买省份	购买城市	类目ID	类目名称	品牌ID	品牌名称	商品ID	商品名称	成交金额
交易	下单	Y	Y	Y	Y	Y	Y	Y	Y	N
	支付	Y	Y	Y	Y	Y	Y	Y	Y	N
	发货	Y	Y	Y	Y	Y	Y	Y	Y	N
	确认收货	Y	Y	Y	Y	Y	Y	Y	Y	Y

说明:

Y代表包含该维度，N代表不包含。

1.2.5 明确统计指标

需求调研输出的文档中，含有原子指标与派生指标，此时我们需要在设计汇总层表模型前完成指标的设计。

指标定义注意事项

原子指标是明确的统计口径、计算逻辑：`原子指标=业务过程+度量`。派生指标即常见的统计指标：`派生指标=时间周期+修饰词+原子指标`。原子指标的创建需要在业务过程定义后方才可创建。派生指标的创建一般需要在了解具体报表需求之后展开，在新建派生指标前必须新建好原子指标。注意事项如下：

- 原子指标、修饰类型及修饰词，直接归属在业务过程下，其中修饰词继承修饰类型的数据域。
- 派生指标可以选择多个修饰词，由具体的派生指标语义决定。例如，支付金额为原子指标，则客单价（支付金额除以买家数）为派生指标。
- 派生指标唯一归属一个原子指标，继承原子指标的数据域，与修饰词的数据域无关。

根据业务需求确定指标

本教程中，用户是电商营销部门的营销数据分析师。数据需求为最近一天厨具类目的商品在各省的销售总额、该类目Top10销售额商品名称、各省用户购买力分布（人均消费额）等，用于营销分析。

根据之前的分析，我们确认业务过程为：确认收货（交易成功），而度量为商品的销售金额。因此根据业务需求，我们可以定义出原子指标：商品成功交易金额。

派生指标为：

- 最近一天全省厨具类目各商品销售总额
- 最近一天全省厨具类目人均消费额（消费总额除以人数）

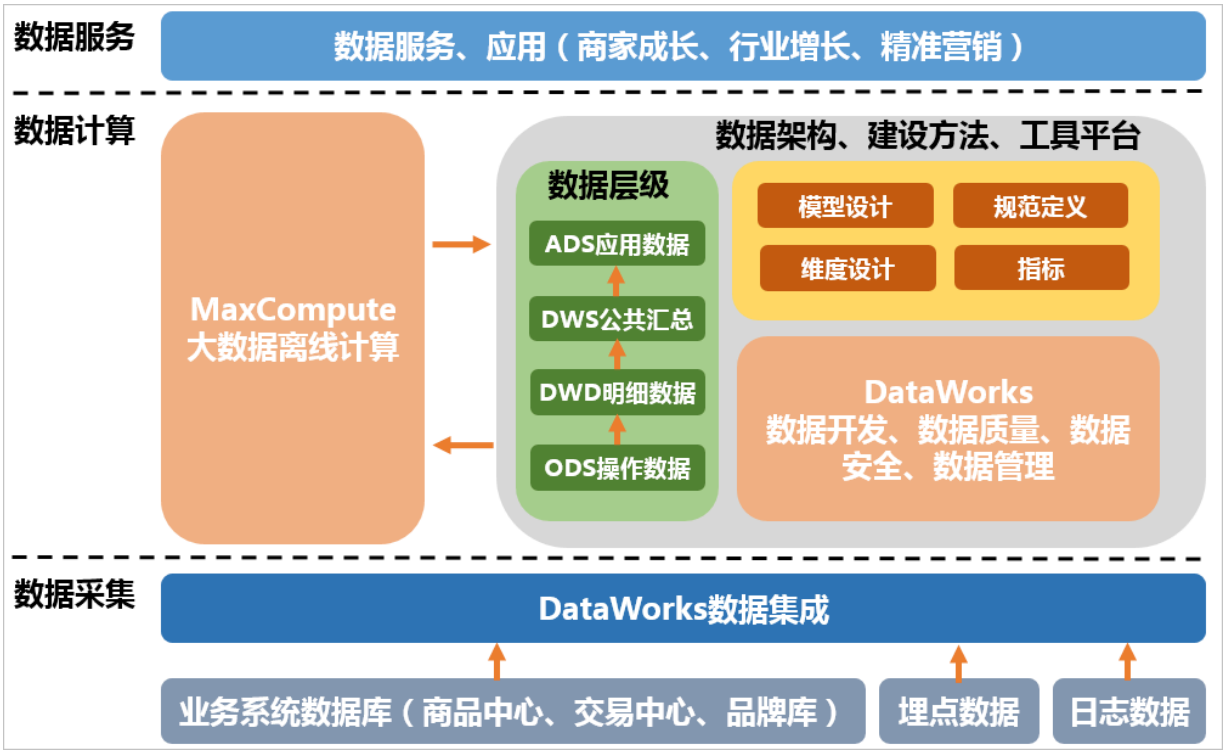
最近一天全省厨具类目各商品销售总额进行降序排序后取前10名的名称，即可得到该类目Top10销售额商品名称。

1.3 架构与模型设计

1.3.1 技术架构选型

在数据模型设计之前，您需要首先完成技术架构的选型。本教程中使用阿里云大数据产品MaxCompute配合DataWorks，完成整体的数据建模和研发流程。

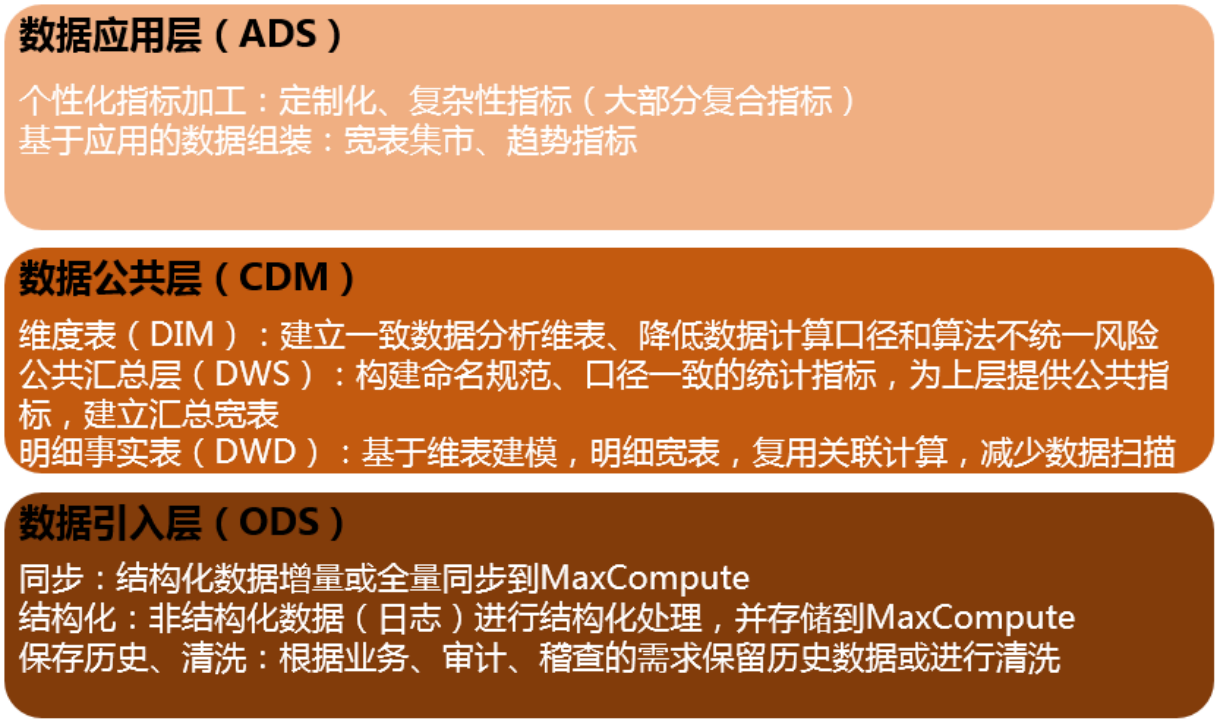
完整的技术架构图如下图所示。其中，DataWorks的数据集成负责完成数据的采集和基本的ETL。MaxCompute作为整个大数据开发过程中的离线计算引擎。DataWorks则囊括数据开发、数据质量、数据安全、数据管理等在内的一系列功能。



1.3.2 数仓分层

在阿里巴巴的数据体系中，我们建议将数据仓库分为三层，自下而上为：数据引入层（ODS，Operation Data Store）、数据公共层（CDM，Common Data Model）和数据应用层（ADS，Application Data Service）。

数据仓库的分层和各层级用途如下图所示。



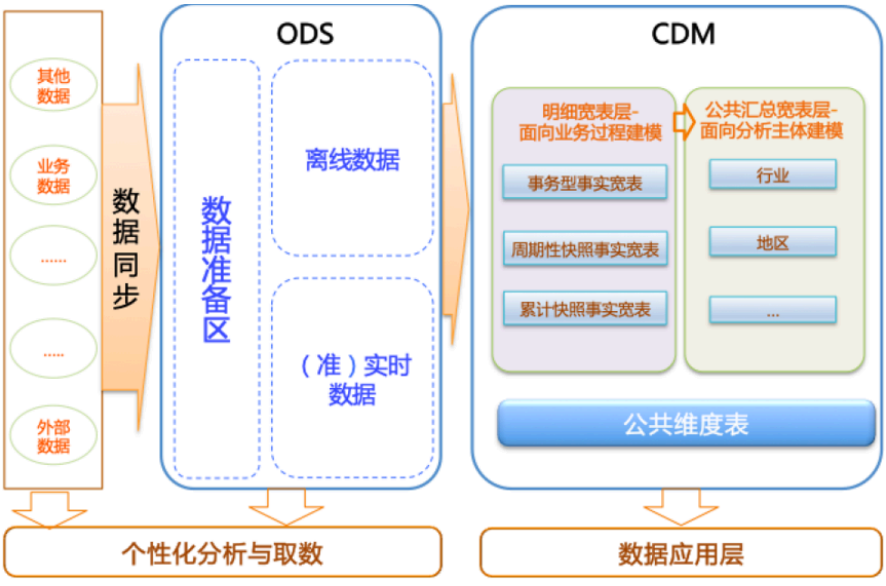
- 数据引入层（ODS，Operation Data Store）：将原始数据几乎无处理的存放在数据仓库系统，结构上与源系统基本保持一致，是数据仓库的数据准备区。主要完成基础数据引入到MaxCompute的职责，同时可以基础数据记录的历史变化的。
- 数据公共层（CDM，Common Data Model，又称通用数据模型层），包含DIM维度表、DWD和DWS，由ODS层数据加工而成。主要完成数据加工与整合，建立一致性的维度，构建可复用的面向分析和统计的明细事实表，以及汇总公共粒度的指标。
  - 公共维度层（DIM）：基于维度建模理念思想，建立整个企业的一致性维度。降低数据计算口径和算法不统一风险。

    公共维度层的表通常也被称为逻辑维度表，维度和维度逻辑表通常一一对应。
  -
  - 公共汇总粒度事实层（DWS）：以分析的主题对象作为建模驱动，基于上层的应用和产品的指标需求，构建公共粒度的汇总指标事实表，以宽表化手段物理化模型。构建命名规范、口径一致的统计指标，为上层提供公共指标，建立汇总宽表、明细事实表。

    公共汇总粒度事实层的表通常也被称为汇总逻辑表，用于存放派生指标数据。
  - 明细粒度事实层（DWD）：以业务过程作为建模驱动，基于每个具体的业务过程特点，构建最细粒度的明细层事实表。可以结合企业的数据使用特点，将明细事实表的某些重要维度属性字段做适当冗余，也即宽表化处理。

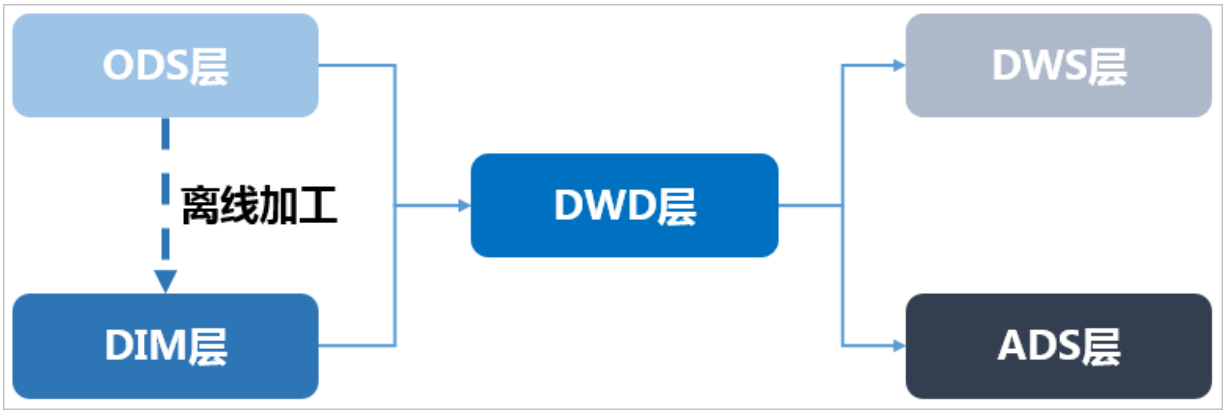
    明细粒度事实层的表通常也被称为逻辑事实表。
- 数据应用层（ADS，Application Data Service）：存放数据产品个性化的统计指标数据。根据CDM与ODS层加工生成。

该数据分类架构在ODS层分为三部分：数据准备区、离线数据和准实时数据区。整体数据分类架构如下图所示。



在本教程中，从交易数据系统的数据经过DataWorks数据集成，同步到数据仓库的ODS层。经过数据开发形成事实宽表后，再以商品、地域等为维度进行公共汇总。

整体的数据流向如下图所示。其中，ODS层到DIM层的ETL（萃取（Extract）、转置（Transform）及加载（Load））处理是在MaxCompute中进行的，处理完成后会同步到所有存储系统。ODS层和DWD层会放在数据中间件中，供下游订阅使用。而DWS层和ADS层的数据通常会落地到在线存储系统中，下游通过接口调用的形式使用。



1.3.3 层次调用规范

在完成数据仓库的分层后，您需要对各层次的数据之间的调用关系作出约定。

层次调用规范

ADS应用层优先调用数据仓库公共层数据。如果已经存在CDM层数据，不允许ADS应用层跨过CDM中间层从ODS层重复加工数据。CDM中间层应该积极了解应用层数据的建设需求，将公用的数据沉淀到公共层，为其他数据层次提供数据服务。同时，ADS应用层也需积极配合CDM中间层进行持续的数据公共建设的改造。避免出现过度的ODS层引用、不合理的数据复制和子集合冗余。总体遵循的层次调用原则如下：

- ODS层数据不能直接被应用层任务引用。如果中间层没有沉淀的ODS层数据，则通过CDM层的视图访问。CDM层视图必须使用调度程序进行封装，保持视图的可维护性与可管理性。
- CDM层任务的深度不宜过大（建议不超过10层）。
- 一个计算刷新任务只允许一个输出表，特殊情况除外。
- 如果多个任务刷新输出一个表（不同任务插入不同的分区），DataWorks上需要建立一个虚拟任务，依赖多个任务的刷新和输出。通常，下游应该依赖此虚拟任务。
- CDM汇总层优先调用CDM明细层，可累加指标计算。CDM汇总层尽量优先调用已经产出的粗粒度汇总层，避免大量汇总层数据直接从海量的明细数据层中计算得出。
- CDM明细层累计快照事实表优先调用CDM事务型事实表，保持数据的一致性产出。
- 有针对性地建设CDM公共汇总层，避免应用层过度引用和依赖CDM层明细数据。

1.3.4 数据模型

1.3.4.1 数据引入层（ODS）

ODS层存放您从业务系统获取的最原始的数据，是其他上层数据的源数据。业务数据系统中的数据通常为非常细节的数据，经过长时间累积，且访问频率很高，是面向应用的数据。

说明:

在构建MaxCompute数据仓库的表之前，您需要首先了解MaxCompute支持的数据类型。

数据引入层表设计

本教程中，在ODS层主要包括的数据有：交易系统订单详情、用户信息详情、商品详情等。这些数据未经处理，是最原始的数据。逻辑上，这些数据都是以二维表的形式存储。虽然严格的说ODS层不属于数仓建模的范畴，但是合理的规划ODS层并做好数据同步也非常重要。本教程中，使用了6张ODS表：

- 记录用于拍卖的商品信息：s_auction。
- 记录用于正常售卖的商品信息：s_sale。
- 记录用户详细信息：s_users_extra。
- 记录新增的商品成交订单信息：s_biz_order_delta。
- 记录新增的物流订单信息：s_logistics_order_delta。
- 记录新增的支付订单信息：s_pay_order_delta。

说明:

- 表或字段命名尽量保持和业务系统保持一致，但是需要通过额外的标识来区分增量和全量表。例如，我们通过_delta来标识该表为增量表。
- 命名时需要特别注意冲突处理，例如不同业务系统的表可能是同一个名称。为区分两个不同的表，您可以将这两个同名表的来源数据库名称作为后缀或前缀。例如，表中某些字段的名称刚好和关键字重名了，可以通过规范定义后缀添加_col1解决。

ODS层设计规范

ODS层表命名、数据同步任务命名、数据产出及生命周期管理及数据质量规范请参见ODS层设计规范。

建表示例

为方便您使用，集中提供建表语句如下，示例数据请参见附录。

```
CREATE TABLE IF NOT EXISTS s_auction
(
```

```
    id                                  STRING COMMENT '商品ID',
    title                               STRING COMMENT '商品名',
    gmt_modified                        STRING COMMENT '商品最后修改日期',
    price                               DOUBLE COMMENT '商品成交价格，单位元',
    starts                              STRING COMMENT '商品上架时间',
    minimum_bid                         DOUBLE COMMENT '拍卖商品起拍价，单位
元',
    duration                            STRING COMMENT '有效期，销售周期，单位
天',
    incrementnum                        DOUBLE COMMENT '拍卖价格的增价幅度',
    city                                STRING COMMENT '商品所在城市',
    prov                                STRING COMMENT '商品所在省份',
    ends                                STRING COMMENT '销售结束时间',
    quantity                            BIGINT COMMENT '数量',
    stuff_status                        BIGINT COMMENT '商品新旧程度 0 全新 1
 闲置 2 二手',
    auction_status                      BIGINT COMMENT '商品状态 0 正常 1用户
删除 2下架 3 从未上架',
    cate_id                              BIGINT COMMENT '商品类目id',
    cate_name                             STRING COMMENT '商品类目名称',
    commodity_id                          BIGINT COMMENT '品类ID',
    commodity_name                         STRING COMMENT '品类名称',
    umid                                   STRING COMMENT '买家umid'
)
COMMENT '商品拍卖ODS'
PARTITIONED BY (ds          STRING COMMENT '格式: YYYYMMDD')
LIFECYCLE 400;

CREATE TABLE IF NOT EXISTS s_sale
(
    id                                  STRING COMMENT '商品ID',
    title                               STRING COMMENT '商品名',
    gmt_modified                        STRING COMMENT '商品最后修改日期',
    starts                              STRING COMMENT '商品上架时间',
    price                               DOUBLE COMMENT '商品价格，单位元',
    city                                STRING COMMENT '商品所在城市',
    prov                                STRING COMMENT '商品所在省份',
    quantity                            BIGINT COMMENT '数量',
    stuff_status                        BIGINT COMMENT '商品新旧程度 0 全新 1
 闲置 2 二手',
    auction_status                       BIGINT COMMENT '商品状态 0 正常 1户删
除 2下架 3 从未上架',
    cate_id                            BIGINT COMMENT '商品类目id',
    cate_name                          STRING COMMENT '商品类目名称',
    commodity_id                       BIGINT COMMENT '品类ID',
    commodity_name                      STRING COMMENT '品类名称',
    umid                                STRING COMMENT '买家umid'
)
COMMENT '商品正常购买ODS'
PARTITIONED BY (ds      STRING COMMENT '格式: YYYYMMDD')
LIFECYCLE 400;


s_users_extra
CREATE TABLE IF NOT EXISTS s_users_extra
(
    id                  STRING COMMENT '用户id',
    logincount          BIGINT COMMENT '登录次数',
    buyer_goodnum       BIGINT COMMENT '作为买家的好评数',
    seller_goodnum      BIGINT COMMENT '作为卖家的好评数',
    level_type          BIGINT COMMENT '1 一级店铺 2 二级店铺 3 三级店铺',
    promoted_num        BIGINT COMMENT '1 A级服务  2 B级服务  3 C级服务',
    gmt_create          STRING COMMENT '创建时间',
    order_id            BIGINT COMMENT '订单ID',
```

```
    buyer_id           BIGINT COMMENT '买家id',
    buyer_nick         STRING COMMENT '买家昵称',
    buyer_star_id      BIGINT COMMENT '买家星级 ID',
    seller_id          BIGINT COMMENT '卖家ID',
    seller_nick        STRING COMMENT '卖家昵称',
    seller_star_id     BIGINT COMMENT '卖家星级id',
    shop_id            BIGINT COMMENT '店铺id',
    shop_name          STRING COMMENT '店铺名称'
)
COMMENT '用户扩展表'
PARTITIONED BY (ds       STRING COMMENT 'yyyymmdd')
LIFECYCLE 400;

CREATE TABLE IF NOT EXISTS s_biz_order_delta
(
    biz_order_id          STRING COMMENT '订单ID',
    pay_order_id          STRING COMMENT '支付订单id',
    logistics_order_id    STRING COMMENT '物流订单id',
    buyer_nick            STRING COMMENT '买家昵称',
    buyer_id              STRING COMMENT '买家id',
    seller_nick           STRING COMMENT '卖家昵称',
    seller_id             STRING COMMENT '卖家id',
    auction_id            STRING COMMENT '商品id',
    auction_title         STRING COMMENT '商品标题 ',
    auction_price         DOUBLE COMMENT '商品价格',
    buy_amount            BIGINT COMMENT '购买数量',
    buy_fee               BIGINT COMMENT '购买金额',
    pay_status            BIGINT COMMENT '支付状态 1 未付款  2 已付款 3 已
退款',
    logistics_id          BIGINT COMMENT '物流订单id',
    mord_cod_status       BIGINT COMMENT '物流状态 0 初始状态 1 接单成功 2
 接单超时3 揽收成功 4揽收失败 5 签收成功 6 签收失败 7 用户取消物流订单',
    status                BIGINT COMMENT '状态 0 订单正常 1 订单不可见',
    sub_biz_type          BIGINT COMMENT '业务类型 1 拍卖 2 购买',
    end_time              STRING COMMENT '交易结束时间',
    shop_id               BIGINT COMMENT '店铺id'
)
COMMENT '交易成功订单日增量表'
PARTITIONED BY (ds       STRING COMMENT 'yyyymmdd')
LIFECYCLE 7200;


CREATE TABLE IF NOT EXISTS s_logistics_order_delta
(
    logistics_order_id STRING COMMENT '物流订单id ',
    post_fee           DOUBLE COMMENT '物流费用',
    address            STRING COMMENT '收货地址',
    full_name          STRING COMMENT '收货人全名',
    mobile_phone       STRING COMMENT '移动电话',
    prov               STRING COMMENT '省份',
    prov_code          STRING COMMENT '省份ID',
    city               STRING COMMENT '市',
    city_code          STRING COMMENT '城市ID',
    logistics_status   BIGINT COMMENT '物流状态
1 - 未发货
2 - 已发货
3 - 已收货
4 - 已退货
5 - 配货中',
    consign_time       STRING COMMENT '发货时间',
    gmt_create         STRING COMMENT '订单创建时间',
    shipping           BIGINT COMMENT '发货方式
1, 平邮
2, 快递
```

```
3, EMS',
    seller_id          STRING COMMENT '卖家ID',
    buyer_id           STRING COMMENT '买家ID'
)
COMMENT '交易物流订单日增量表'
PARTITIONED BY (ds                 STRING COMMENT '日期')
LIFECYCLE 7200;


CREATE TABLE IF NOT EXISTS s_pay_order_delta
(
    pay_order_id     STRING COMMENT '支付订单id',
    total_fee        DOUBLE COMMENT '应支付总金额 （数量*单价）',
    seller_id STRING COMMENT '卖家id',
    buyer_id  STRING COMMENT '买家id',
    pay_status       BIGINT COMMENT '支付状态
1等待买家付款,
2等待卖家发货,
3交易成功',
    pay_time         STRING COMMENT '付款时间',
    gmt_create       STRING COMMENT '订单创建时间',
    refund_fee       DOUBLE COMMENT '退款金额(包含运费)',
    confirm_paid_fee DOUBLE COMMENT '已经确认收货的金额'
)
COMMENT '交易支付订单增量表'
PARTITIONED BY (ds        STRING COMMENT '日期')
LIFECYCLE 7200;
```

数据引入层存储

为了满足历史数据分析需求，您可以在ODS层表中添加时间维度作为分区字段。实际应用中，您可以选择采用增量、全量存储或拉链存储的方式。

- 增量存储

  以天为单位的增量存储，以业务日期作为分区，每个分区存放日增量的业务数据。 举例如下：

  - 1月1日，用户A访问了A公司电商店铺B，A公司电商日志产生一条记录t1。1月2日，用户A又访问了A公司电商店铺C，A公司电商日志产生一条记录t2。采用增量存储方式，t1将存储在1月1日这个分区中，t2将存储在1月2日这个分区中。
  - 1月1日，用户A在A公司电商网购买了B商品，交易日志将生成一条记录t1。1月2日，用户A又将B商品退货了，交易日志将更新t1记录。采用增量存储方式，初始购买的t1记录将存储在1月1日这个分区中，更新后的t1将存储在1月2日这个分区中。

  说明：

  交易、日志等事务性较强的ODS表适合增量存储方式。这类表数据量较大，采用全量存储的方式存储成本压力大。此外，这类表的下游应用对于历史全量数据访问的需求较小（此类需求可通过数据仓库后续汇总后得到）。例如，日志类ODS表没有数据更新的业务过程，因此所有增量分区UNION在一起就是一份全量数据。

- 全量存储

  以天为单位的全量存储，以业务日期作为分区，每个分区存放截止到业务日期为止的全量业务数据。 举例如下：

  - 1月1日，卖家A在A公司电商网发布了B、C两个商品，前端商品表将生成两条记录t1、t2。1月2日，卖家A将B商品下架了，同时又发布了商品D，前端商品表将更新记录t1，同时新生成记录t3。采用全量存储方式， 在1月1日这个分区中存储t1和t2两条记录，在1月2日这个分区中存储更新后的t1以及t2、t3记录。

   说明：
  对于小数据量的缓慢变化维度数据，例如商品类目，可直接使用全量存储。

- 拉链存储

  拉链存储通过新增两个时间戳字段（start_dt和end_dt），将所有以天为粒度的变更数据都记录下来，通常分区字段也是这两个时间戳字段。

  拉链存储举例如下。

商品	start_dt	end_dt	卖家	状态
B	20160101	20160102	A	上架
C	20160101	30001231	A	上架
B	20160102	30001231	A	下架

  这样，下游应用可以通过限制时间戳字段来获取历史数据。例如，用户访问1月1日数据，只需限制 stat_dt<=20160101 and end_dt>20160101。

缓慢变化维度

MaxCompute#推荐使用代理键，推荐使用自然键作为维度主键，主要原因有两点：

1. MaxCompute是分布式计算引擎，生成全局唯一的代理键工作量非常大。当遇到大数据量情况下，这项工作就会更加复杂，且没有必要。
2. 使用代理键会增加ETL的复杂性，从而增加ETL任务的开发和维护成本。

在#使用代理键的情况下，缓慢变化维度应该通过两种方式处理：

1. 快照方式。

   数据的计算周期通常为每天一次。基于该周期，处理维度变化的方式为每天一份全量快照。

   例如商品维度，每天保#一份全量商品快照数据。任意一天的事实表均可以取到当天的商品信息，也可以取到最新的商品信息，通过限定日期，采用自然键进行关联即可。该方式的优势主要有以下两点:

   - 处理缓慢变化维度的方式简单有效，开发和维护成本低。
   - 使用方便，易于理解。数据使用方只需要限定日期即可取到当天的快照数据。任意一天的事实快照与任意一天的维度快照通过维度的自然键进行关联即可。

   该方法的弊端主要是存储字样的极大浪费。例如某维度每天的变化量占总体数据量比例很低，极端情况下，每天无变化，这种情况下存储浪费严重。该方法主要实现了通过牺牲存储获取ETL效率的优化和逻辑上的简化。请避免过度使用该方法，且必须要有对应的数据生命周期制度，清除无用的历史数据。

2. 拉链存储（极限存储）

   前文已经为您介绍了拉链存储。该方法不利于数据使用者对数仓的理解，同时因为限定生效日期，产生大量分区，考虑到MaxCompute对分区数量有上限，不利于长远的数仓维护。

   极限存储底层的数据进行拉链存储，但上层通过视图操作或Hook，借助分析语句的语法树，将对极限存储之前的表查询转化成对极限存储表的查询。对于下游用户而言，极限存储表和全量存储方式是一样的。此外，为解决历史拉链分区大量产生的问题，极限存储中把日分区缩减为月份区，即以月为周期完成拉链存储。

   极限存储虽然可以压缩大量的存储空间，但使用麻烦。主要难点在于全量表的维护和过滤变更频繁的维度属性。

综上，通常情况下推荐您使用快照方式处理缓慢变化维。在数据量巨大的情况下，建议您使用拉链存储（极限存储）。

数据同步加载与处理

ODS的数据需要由各数据源系统同步到MaxCompute，才能用于进一步的数据开发。本教程建议您使用DataWorks数据集成功能完成数据同步，详情请参见数据集成概述。在使用数据集成的过程中，建议您遵循以下规范:

- 一个系统的源表只允许同步一次到MaxCompute，保持表结构的一致性。
- 数据集成仅用于离线全量数据同步，实时增量数据同步需要您使用数据传输服务DTS实现，详情请参见数据传输服务DTS。
- 数据集成全量同步的数据直接进入全量表的当日分区。

- ODS层的表建议以统计日期及时间分区表的方式存储，便于管理数据的存储成本和策略控制。
- 数据集成可以自适应处理源系统字段的变更：
  - 如果源系统字段在MaxCompute上目标表不存在，可以由数据集成自动添加不存在的表字段。
  - 如果目标表的字段在源系统不存在，数据集成填充NULL。

1.3.4.2 公共维度汇总层（DIM）

公共维度层（DIM）基于维度建模理念，建立整个企业的一致性维度。

公共维度层主要由维度表（维表）构成。维度是逻辑概念，是衡量和观察业务的角度。维表是根据维度及其属性物理化在大数据平台上构建的表，采用宽表设计的原则。因此，构建公共维度层（DIM）首先需要定义维度。

定义维度

在划分数据域、构建总线矩阵时，需要结合对业务过程的分析定义维度。以本教程中A电商公司的营销业务板块为例，在交易数据域中，我们重点考察确认收货（交易成功）的业务过程。

在确认收货的业务过程中，主要有商品和收货地点（本教程中，假设收货和购买是同一个地点）两个维度所依赖的业务角度。从商品角度可以定义出以下维度：

- 商品ID
- 商品名称
- 商品价格
- 商品新旧程度： 1 全新 2 闲置 3 二手
- 商品类目ID
- 商品类目名称
- 品类ID
- 品类名称
- 买家ID
- 商品状态： 0 正常 1户删除 2下架
- 商品所在城市
- 商品所在省份

从地域角度，可以定义出以下维度：

- 买家ID
- 城市code
- 城市名称

- 省份code
- 省份名称

作为维度建模的核心，在企业级数据仓库中必须保证维度的唯一性。以A公司的商品维度为例，有且只允许有一种维度定义。例如，省份code这个维度，对于任何业务过程所传达的信息都是一致的。

设计维表

完成维度定义后，您就可以对维度进行补充，进而生成维表了。维表的设计需要注意：

- 建议维表单表信息不超过1000万条。
- 维表与其他表进行Join时，建议您使用Map Join。
- 避免过于频繁的更新维表的数据。

在设计维表时，您需要从下列方面进行考虑：

- 维表中数据的稳定性。例如A公司电商会员通常不会出现消亡，但会员数据可能在任何时候更新，此时要考虑创建单个分区存储全量数据。如果存在永远不会更新的记录，您可能需要分别创建历史表与日常表。日常表用于存放当前有效的记录，保持表的数据量不会膨胀；历史表根据消亡时间插入对应分区，使用单个分区存放分区对应时间的消亡记录。
- 是否需要垂直拆分。如果一个维表存在大量属性不被使用，或由于承载过多属性字段导致产出变慢，则需考虑对字段进行拆分，创建多个维表。
- 是否需要水平拆分。如果记录之间有明显的界限，可以考虑拆成多个表或设计成多级分区。
- 核心的维表产出时间通常有严格的要求。

设计维表的主要步骤如下：

1. 完成维度的初步定义，并保证维度的一致性。
2. 确定主维表（中心事实表，本教程中采用星型模型）。此处的主维表通常是数据引入层（ODS）表，直接与业务系统同步。例如，`s_auction`是与前台商品中心系统同步的商品表，此表即是主维表。
3. 确定相关维表。数据仓库是业务源系统的数据整合，不同业务系统或者同一业务系统中的表之间存在关联性。根据对业务的梳理，确定哪些表和主维表存在关联关系，并选择其中的某些表用于生成维度属性。以商品维度为例，根据对业务逻辑的梳理，可以得到商品与类目、卖家、店铺等维度存在关联关系。

4. 确定维度属性，主要包括两个阶段。第一个阶段是从主维表中选择维度属性或生成新的维度属性；第二个阶段是从相关维表中选择维度属性或生成新的维度属性。以商品维度为例，从主维表（s_auction）和类目 、卖家、店铺等相关维表中选择维度属性或生成新的维度属性。
   - 尽可能生成丰富的维度属性。
   - 尽可能多地给出富有意义的文字性描述。
   - 区分数值型属性和事实。
   - 尽量沉淀出通用的维度属性。

公共维度汇总层（DIM）维表规范

公共维度汇总层（DIM）维表命名规范：dim_{业务板块名称/pub}_{维度定义}[_{自定义命名标签}]，所谓pub是与具体业务板块无关或各个业务板块都可公用的维度，如时间维度。举例如下：

- 公共区域维表 dim_pub_area
- A公司电商板块的商品全量表dim_asale_itm

建表示例

本例中，最终的维表建表语句如下所示。

```
CREATE TABLE IF NOT EXISTS dim_asale_itm
(
    item_id                             BIGINT COMMENT '商品ID',
    item_title                         STRING COMMENT '商品名称',
    item_price                        DOUBLE COMMENT '商品成交价格_元',
    item_stuff_status                 BIGINT COMMENT '商品新旧程度_0全新1闲
置2二手',
    cate_id                             BIGINT COMMENT '商品类目id',
    cate_name                           STRING COMMENT '商品类目名称',
    commodity_id                         BIGINT COMMENT '品类ID',
    commodity_name                     STRING COMMENT '品类名称',
    umid                              STRING COMMENT '买家id',
    item_status                       BIGINT COMMENT '商品状态_0正常1用户删
除2下架3未上架',
    city                              STRING COMMENT '商品所在城市',
    prov                              STRING COMMENT '商品所在省份',
)
COMMENT '商品全量表'
PARTITIONED BY (ds        STRING COMMENT '日期,yyyymmdd');

CREATE TABLE IF NOT EXISTS dim_pub_area
(
    buyer_id       STRING COMMENT '买家ID'
    city_code      STRING COMMENT '城市code',
    city_name      STRING COMMENT '城市名称',
    prov_code      STRING COMMENT '省份code',
    prov_name      STRING COMMENT '省份名称',
)
COMMENT '公共区域维表'
PARTITIONED BY (ds             STRING COMMENT '日期分区，格式yyyymmdd')
```

```
LIFECYCLE 3600;
```

1.3.4.3 明细粒度事实层（DWD）

明细粒度事实层以业务过程作为建模驱动，基于每个具体的业务过程特点，构建最细粒度的明细层事实表。您可以结合企业的数据使用特点，将明细事实表的某些重要维度属性字段做适当冗余，即宽表化处理。

公共汇总粒度事实层（DWS）和明细粒度事实层（DWD）的事实表作为数据仓库维度建模的核心，需紧绕业务过程来设计。通过获取描述业务过程的度量来表达业务过程，包括引用的维度和与业务过程有关的度量。度量通常为数值型数据，作为事实逻辑表的事实。事实属性则作为事实逻辑表的描述信息，关联维度则将事实属性中的外键字段关联对应维度。

事实表中一条记录所表达的业务细节程度被称为粒度。通常粒度可以通过两种方式来表述：一种是维度属性组合所表示的细节程度，一种是所表示的具体业务含义。

作为度量业务过程的事实，通常为整型或浮点型的十进制数值，有可加性、半可加性和不可加性三种类型：

- 可加性事实是指可以按照与事实表关联的任意维度进行汇总。
- 半可加性事实只能按照特定维度汇总，不能对所有维度汇总。例如库存可以按照地点和商品进行汇总，而按时 间维度把一年中每个月的库存累加则毫无意义。
- 完全不可加性，例如比率型事实。对于不可加性的事实，可分解为可加的组件来实现聚集。

事实表相对维表通常更加细长，行增加速度也更快。维度属性可以存储到事实表中，这种存储到事实表中的维度列称为维度退化，可加快查询速度。与其他存储在维表中的维度一样，维度退化可以用来进行事实表的过滤查询、实现聚合操作等。

明细粒度事实层（DWD）通常分为三种：事务事实表、周期快照事实表和累积快照事实表，详情请参见数仓建设规范管理指南。事务事实表用来描述业务过程，跟踪空间或时间上某点的度量事件，保存的是最原子的数据，也称为原 子事实表。周期快照事实表以具有规律性的、可预见的时间间隔记录事实。累积快照事实表用来表述过程开始和结束之间的关键步骤事件，覆盖过程的整个生命周期，通常具有多个日期字段来记录关键时间点。当累积快照事实表随着生命周期不断变化时，记录也会随着过程的变化而被修改。

明细粒度事实表设计原则

明细粒度事实表设计原则如下所示：

- 通常，一个明细粒度事实表仅和一个维度关联。
- 尽可能包含所有与业务过程相关的事实 。
- 只选择与业务过程相关的事实。
- 分解不可加性事实为可加的组件。

- 在选择维度和事实之前必须先声明粒度。
- 在同一个事实表中不能有多种不同粒度的事实。
- 事实的单位要保持一致。
- 谨慎处理Null值。
- 使用退化维度提高事实表的易用性。

明细粒度事实表整体设计流程如下图所示。

在一致性度量中已定义好了交易业务过程及其度量。明细事实表注意针对业务过程进行模型设计。明细事实表的设计可以分为四个步骤：选择业务过程、确定粒度、选择维度、确定事实(度量)。粒度主要是在维度未展开的情况下记录业务活动的语义描述。在您建设明细事实表时，需要选择基于现有的表进行明细层数据的开发，清楚所建表记录存储的是什么粒度的数据。

明细粒度事实层（DWD）规范

通常您需要遵照的命名规范为：dwd_{业务板块/pub}_{数据域缩写}_{业务过程缩写}[_{自定义表命名标签缩写}] _{单分区增量全量标识}，pub表示数据包括多个业务板块的数据。单分区增量全量标识通常为：i：表示增量，f表示全量。例如： dwd_asale_trd_ordcrt_trip_di（A电商公司航旅机票订单下单事实表，日刷新增量）及dwd_asale_itm_item_df（A电商商品快照事实表，日刷新全量）。

本教程中，DWD层主要由三个表构成：

- 交易商品信息事实表：dwd_asale_trd_itm_di。
- 交易会员信息事实表：ods_asale_trd_mbr_di。
- 交易订单信息事实表：dwd_asale_trd_ord_di。

DWS层数据存储及生命周期管理规范请参见CDM明细层设计规范。

建表示例

本教程中充分使用了维度退化以提升查询效率，建表语句如下所示。

```
CREATE TABLE IF NOT EXISTS dwd_asale_trd_itm_di
(
    item_id               BIGINT COMMENT '商品ID',
    item_title            STRING COMMENT '商品名称',
    item_price            DOUBLE COMMENT '商品价格',
    item_stuff_status     BIGINT COMMENT '商品新旧程度_0全新1闲置2二手',
    item_prov             STRING COMMENT '商品省份',
    item_city             STRING COMMENT '商品城市',
    cate_id               BIGINT COMMENT '商品类目id',
    cate_name             STRING COMMENT '商品类目名称',
    commodity_id          BIGINT COMMENT '品类ID',
```

```
    commodity_name       STRING COMMENT '品类名称',
    buyer_id             BIGINT COMMENT '买家id',
)
COMMENT '交易商品信息事实表'
PARTITIONED BY (ds     STRING COMMENT '日期')
LIFECYCLE 400;

CREATE TABLE IF NOT EXISTS ods_asale_trd_mbr_di
(
    order_id         BIGINT COMMENT '订单ID',
    bc_type          STRING COMMENT '业务分类',
    buyer_id         BIGINT COMMENT '买家id',
    buyer_nick       STRING COMMENT '买家昵称',
    buyer_star_id    BIGINT COMMENT '买家星级id',
    seller_id        BIGINT COMMENT '卖家ID',
    seller_nick      STRING COMMENT '卖家昵称',
    seller_star_id   BIGINT COMMENT '卖家星级id',
    shop_id          BIGINT COMMENT '店铺id',
    shop_name        STRING COMMENT '店铺名称'
)
COMMENT '交易会员信息事实表'
PARTITIONED BY (ds     STRING COMMENT '日期')
LIFECYCLE 400;

CREATE TABLE IF NOT EXISTS dwd_asale_trd_ord_di
(
    order_id              BIGINT COMMENT '订单ID',
    pay_order_id          BIGINT COMMENT '支付订单ID',
    pay_status            BIGINT COMMENT '支付状态_1未付款2已付款3已退款',
    succ_time             STRING COMMENT '订单交易结束时间',
    item_id               BIGINT COMMENT '商品ID',
    item_quantity         BIGINT COMMENT '购买数量',
    confirm_paid_amt      DOUBLE COMMENT '订单已经确认收货的金额',
    logistics_id          BIGINT COMMENT '物流订单id',
    mord_prov             STRING COMMENT '收货人省份',
    mord_city             STRING COMMENT '收货人城市',
    mord_lgt_shipping     BIGINT COMMENT '发货方式_1平邮2快递3EMS',
    mord_address          STRING COMMENT '收货人地址',
    mord_mobile_phone     STRING COMMENT '收货人手机号',
    mord_fullname         STRING COMMENT '收货人姓名',
    buyer_nick            STRING COMMENT '买家昵称',
    buyer_id              BIGINT COMMENT '买家ID'
)
COMMENT '交易订单信息事实表'
PARTITIONED BY (ds       STRING COMMENT '日期')
LIFECYCLE 400;
```

1.3.4.4 公共汇总粒度事实层（DWS）

公共汇总粒度事实层以分析的主题对象作为建模驱动，基于上层的应用和产品的指标需求构建公共粒度的汇总指标事实表。公共汇总层的一个表通常会对应一个派生指标。

公共汇总事实表设计原则

聚集是指针对原始明细粒度的数据进行汇总。DWS公共汇总层是面向分析对象的主题聚集建模。在本教程中，最终的分析目标为：最近一天某个类目（例如：厨具）商品在各省的销售总额、该类目Top10销售额商品名称、各省用户购买力分布。因此，我们可以以最终交易成功的商品、类目、买家等角度对最近一天的数据进行汇总。数据聚集的注意事项如下：

- 聚集是不跨越事实的。聚集是针对原始星形模型进行的汇总。为获取和查询与原始模型一致的结果，聚集的维度和度量必须与原始模型保持一致，因此聚集是不跨越事实的。
- 聚集会带来查询性能的提升，但聚集也会增加ETL维护的难度。当子类目对应的一级类目发生变更时，先前存在的、已经被汇总到聚集表中的数据需要被重新调整。

此外，进行DWS层设计时还需遵循以下原则：

- 数据公用性：需考虑汇总的聚集是否可以提供给第三方使用。您可以思考，基于某个维度的聚集是否经常用于数据分析中。如果答案是肯定的，就有必要把明细数据经过汇总沉淀到聚集表中。
- 不跨数据域。数据域是在较高层次上对数据进行分类聚集的抽象。数据域通常以业务过程进行分类，如交易统一划到交易域下， 商品的新增、修改放到商品域下。
- 区分统计周期。在表的命名上要能说明数据的统计周期，如_1d 表示最近1天， td 表示截至当天，nd 表示最近N天。

公共汇总事实表规范

公共汇总事实表命名规范：dws_{业务板块缩写/pub}_{数据域缩写}_{数据粒度缩写}[_{自定义表命名标签缩写}]_{统计时间周期范围缩写}。

- 关于统计实际周期范围缩写，缺省情况下，离线计算应该包括最近一天（_1d）,最近N天（_nd）和历史截至当天(_td)三个表。如果出现_nd的表字段过多需要拆分时，只允许以一个统计周期单元作为原子拆分。即一个统计周期拆分一个表，例如最近7天(_1w)拆分一个表。不允许拆分出来的一个表存储多个统计周期。
- 对于小时表[无论是天刷新还是小时刷新], 都用_hh 来表示。
- 对于分钟表[无论是天刷新还是小时刷新],都用_mm来表示。

举例如下：

- dws_asale_trd_byr_subpay_1d (A电商公司买家粒度交易分阶段付款一日汇总事实表)
- dws_asale_trd_byr_subpay_td（A电商公司买家粒度分阶段付款截至当日汇总表）
- dws_asale_trd_byr_cod_nd (A电商公司买家粒度货到付款交易汇总事实表)
- dws_asale_itm_slr_td (A电商公司卖家粒度商品截至当日存量汇总表)
- dws_asale_itm_slr_hh (A电商公司卖家粒度商品小时汇总表)---维度为小时
- dws_asale_itm_slr_mm (A电商公司卖家粒度商品分钟汇总表)---维度为分钟

DWS层数据存储及生命周期管理规范请参见CDM汇总层设计规范。

建表示例

满足业务需求的DWS层建表语句如下。

```
CREATE TABLE IF NOT EXISTS dws_asale_trd_byr_ord_1d
(
```

```
    buyer_id                 BIGINT COMMENT '买家id',
    buyer_nick               STRING COMMENT '买家昵称',
    mord_prov                STRING COMMENT '收货人省份',
    cate_id                  BIGINT COMMENT '商品类目id',
    cate_name                STRING COMMENT '商品类目名称',
    confirm_paid_amt_sum_1d DOUBLE COMMENT '最近一天订单已经确认收货的金额
总和'
)
COMMENT '买家粒度所有交易最近一天汇总事实表'
PARTITIONED BY (ds          STRING COMMENT '分区字段YYYYMMDD')
LIFECYCLE 36000;

CREATE TABLE IF NOT EXISTS dws_asale_trd_itm_ord_1d
(
    item_id                  BIGINT COMMENT '商品ID',
    item_title                STRING COMMENT '商品名称',
    cate_id                  BIGINT COMMENT '商品类目id',
    cate_name                STRING COMMENT '商品类目名称',
    mord_prov                STRING COMMENT '收货人省份',
    confirm_paid_amt_sum_1d DOUBLE COMMENT '最近一天订单已经确认收货的金额
总和'
)
COMMENT '商品粒度交易最近一天汇总事实表'
PARTITIONED BY (ds          STRING COMMENT '分区字段YYYYMMDD')
LIFECYCLE 36000;
```

1.3.4.5 附录：示例数据

本文为您提供ODS层各表格的示例数据，仅供您测试参考。

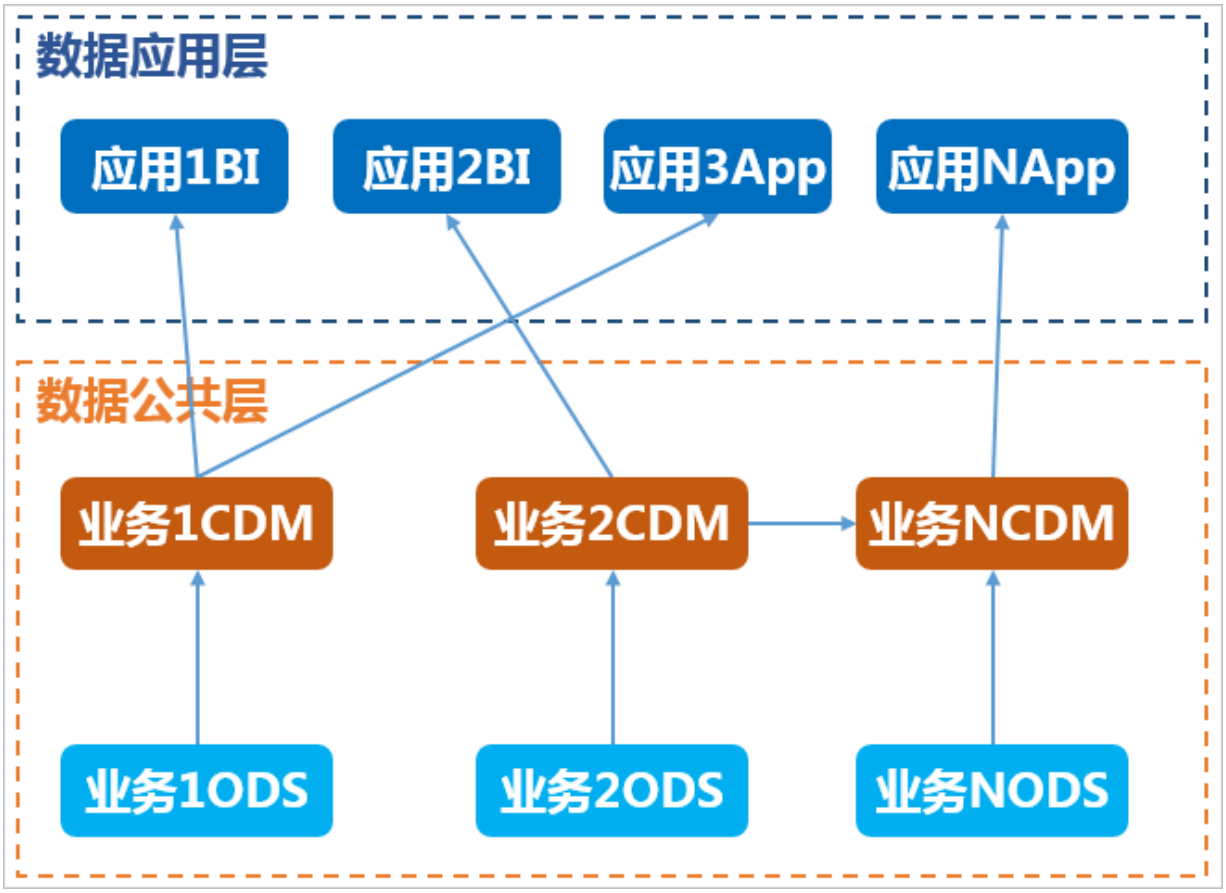
- s_auction.csv
- s_biz_order_delta.csv
- s_logistics_order_delta.csv
- s_pay_order_delta.csv
- s_sale.csv
- s_users_extra.csv

1.4 项目分配与安全

在为企业级大数据平台创建项目时，建议您对于ODS层、DWD及DWS层的数据按照业务板块的粒度建立项目，对于ADS层的数据，按照应用的粒度建立项目。

项目分配

在本教程中，建议您参考下图建立您的MaxCompute项目，图中的每一个方块代表一个项目。



- 对于ODS层项目，建议以ods为后缀，例如asaleods。
- 对于CDM层项目，建议以cdm为后缀，例如asalecdm。
- ADS应用层数据分为两类：
  - 数据报表、数据分析等以bi作为后缀，例如asalebi。
  - 数据产品应用以app作为后缀，例如asaleapp。

考虑到本教程仅聚焦于电商业务板块中交易成功的业务流程，您可以为ODS、CDM和ADS层分别仅建立一个项目。

项目模式选择

标准模式是指一个Dataworks项目对应两个MaxCompute项目，可设置开发和生产双环境，提升代码开发规范，并能够对表权限进行严格控制，禁止随意操作生产环境的表，保证生产表的数据安全。

当您在DataWorks建立项目时，建议您使用标准模式以保证生成环境项目安全，详情请参见简单模式和标准模式的区别。

完成项目创建后，您会得到一个生产环境项目和以_dev结尾的开发环境项目。例如`asaleods`**和**`asaleods_dev`**。**

项目权限配置

您需要重点考虑为项目中的不同成员角色赋予不同的权限，例如生产任务如何保障不可随意变更、哪些成员可以进行代码编辑调试、哪些成员可以进行发布生产任务等。同时要为在数据开发过程中的资源使用赋权，并做好数据安全隔离。

关于MaxCompute数仓安全和权限配置详情，请参见安全模型。

1.5 建立性能基准

MaxCompute性能表现优劣，主要取决您的表设计是否符合规范。为方便您衡量MaxCompute表的性能表现，建议您在优化之前首先建立性能基准。

> 说明:
> MaxCompute表设计规范详情请参见表设计规范。

在优化表前后测试系统性能时，您需要记录每张表的数据同步时间、占用存储大小以及查询性能的详细信息。如果您使用的是预付费（包年包月）方式购买的MaxCompute项目资源，您还需要记录购买数。

测试项	测试值
数据同步时间	
占用存储大小	
查询执行时间	
查询费用预估	

记录数据同步时间

在您执行数据同步任务后，可以在运维中心 > 周期实例的页面右键查看用户任务运行时间，如下图所示。

```
].
2019-01-11 00:30:38.871 [33181128-0-0-writer] INFO  OdpsWriter$Task - Slave which uploadId=[20190111002916[illegible]] commit blocks ok.
2019-01-11 00:30:39.346 [taskGroup-0] INFO  TaskGroupContainer - taskGroup[0] taskId[0] is successed, used[82532]ms
2019-01-11 00:30:39.346 [taskGroup-0] INFO  TaskGroupContainer - taskGroup[0] completed it's tasks.
Exit with SUCCESS.
2019-01-11 00:30:53 [INFO] Sandbox context cleanup temp file success.
2019-01-11 00:30:53 [INFO] Data synchronization ended with return code: [0].
2019-01-11 00:30:53 INFO ================================================================
2019-01-11 00:30:53 INFO Exit code of the Shell command 0
2019-01-11 00:30:53 INFO --- Invocation of Shell command completed ---
2019-01-11 00:30:53 INFO Shell run successfully!
2019-01-11 00:30:53 INFO Current task status: FINISH
2019-01-11 00:30:53 INFO Cost time is: 105.46s
/home/admin/alisatasknode/taskinfo//20190111/phoenix/00/29/01/[illegible]/T3_0690678015.log-END-EOF
```

关于如何查看及优化同步任务的详情请参见数据同步任务调优。

记录占用存储大小

您可以使用`describe`命令查看全表或表中某个分区占用物理存储的大小，如下图所示。

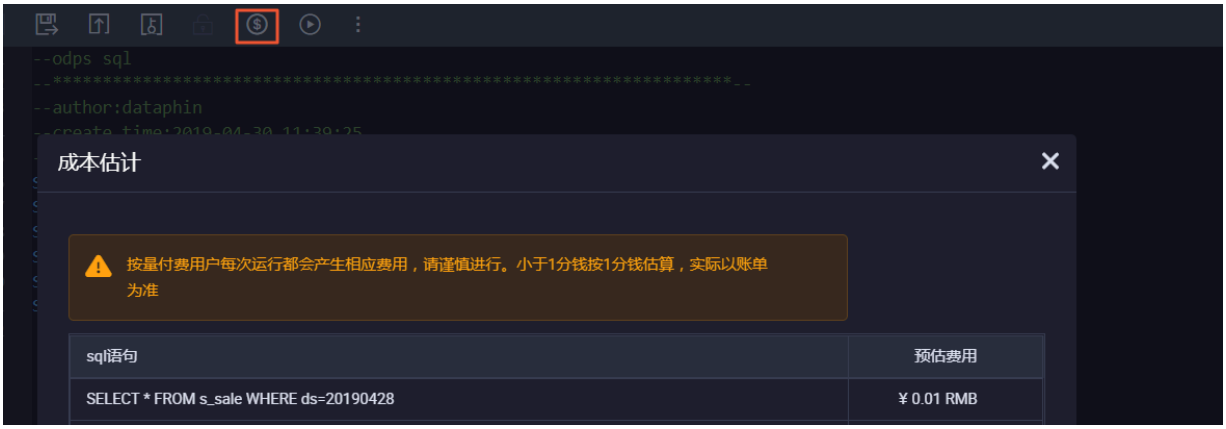
```
--odps sql
--********************************************************************--
--author:dataphin
--create time:2019-05-13 16:08:04
--********************************************************************--
DESCRIBE s_sale;
```

运行日志

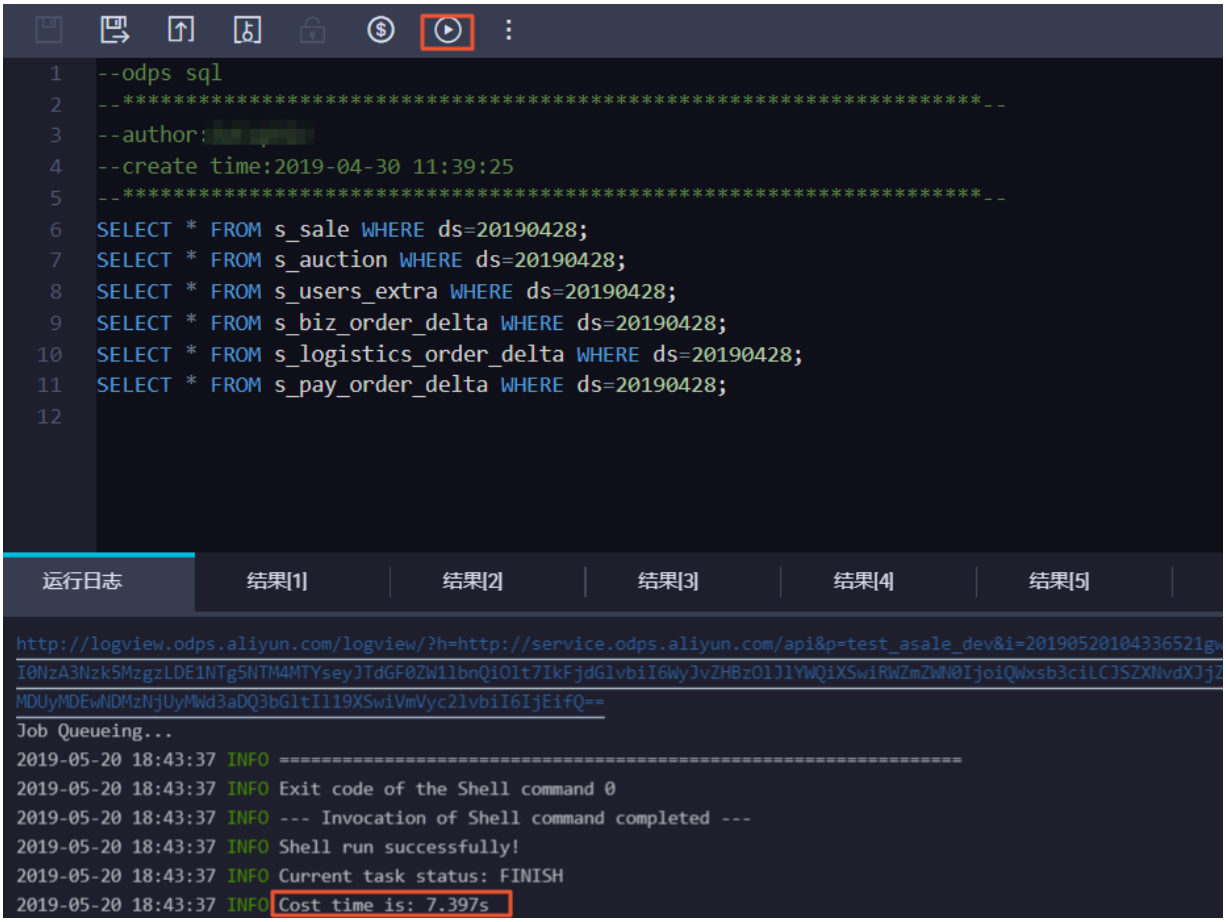
```
+------------------------------------------------------------------------------------+
| Owner: ALIYUN$[redacted]      | Project: test_asale_dev                            |
| TableComment: 正常购买ods                                                          |
+------------------------------------------------------------------------------------+
| CreateTime:               2019-04-30 13:29:03                                      |
| LastDDLTime:              2019-04-30 13:29:03                                      |
| LastModifiedTime:         2019-04-30 19:26:46                                      |
| Lifecycle:                400                                                      |
+------------------------------------------------------------------------------------+
| InternalTable: YES        | Size: 9408                                             |
+------------------------------------------------------------------------------------+
| Native Columns:                                                                    |
+------------------------------------------------------------------------------------+
| Field           | Type          | Label | Comment                                  |
+------------------------------------------------------------------------------------+
```

记录查询执行时间及预估费用

如果您使用DataWorks进行查询，可以在运行任务时或通过点击下列图标直接通过图形页面查看预估费用。



任务完成运行后，可在运行日志中查看到运行时间。



1.6 数仓性能优化

针对数仓的性能优化，主要是针对表和数据分布的优化。表设计的最佳实践请参见表设计最佳实践。

Hash Clustering

Hash Clustering表的优势在于可以实现Bucket Pruning优化、Aggregation优化以及存储优化。在创建表时，使用`clustered by`指定`Hash Key`后，MaxCompute将对指定列进行Hash运算，按照Hash值分散到各个Bucket里。Hash Key值的选择原则为选择重复键值少的列。Hash Clustering表的使用方法详情请参见修改表的Hash Clustering属性。

如何转化为Hash Clustering表：

```
ALTER TABLE table_name [CLUSTERED BY (col_name [, col_name, ...]) [
SORTED BY (col_name [ASC | DESC] [, col_name [ASC | DESC] ...])] INTO
number_of_buckets BUCKETS]
```

`alter table`语句适用于存量表，在增加了新的聚集属性之后，新的分区将做Hash Clustering存储。

创建完Hash Clustering表后，您可以使用`insert overwrite`语句将源表转化为Hash Clustering表。

> 说明:
> Hash Clustering表存在以下限制：
> - 不支持`insert into`语句，只能通过`insert overwrite`来添加数据。
> - 不支持直接使用`tunnel upload`命令将数据导入到range cluster表，因为tunnel上传的数据是无序的。

表的其他优化技巧

建议您严格遵循表设计规范。此外，您还可以利用下列技巧完成表的优化：

- 中间表的利用：适用于数据量非常大，下游任务很多的表。
- 拆表：适用于个别字段产出极慢的情况，您可以讲字段拆分为单独的表。
- 合表：随着数仓的发展，针对业务重叠或重复的表，您可以进行任务和数据合并。
- 拉链表：合理利用拉链表能减少您的存储消耗，关于拉链存储的详情请参见拉链存储。
- 利用MaxCompute表的特殊功能：详情请参见MaxCompute表的特殊功能。

1.7 结果验证

完成数仓的优化后，您需要对结果进行评估验证，确认优化的有效性。

如果您在优化过程中改变了表结构，您需要删除原有的表，并根据优化策略新建表和分区。本教程中提供的测试数据也需要进行对应的结构调整，方便您完成数据的导入。

在重新创建表并导入数据后，您需要开始重新测试数仓性能。您可以通过下列表格记录相关数据，并与性能基准进行比对，性能基准详情请参考建立性能基准。

测试项	测试值
数据同步时间	
占用存储大小	
查询执行时间	
查询费用预估	

2 搭建互联网在线运营分析平台

2.1 业务场景与开发流程

本教程基于大数据时代在线运营分析的平台的基础需求，为开发者提供从数据高并发写入存储、便捷高效的数据加工处理到数据分析与展示的全链路解决方案。本教程助您了解并实操阿里云的大数据产品，完成在线运营分析平台的搭建。

业务场景

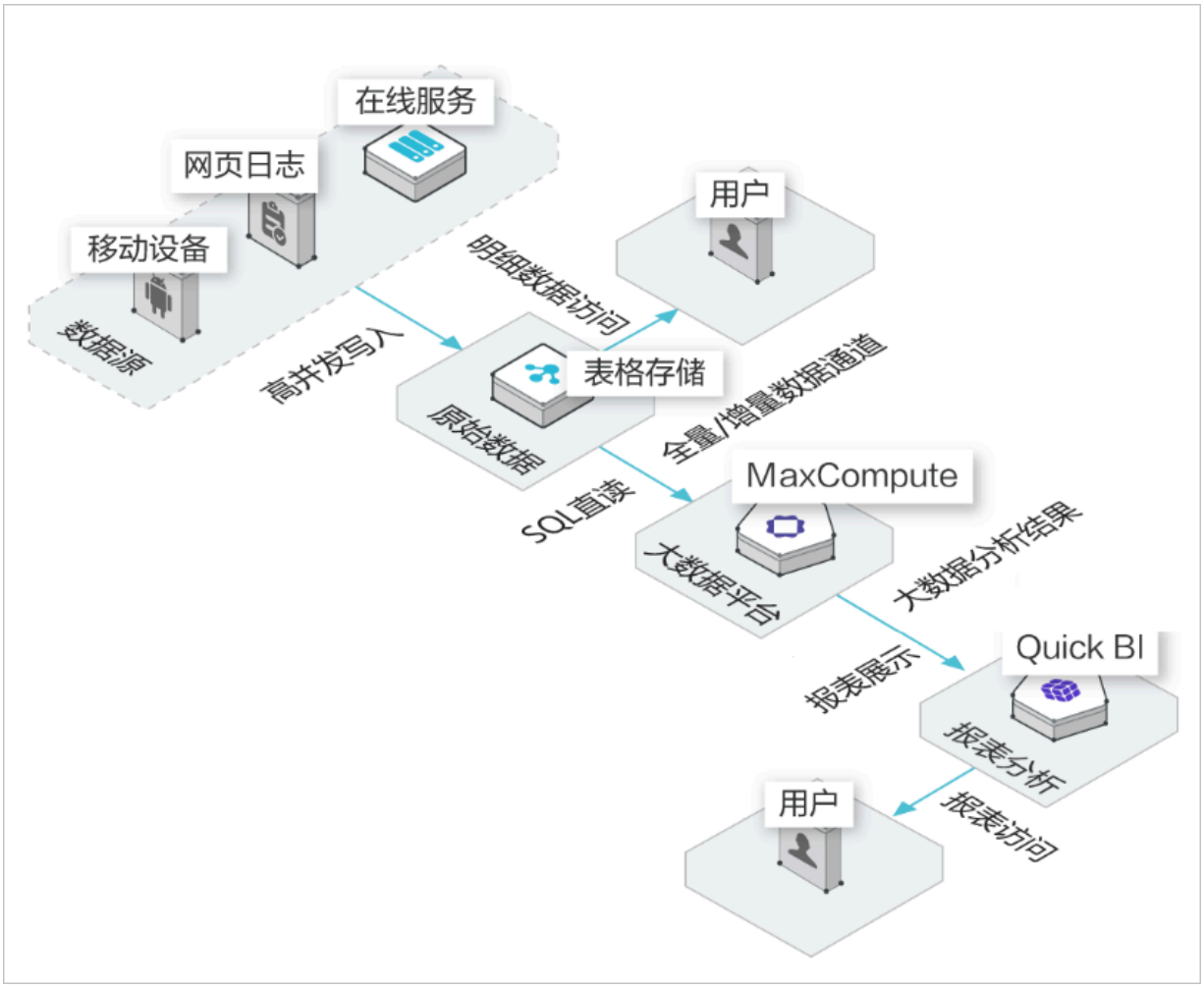
本节的示例说明基于一份真实的网站日志数据集，数据来源于某网站上的HTTP访问日志数据。基于这份网站日志，您可以实现如下分析需求：

- 统计并展现网站的PV和UV，并能够按照用户的终端类型（如Android、iPad、iPhone、PC等）分别统计。

  说明：
  浏览次数（PV）和独立访客（UV）是衡量网站流量的两项最基本指标。用户每打开一个网站页面，记录一个PV，多次打开同一页面PV累计多次。独立访客是指一天内访问网站的不重复用户数，一天内同一访客多次访问网站只计算一次。

- 统计并展现网站的流量来源地域。

开发流程



本教程涉及的具体开发流程如下：

- 步骤一：环境准备。
- 步骤二：数据准备。
- 步骤三：新建数据表。
- 步骤四：设计工作流。
- 步骤五：节点配置。
- 步骤六：任务提交与测试。
- 步骤七：数据可视化展现。

整体数仓研发的规划建议请参见数据仓库研发规范概述。

2.2 环境准备

为保证您可以顺利完成本教程，请您首先确保自己云账号已开通表格存储TableStore、大数据计算服务MaxCompute、数据工场DataWorks和智能分析套件Quick BI。

前提条件

- 阿里云账号注册，详情请参见阿里云账号注册流程。
- 实名认证，详情请参见个人实名认证或企业实名认证。

背景信息

本教程涉及的阿里云产品如下：

- 表格存储 TableStore
- 大数据计算服务 MaxCompute
- 数据工场 DataWorks
- 智能分析套件Quick BI

说明：

在本教程中，表格存储服务选择华北2区域。

操作步骤

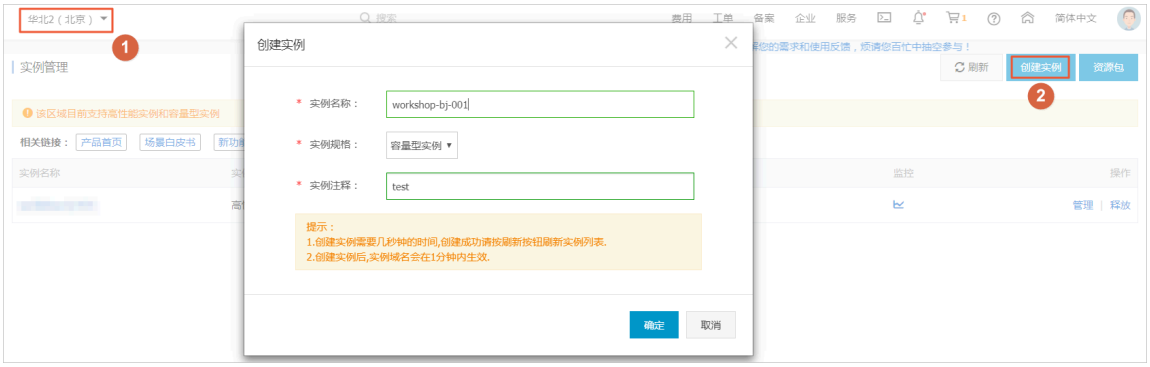
1. 创建表格存储实例

a) 进入表格存储TableStore产品详情页，单击立即开通。

b) 进入开通页面后，单击立即开通。

c) 单击管理控制台。

d) 选择区域为华北2（北京），单击创建实例。填写实例名称，实例规格请选择容量型实例，单击确定完成创建。



说明:

实例名称在表格存储同一个区域内必须全局唯一，建议您选用自己可辨识且符合规则的名称。实例名称在MaxCompute数据处理中也会被实用，本例中为`workshop-bj-001`，关于实例的详细解释请参见#unique_52。

e) **完成创建后，您可以在实例列表 > 实例管理中看到您刚刚创建的实例，状态为运行中。**

2. 开通大数据计算服务MaxCompute

a) 进入MaxCompute产品详情页，单击立即购买。

b) 选择按量付费，选择区域为华东2（上海），规格类型为默认的标准版，单击立即购买。

说明:

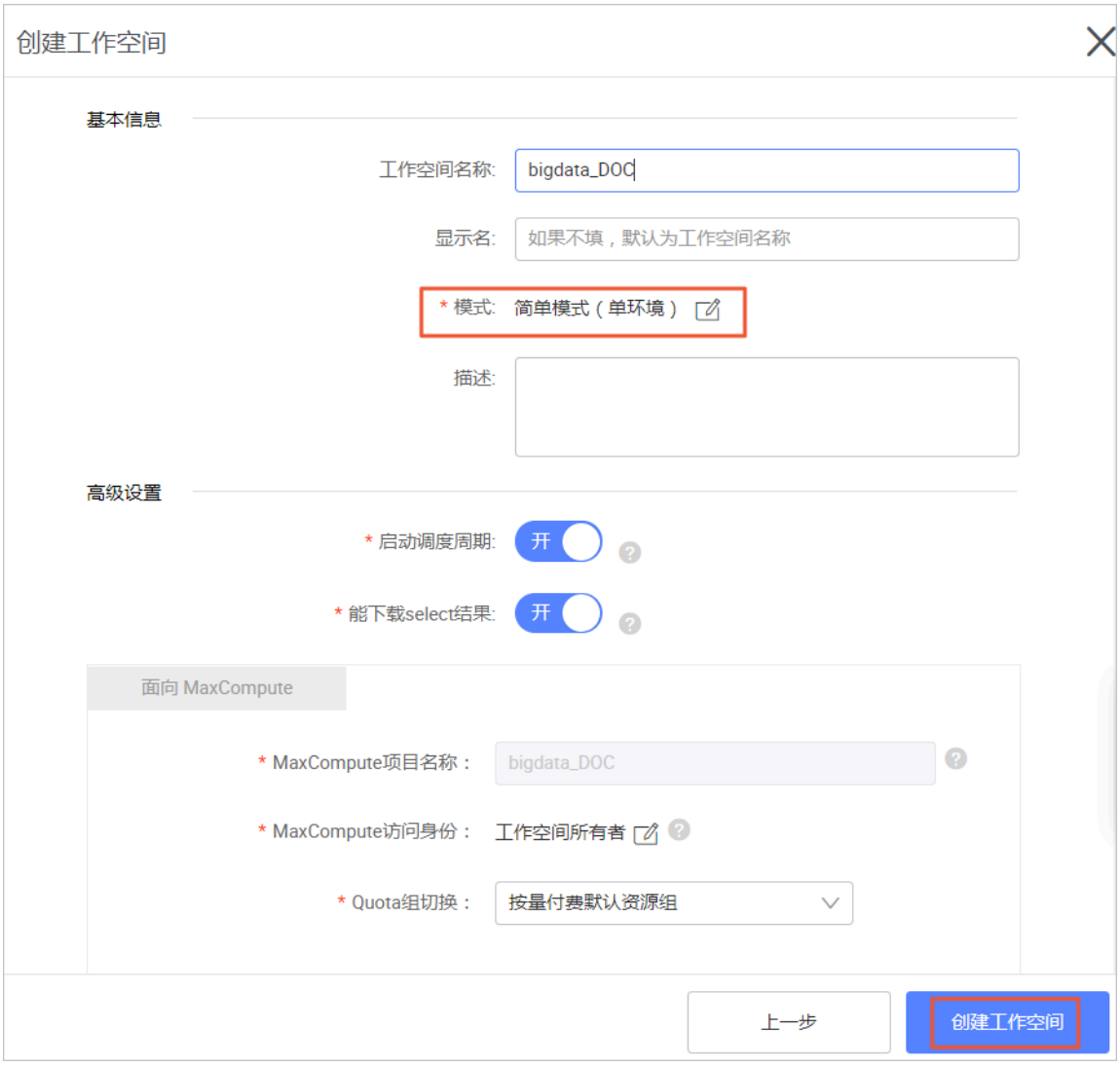
选择MaxCompute区域与表格存储相同可以节省您的流量费用，因此您可以选择区域为华北2（北京）。本例中MaxCompute区域选择为华东2（上海），以便为您展示跨地域的外部表使用过程。

3. 创建DataWorks工作空间

a) 进入DataWorks工作空间列表，选择区域为华东1，单击创建工作空间。

b) 选择计算引擎服务为MaxCompute、按量付费。为方便使用，本教程中DataWorks工作空间模式为简单模式（单环境）。

在简单模式下，DataWorks工作空间与MaxCompute项目一一对应，详情请参见简单模式和标准模式的区别。



说明:

工作空间名称全局唯一，建议您使用易于区分的名称。

4. 开通Quick BI

a) 进入Quick BI产品详情页，单击管理控制台。

b) 进入控制台后，单击高级版30天试用申请或专业版30天试用申请。勾选同意Quick BI服务协议，单击开通试用。成功开通Quick BI专业版试用后的界面如下图所示。

说明:

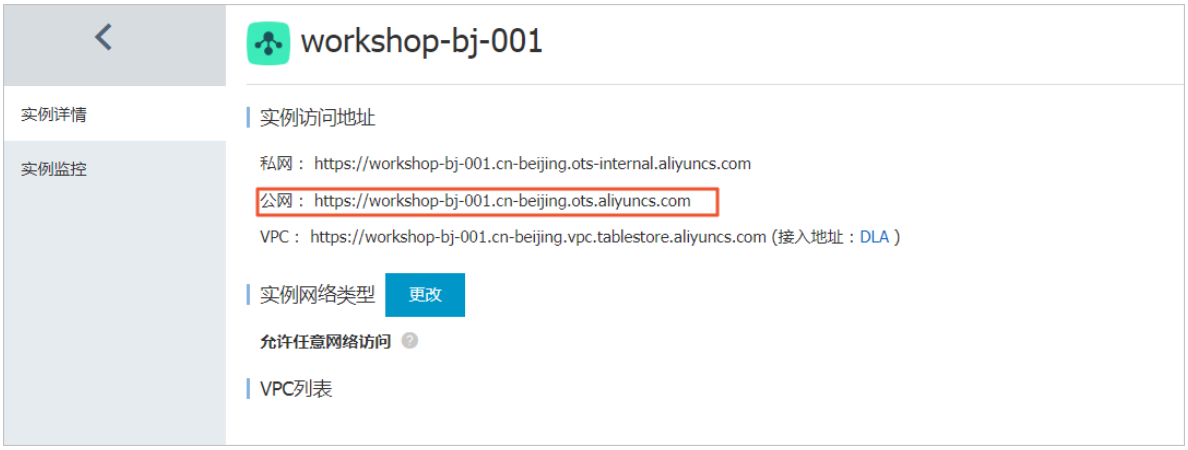
您可以选择使用个人空间或默认空间，推荐您使用默认空间。

2.3 数据准备

在数据准备阶段，您需要通过数据Demo包生成出模拟真实环境的数据，以便后续数据开发使用。

前提条件

1. 请您首先参考环境准备，创建华北2区域的表格存储示例，同时记录实例名称和实例访问地址。单击表格存储控制台中的实例名称之后，您可以获得实例访问地址。对于跨区域的访问，建议您使用公网地址。



2. 使用主账号登录安全信息管理控制台，获取并记录您的AccessKey ID和AccessKey Secret信息。

> **说明:**
> AccessKey ID和AccessKey Secret是您访问阿里云API的密钥，具有该账户完全的权限，请您妥善保管。

操作步骤

1. 下载数据Demo包

   数据Demo包下载地址如下，本例中使用环境为Windows7 64位：

   - Mac下载地址
   - Linux 下载地址
   - Windows7 64位 下载地址

2. 配置Demo环境

完成下载后，您需要解压下载包，编辑conf文件夹内的app.conf文件。

app.conf文件内容举例如下，其中endpoint信息即为您的实例访问地址。

```
endpoint = "https://workshop-bj-001.cn-beijing.ots.aliyuncs.com"
instanceName = "workshop-bj-001"
accessKeyId = "LTAIF24u7g******"
accessKeySecret = "CcwFeF3sWTPy0wsKULMw34Px******"
usercount = "200"
daysCount = "7"
```

3. 启动Demo准备测试数据

启动Windows CMD命令行工具，进入您解压缩Demo包的路径，您可以使用workshop_demo.exe -h命令查看Demo包命令用法。

```
workshop_demo.exe -h 会列出该demo的相关命令:
* prepare:  准备测试数据，创建数据表，根据conf中的用户数量，为用户生成一周的行为日志数据。
* raw ${userid} ${date} ${Top条数}:  查询指定用户的日志明细。
```

* new/day_active/month_active/day_pv/month_pv：在结果表中查询上述几种类型的报表数据（新增：new，日活：day_active，月活：month_active，日PV：day_pv，月PV：month_pv）。

使用workshop_demo.exe prepare命令生成准备数据。

```
C:\Users [redacted]\workshop_demo>workshop_demo.exe prepare
OTSObjectAlreadyExist Requested table already exists.
OTSObjectAlreadyExist Requested table already exists.
Prepare the metric data
Prepare the metric data
Prepare the metric data
Prepare the metric data
Prepare the metric data
Prepare User data
finished one round
finished one round
finished one round
finished one round
finished one round
total insert data count is: 41757
```

在这个过程中，Demo包会自动帮助您在表格存储中创建表，结构如下：

· 原始日志数据表：user_trace_log

列名	类型	说明
md5	STRING	用户uid的md5值 undefined前8位，表格存储主键。
uid	STRING	用户uid，表格存储主键。
ts	BIGINT	用户操作时间戳，表格存储主键。
ip	STRING	IP地址
status	BIGINT	服务器返回状态码
bytes	BIGINT	返回给客户端的字节数
device	STRING	终端型号
system	STRING	系统版本：ios xxx/android xxx 。
customize_event	STRING	自定义事件：登录/退出/购买/注册/点击/后台/切换用户/浏览。
use_time	BIGINT	APP单次使用时长，当事件为退出、后台、切换用户时有该项。

列名	类型	说明
customize_event_content	STRING	用户关注的内容信息。

- 分析结果表： analysis_result

列名	类型	说明
metric	STRING	报表的类型：'new'、'day_active'、'month_active'、'day_pv'、'month_pv'，表格存储主键。
ds	STRING	时间yyyy-mm-dd或yyyy-mm，表格存储主键。
num	BIGINT	对应的数据值

4. 数据验证

- 用户明细查询

表格数据对应的日期对应于您创建表格的时间，例如您创建数据时间为2019年6月15日，则可以使用`workshop_demo.exe raw 00010 "2019-06-15" 20`查看20条用户明细数据。

```
C [redacted]nloads\workshop_demo>workshop_demo.exe raw 00010 "2019-06-1
5" 20
       uid                          Date    bytes      customize_event
      device                    ip   status          system
     00010         2019-06-14 11:56:47 PM      759               regist
iPhone7 Plus        61.103.79.217      200          ios11
     00010         2019-06-14 11:26:34 PM      252     backstage    369
   iPad min2       157.249.67.241      200          ios11
     00010         2019-06-14 11:21:30 PM      427        browse travel
    iPhone6s      222.133.108.234      200          ios10
     00010         2019-06-14 11:16:03 PM      764        switch    185
iPhone7 Plus        61.103.79.217      200          ios11
     00010         2019-06-14 11:06:03 PM      436                click
iPhone7 Plus        61.103.79.217      200          ios11
     00010         2019-06-14 10:36:54 PM      131                click
iPhone7 Plus        61.103.79.217      200          ios11
     00010         2019-06-14 10:22:26 PM      778        switch     73
    iPhone6s      222.133.108.234      200          ios10
     00010         2019-06-14 10:06:29 PM      535     backstage    179
   iPad min2       157.249.67.241      200          ios11
     00010         2019-06-14 09:56:11 PM      668                click
   iPad min2       157.249.67.241      200          ios11
     00010         2019-06-14 09:20:45 PM      354               regist
    iPhone6s      222.133.108.234      200          ios10
     00010         2019-06-14 09:15:37 PM      989                click
   iPad min2       157.249.67.241      200          ios11
     00010         2019-06-14 08:51:17 PM      460        logout    462
    iPhone6s      222.133.108.234      200          ios10
     00010         2019-06-14 08:26:06 PM      887       comment  funny
   iPad min2       157.249.67.241      200          ios11
     00010         2019-06-14 08:10:34 PM      278       browse finance
    iPhone6s      222.133.108.234      200          ios10
     00010         2019-06-14 07:56:00 PM      480                click
iPhone7 Plus        61.103.79.217      200          ios11
     00010         2019-06-14 07:30:11 PM       68                click
    iPhone6s      222.133.108.234      200          ios10
     00010         2019-06-14 07:15:09 PM      398        browse   news
iPhone7 Plus        61.103.79.217      200          ios11
     00010         2019-06-14 07:11:21 PM       21                click
    iPhone6s      222.133.108.234      200          ios10
     00010         2019-06-14 06:35:07 PM      207        browse  photo
iPhone7 Plus        61.103.79.217      200          ios11
     00010         2019-06-14 06:24:43 PM      261               regist
iPhone7 Plus        61.103.79.217      200          ios11
```

说明:

由于表格存储是SchemaFree结构，表的属性列不需要预先定义。customize_event 中不同的事件对应了不同的内容，因此Demo中将事件-内容做了对齐显示。

- 报表结果查询

  您可以使用`workshop_demo.exe day_active`命令查看日活数据。

```
C:\ workshop_demo>workshop_demo.exe day_active
          metric                  ds                 num
      day_active          2019-05-19             1416104
      day_active          2019-05-20             1416540
      day_active          2019-05-21             1422314
      day_active          2019-05-22             1422411
      day_active          2019-05-23             1428480
      day_active          2019-05-24             1431989
      day_active          2019-05-25             1436218
      day_active          2019-05-26             1437886
      day_active          2019-05-27             1440633
      day_active          2019-05-28             1444736
      day_active          2019-05-29             1450520
      day_active          2019-05-30             1451543
      day_active          2019-05-31             1457510
      day_active          2019-06-01             1458998
      day_active          2019-06-02             1466801
      day_active          2019-06-03             1468898
      day_active          2019-06-04             1473173
      day_active          2019-06-05             1479770
      day_active          2019-06-06             1483101
      day_active          2019-06-07             1484922
      day_active          2019-06-08             1485347
      day_active          2019-06-09             1492034
      day_active          2019-06-10             1499914
      day_active          2019-06-11             1495458
      day_active          2019-06-12             1500697
      day_active          2019-06-13             1508061
      day_active          2019-06-14             1509108
      day_active          2019-06-15             1510583
      day_active          2019-06-16             1518355
      day_active          2019-06-17             1520938
```

2.4 数据建模与开发

2.4.1 新建数据表

为方便在MaxCompute上对数据进行加工处理，首先您需要在MaxCompute上建立数据表，用于承载原始数据及加工后的数据。

前提条件

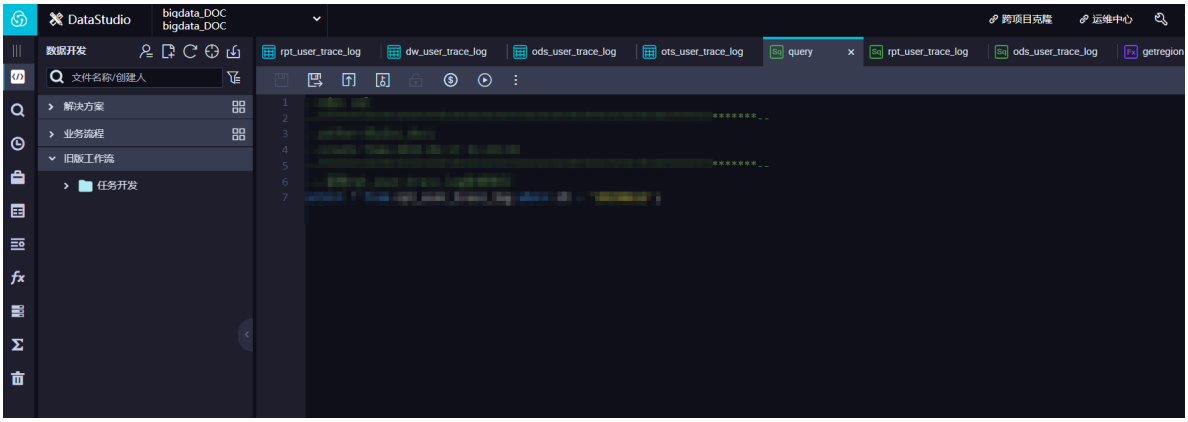
请您参见环境准备章节，完成数据计算服务MaxCompute的开通和DataWorks工作空间的创建。

操作步骤

1. 进入DataWorks工作空间

进入DataWorks工作空间列表，选择区域为华东1，双击您创建好的工作空间（项目）。

双击之后，即可进入工作空间的数据开发界面。



2. 新建数据表

本示例通过DataWorks表管理功能新建数据表。为了与表格存储联动，创建的OTS表类型为MaxCompute外部表，作为原始数据提供层。为满足外部表的授权条

件，当MaxCompute和TableStore的Owner是同一个账号时，您可以单击此处一键授权，详情请参见访问OTS非结构化数据。

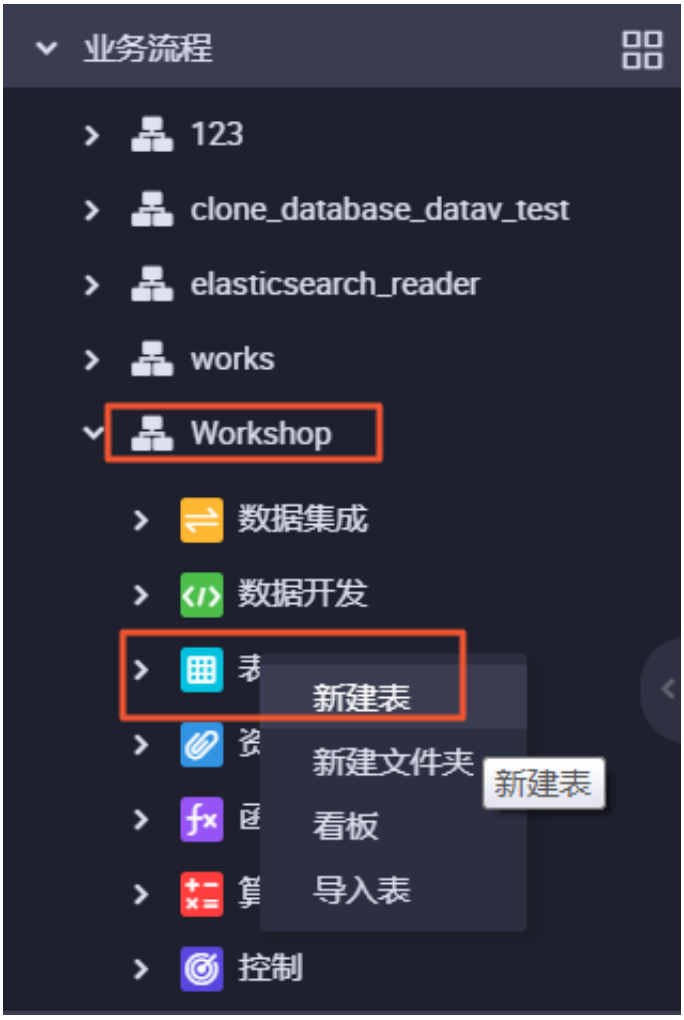
a) 创建业务流程

A. 右键单击业务流程，选择新建业务流程。

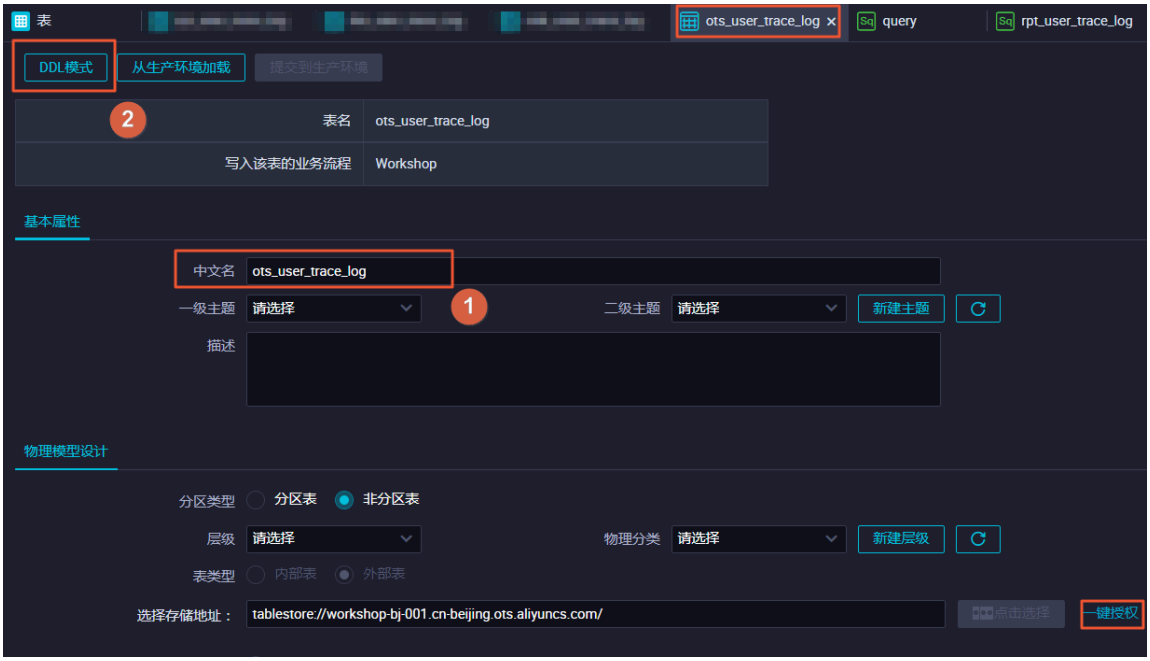
B. 填写业务名称和描述，单击新建。本教程中，业务流程名为Workshop。

b) 创建外部表ots_user_trace_log

双击您新建的业务流程，右键单击表，选择新建表输入您的表名ots_user_trace_log。



填写您的表中文名称，如果您之前未进行一键授权，此时也可以继续完成授权。然后单击DDL模式，开始编辑建表语句。



本例使用的建表语句如下，请您参考环境准备章节，根据自己的表格存储实例访问地址参数填写LOCATION地址。完成填写后单击提交到生产环境。

```
CREATE EXTERNAL TABLE `ots_user_trace_log` (
    `md5` string COMMENT '用户uid的md5值前8位',
    `uid` string COMMENT '用户uid',
    `ts` bigint COMMENT '用户操作时间戳',
    `ip` string COMMENT 'ip地址',
    `status` bigint COMMENT '服务器返回状态码',
    `bytes` bigint COMMENT '返回给客户端的字节数',
    `device` string COMMENT '终端型号',
    `system` string COMMENT '系统版本ios xxx/android xxx',
    `customize_event` string COMMENT '自定义事件：登录/退出/购买/注册/点击/后台/切换用户/浏览',
    `use_time` bigint COMMENT 'APP单次使用时长，当事件为退出、后台、切换用户时有该项',
    `customize_event_content` string COMMENT '用户关注内容信息，在customize_event为浏览和评论时 包含该列'
)
STORED BY 'com.aliyun.odps.TableStoreStorageHandler'
WITH SERDEPROPERTIES (
    'tablestore.columns.mapping'=':md5,:uid,:ts, ip,status,bytes,device,system,customize_event,use_time,customize_event_content',
    'tablestore.table.name'='user_trace_log'
)
LOCATION 'tablestore://workshop-bj-001.cn-beijing.ots.aliyuncs.com/';
```

说明：

如果您使用LOCATION 'tablestore://workshop-bj-001.cn-beijing.ots.aliyuncs.com/'报错，显示网络不同，可尝试更换为LOCATION 'tablestore://workshop-bj-001.cn-beijing.ots-interna.aliyuncs.com/'。

c) 创建ods_user_trace_log表

建表方法同上，建表语句如下，完成填写后单击提交到生产环境。ods_user_trace_log为ODS层表，相关数仓模型定义请参见数据引入层（ODS）。

```
CREATE TABLE IF NOT EXISTS ods_user_trace_log (
    md5 STRING COMMENT '用户uid的md5值前8位',
    uid STRING COMMENT '用户uid',
    ts BIGINT COMMENT '用户操作时间戳',
    ip STRING COMMENT 'ip地址',
    status BIGINT COMMENT '服务器返回状态码',
    bytes BIGINT COMMENT '返回给客户端的字节数',
    device STRING COMMENT '终端型号',
    system STRING COMMENT '系统版本ios xxx/android xxx',
    customize_event STRING COMMENT '自定义事件：登录/退出/购买/注册/点击/后台/切换用户/浏览',
    use_time BIGINT COMMENT 'APP单次使用时长，当事件为退出、后台、切换用户时有该项',
    customize_event_content STRING COMMENT '用户关注内容信息，在customize_event为浏览和评论时 包含该列'
)
PARTITIONED BY (
    dt STRING
```

```
);
```

d) 创建dw_user_trace_log表

建表方法同上，建表语句如下，完成填写后单击提交到生产环境。dw_user_trace_log为DW层表，相关数仓模型定义请参见明细粒度事实层（DWD）。

```
CREATE TABLE IF NOT EXISTS dw_user_trace_log (
    uid STRING COMMENT '用户uid',
    region STRING COMMENT '地域，根据ip得到',
    device_brand string comment '设备品牌',
    device STRING COMMENT '终端型号',
    system_type STRING COMMENT '系统类型，Android、IOS、ipad、
Windows_phone',
    customize_event STRING COMMENT '自定义事件：登录/退出/购买/注册/点
击/后台/切换用户/浏览',
    use_time BIGINT COMMENT 'APP单次使用时长，当事件为退出、后台、切换用
户时有该项',
    customize_event_content STRING COMMENT '用户关注内容信息，在
customize_event为浏览和评论时 包含该列'
)
PARTITIONED BY (
    dt STRING
);
```

e) 创建rpt_user_trace_log表

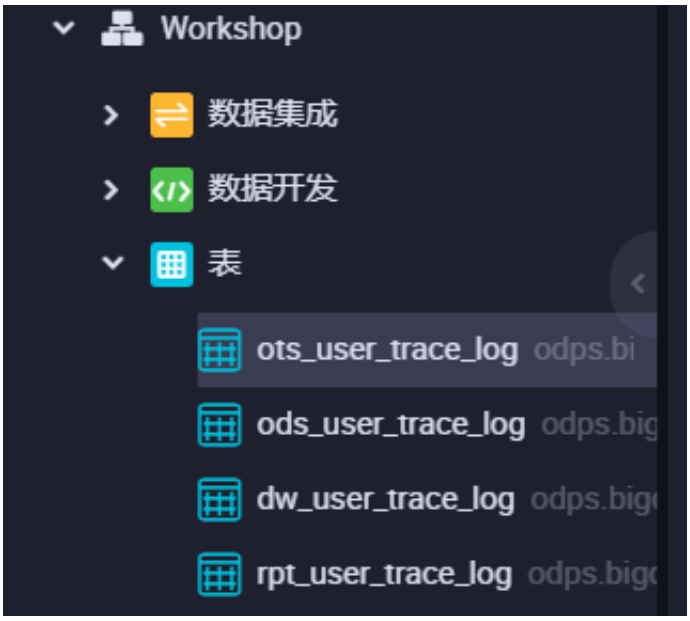
建表方法同上，建表语句如下，完成填写后单击提交到生产环境。rpt_user_trace_log为ADS层表，相关数仓模型定义请参见数仓分层。

```
CREATE TABLE IF NOT EXISTS rpt_user_trace_log (
    country STRING COMMENT '国家',
    province STRING COMMENT '省份',
    city STRING COMMENT '城市',
    device_brand string comment '设备品牌',
    device STRING COMMENT '终端型号',
    system_type STRING COMMENT '系统类型，Android、IOS、ipad、
Windows_phone',
    customize_event STRING COMMENT '自定义事件：登录/退出/购买/注册/点
击/后台/切换用户/浏览',
    use_time BIGINT COMMENT 'APP单次使用时长，当事件为退出、后台、切换用
户时有该项',
    customize_event_content STRING COMMENT '用户关注内容信息，在
customize_event为浏览和评论时 包含该列',
    pv bigint comment '浏览量',
    uv bigint comment '独立访客'
)
PARTITIONED BY (
    dt STRING
```

```
);
```

3. 验证建表结果

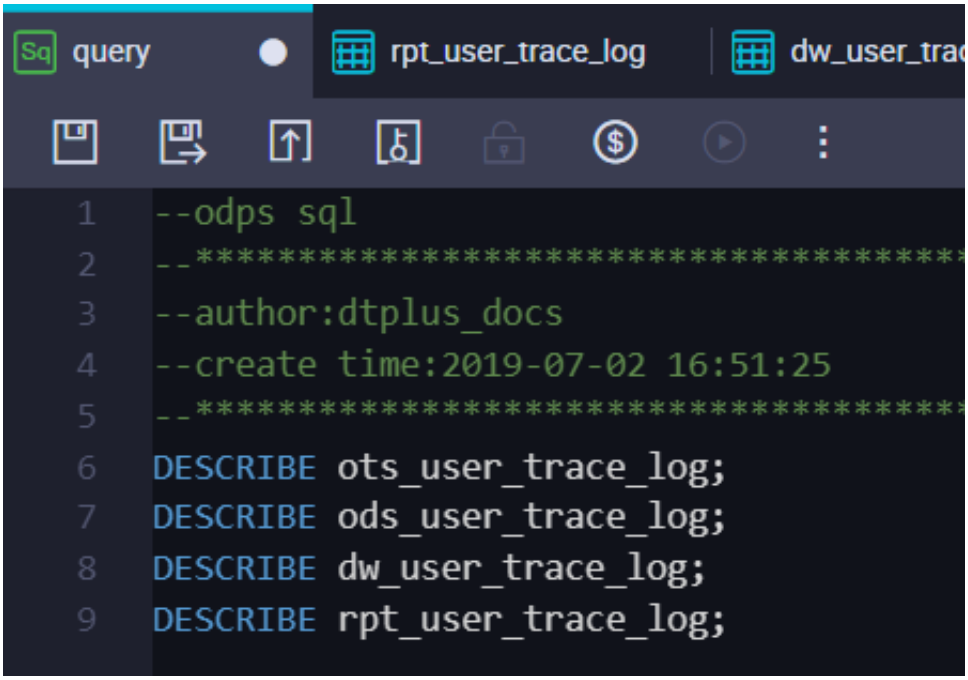
完成建表后，您可以在自己的工作量下看到新建的4张表。



使用数据开发 > 新建数据开发节点 > ODPS SQL，在新建的ODPS SQL节点中写入下列表查询SQL语句。

```
DESCRIBE ots_user_trace_log;
DESCRIBE ods_user_trace_log;
DESCRIBE dw_user_trace_log;
DESCRIBE rpt_user_trace_log;
```

单击运行，查询建表结果。

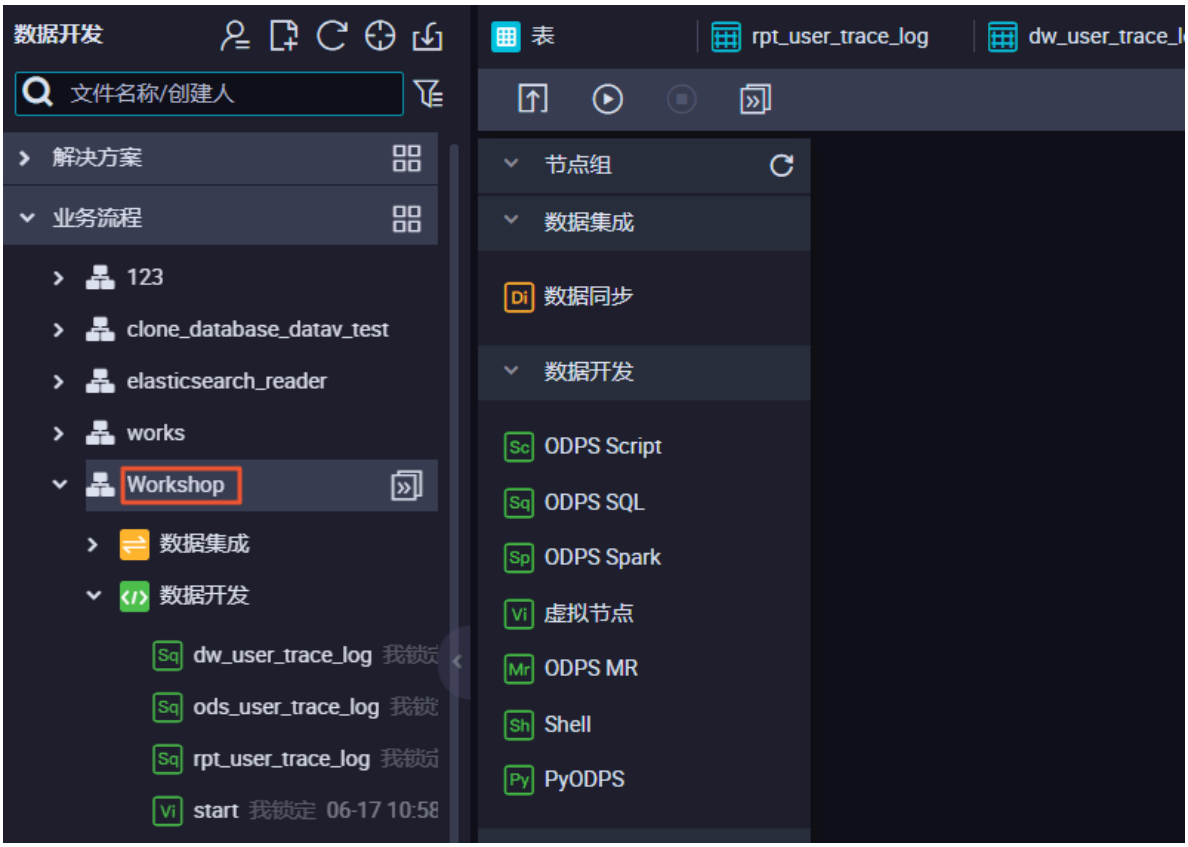


2.4.2 设计工作流

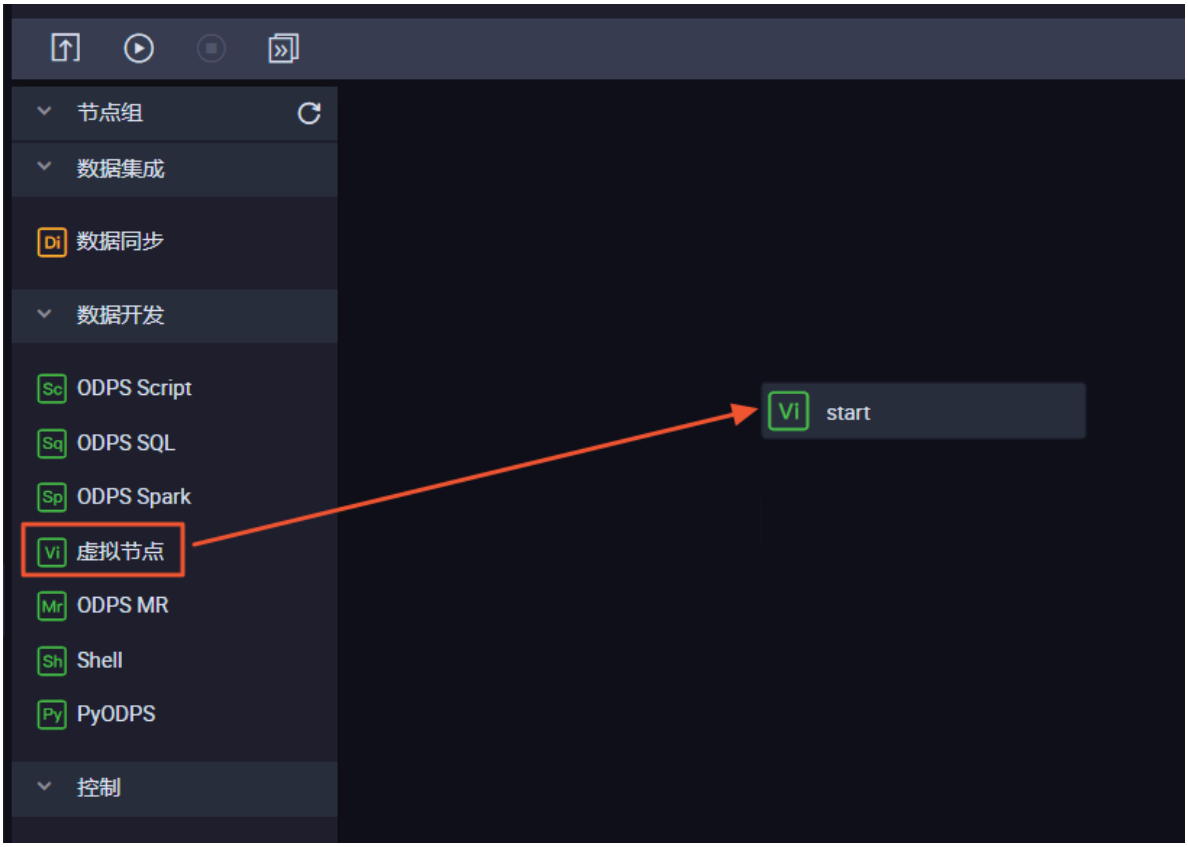
通过设计工作流，您可以明确在整体数据开发过程中各任务节点的排布。对于本教程中这种较为简单的单数据流场景，您可以选择每个数据表（数仓层次）对应一个工作流。

操作步骤

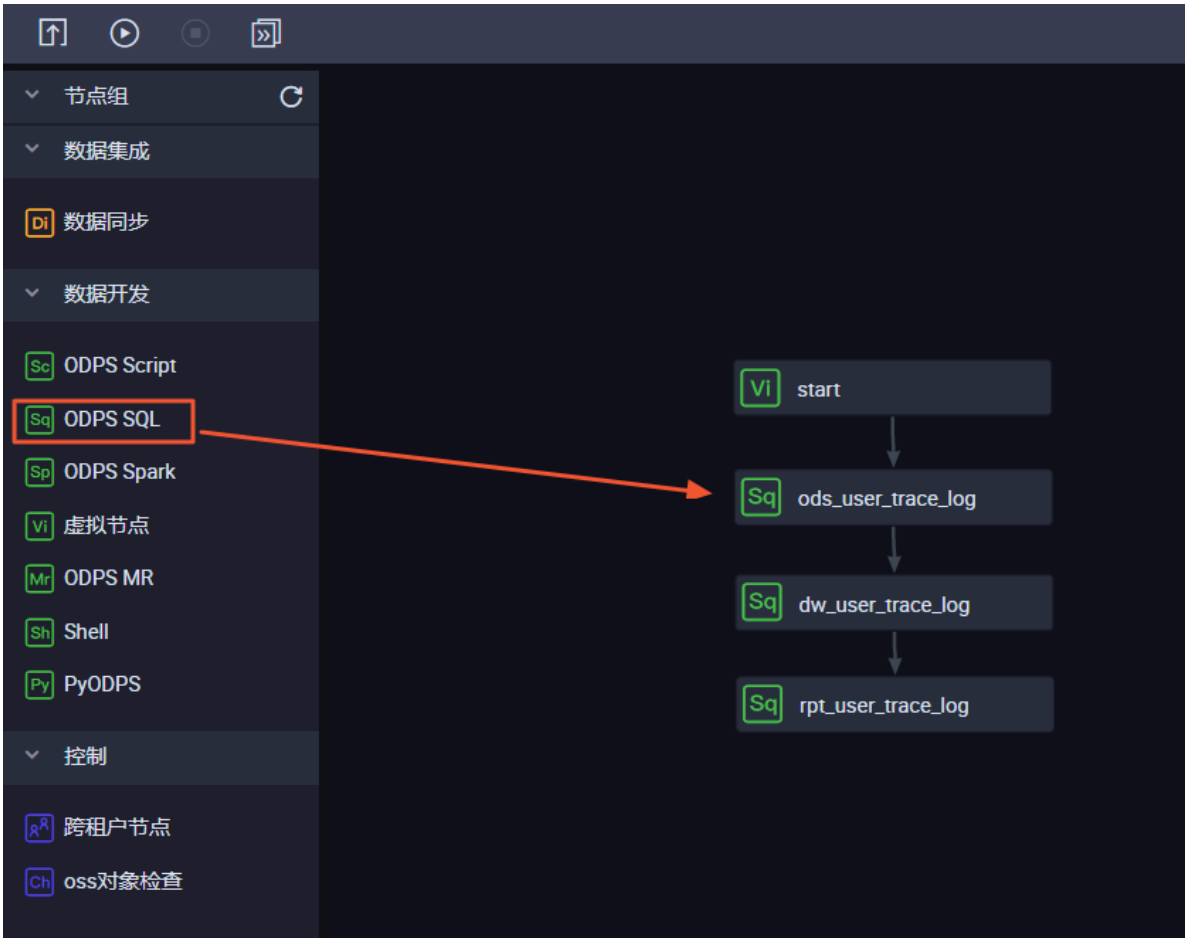
1. 双击您的业务流程，打开画布面板。



2. 向画布中拖入1个虚拟节点。



3. 向画布中拖入3个ODPS SQL节点，依次命名为ods_user_trace_log、dw_user_trace_log、rpt_user_trace_log。通过连接不同节点，配置依赖关系如下。



说明：

ods_user_trace_log、dw_user_trace_log、rpt_user_trace_log分别代表数据仓库的ODS、CDM和ADS层，详情请参见数仓分层。

2.4.3 节点配置

完成工作流设计后，您需要对每个数据开发节点进行配置，填写SQL处理语句。

前提条件

由于本次数据开发过程中需要使用UDF自定义函数，您首先需要完成自定义函数的注册。

操作步骤

1. 添加资源并创建自定义函数

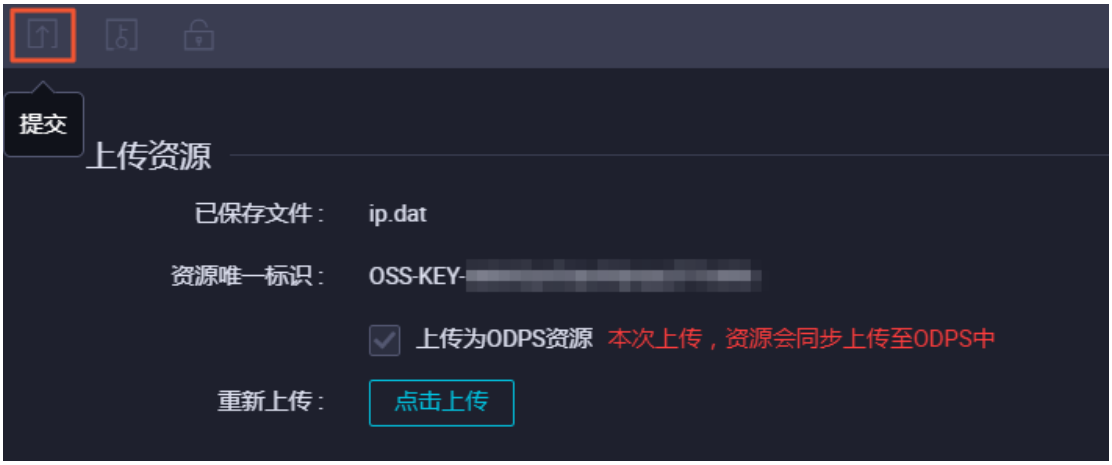
   a) 单击此处，下载用于IP地转换的自定义函数Java包getaddr.jar以及地址库ip.dat。

   本教程不关注IP地址转换的自定义函数内容。如果您有兴趣了解，请参见MaxCompute中实现IP地址归属地转换。

   b) 右键单击您的业务流程下的资源，单击新建资源。

   · File类型对应地址库ip.dat。您需要勾选大文件（内容超过500KB）及上传为ODPS资源，然后点击上传。

上传完成后，请务必记得单击提交。

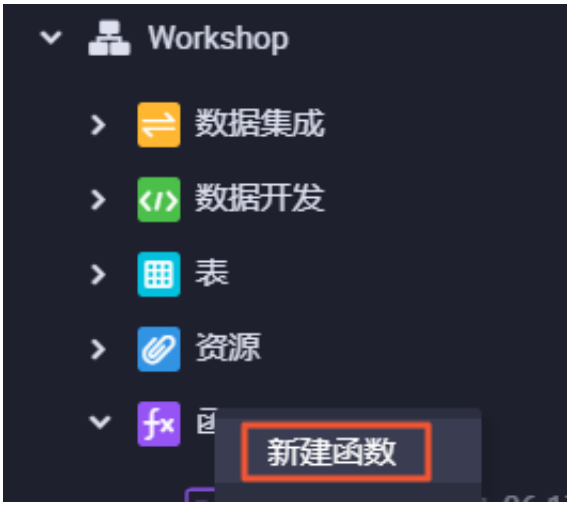


- JAR类型对应Java包getaddr.jar。您需要勾选上传为ODPS资源，然后点击上传。

上传完成后，请务必记得单击提交。

c) 注册函数。

在您的业务流程下右键单击函数，选择新建函数。



请依次填写函数名为getregion，类名为odps.test.GetAddr，资源列表为getaddr.jar,ip.dat，命令格式为getregion(ip string)。填写完成后，单击提交。

2. 配置ODPS SQL节点

a) 双击ods_user_trace_log节点，进入节点配置界面，编写处理逻辑。

SQL代码如下。

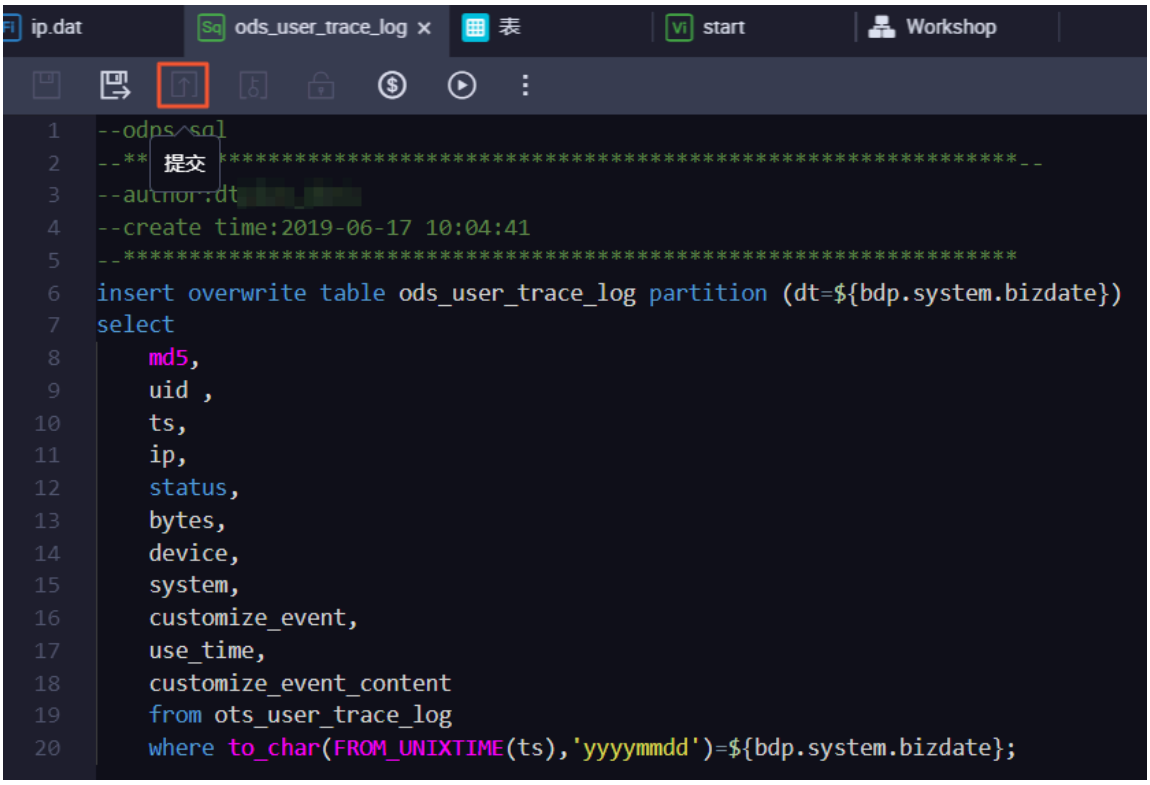
```
insert overwrite table ods_user_trace_log partition (dt=${bdp.
system.bizdate})
select
    md5,
    uid ,
    ts,
    ip,
    status,
    bytes,
    device,
    system,
    customize_event,
```

```
    use_time,
    customize_event_content
    from ots_user_trace_log
    where to_char(FROM_UNIXTIME(ts),'yyyymmdd')=${bdp.system.
bizdate};
```

说明:

关于${bdp.system.bizdate}释义请参见参数配置。

b) 完成代码编写后，单击提交。



3. 配置dw_user_trace_log节点

您可以使用与ods_user_trace_log节点一样的方法配置dw_user_trace_log节点，SQL代码如下。

```
INSERT OVERWRITE TABLE dw_user_trace_log PARTITION (dt=${bdp.system.
bizdate})
SELECT uid, getregion(ip) AS region
    , CASE
        WHEN TOLOWER(device) RLIKE 'xiaomi' THEN 'xiaomi'
        WHEN TOLOWER(device) RLIKE 'meizu' THEN 'meizu'
        WHEN TOLOWER(device) RLIKE 'huawei' THEN 'huawei'
        WHEN TOLOWER(device) RLIKE 'iphone' THEN 'iphone'
        WHEN TOLOWER(device) RLIKE 'vivo' THEN 'vivo'
        WHEN TOLOWER(device) RLIKE 'honor' THEN 'honor'
        WHEN TOLOWER(device) RLIKE 'samsung' THEN 'samsung'
        WHEN TOLOWER(device) RLIKE 'leeco' THEN 'leeco'
        WHEN TOLOWER(device) RLIKE 'ipad' THEN 'ipad'
        ELSE 'unknown'
    END AS device_brand, device
    , CASE
        WHEN TOLOWER(system) RLIKE 'android' THEN 'android'
```

```
        WHEN TOLOWER(system) RLIKE 'ios' THEN 'ios'
        ELSE 'unknown'
    END AS system_type, customize_event, use_time, customize_
event_content
FROM ods_user_trace_log
WHERE dt = ${bdp.system.bizdate};
```

4. **配置rpt_user_trace_log节点**

   您可以使用与ods_user_trace_log节点一样的方法配置rpt_user_trace_log节点，SQL代码如下。

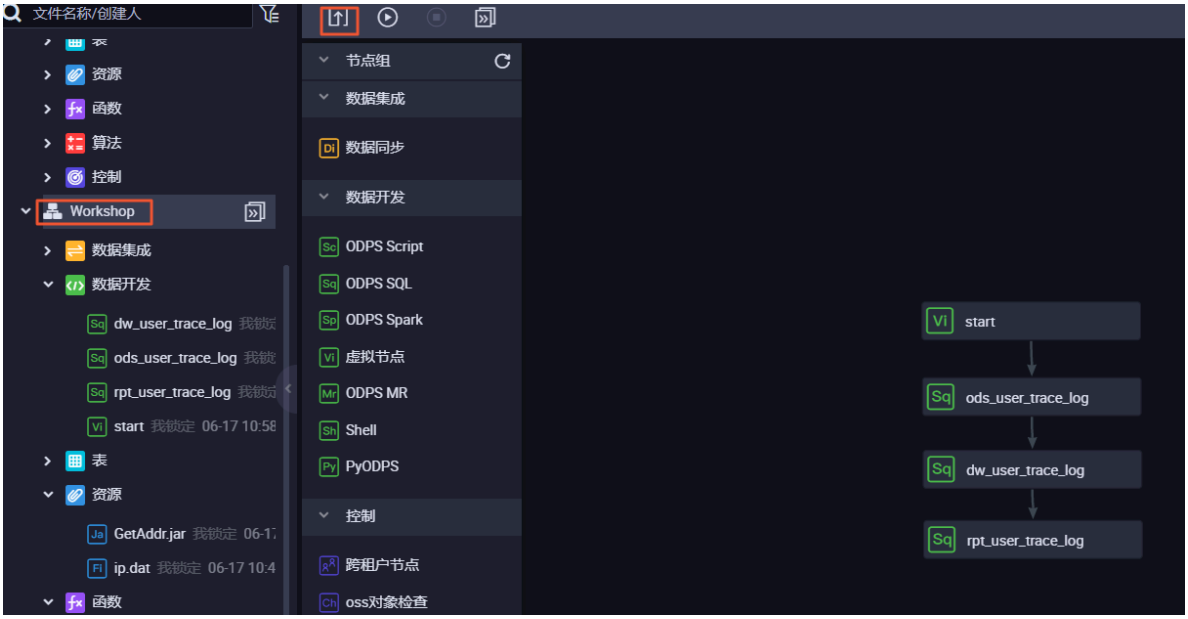
```
INSERT OVERWRITE TABLE rpt_user_trace_log PARTITION (dt=${bdp.system
.bizdate})
SELECT split_part(split_part(region, ',', 1),'[',2) AS country
    , trim(split_part(region, ',', 2)) AS province
    , trim(split_part(region, ',', 3)) AS city
    , MAX(device_brand), MAX(device)
    , MAX(system_type), MAX(customize_event)
    , FLOOR(AVG(use_time / 60))
    , MAX(customize_event_content), COUNT(uid) AS pv
    , COUNT(DISTINCT uid) AS uv
FROM dw_user_trace_log
WHERE dt = ${bdp.system.bizdate}
GROUP BY uid,
    region;
```

2.4.4 任务提交与测试

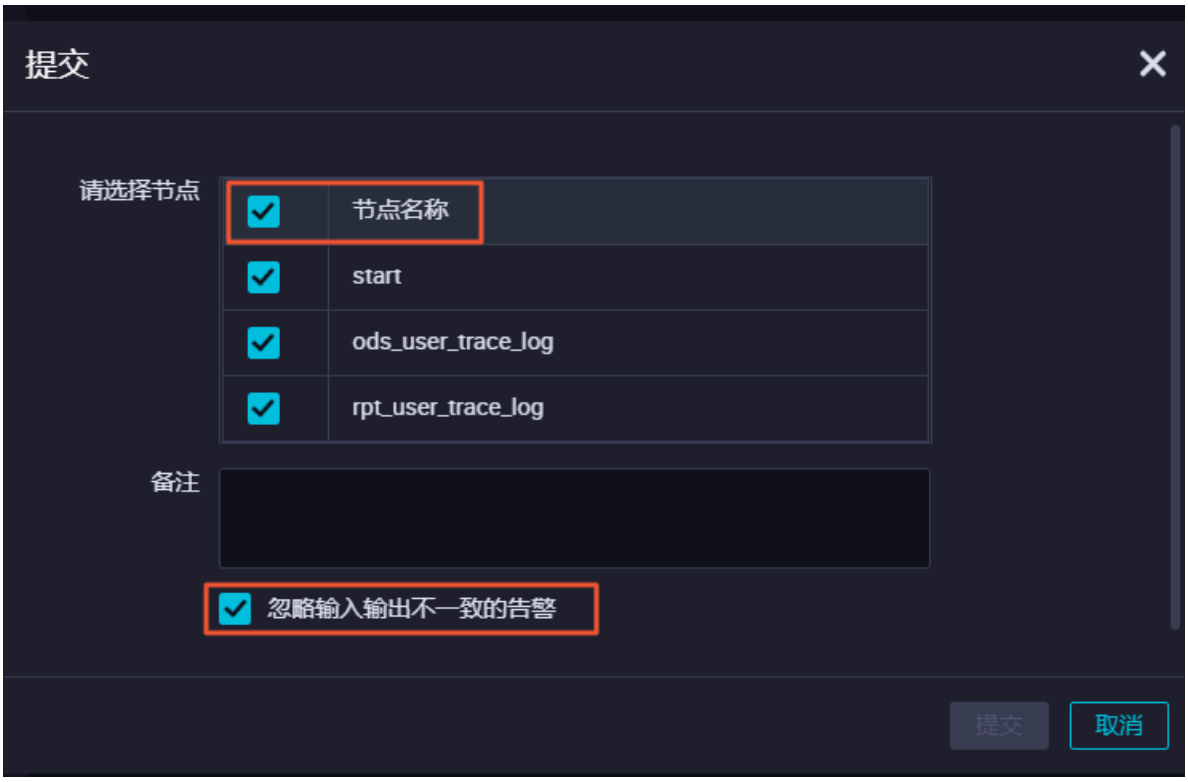
在您完成节点配置后，还需要提交任务到运维中心，才能对任务进行测试。

操作步骤

1. 双击您的业务流程名称，单击提交。

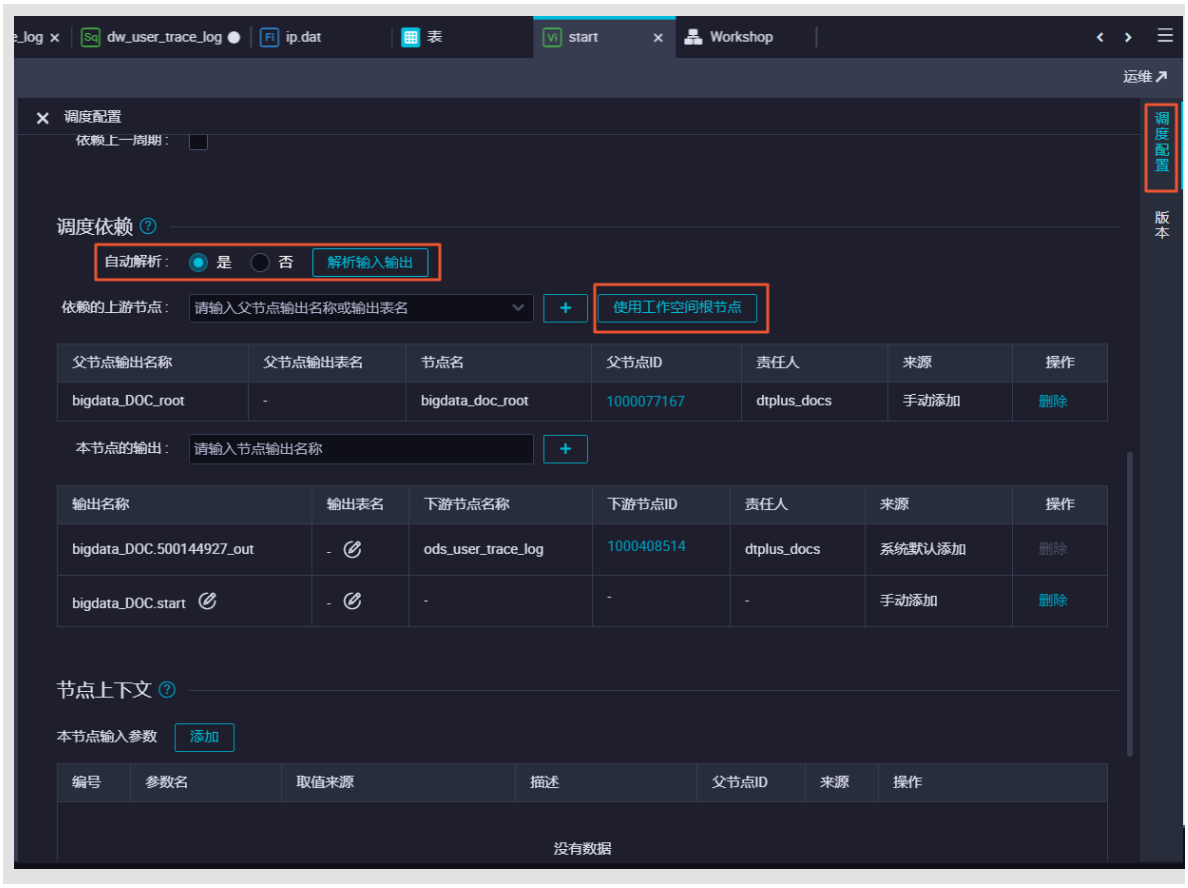


勾选所有可提交节点及忽略输入输出不一致的告警。如果您的节点在配置完成后已经提交完毕且无更新，此处您会发现没有可以提交的节点，直接跳过本步骤。



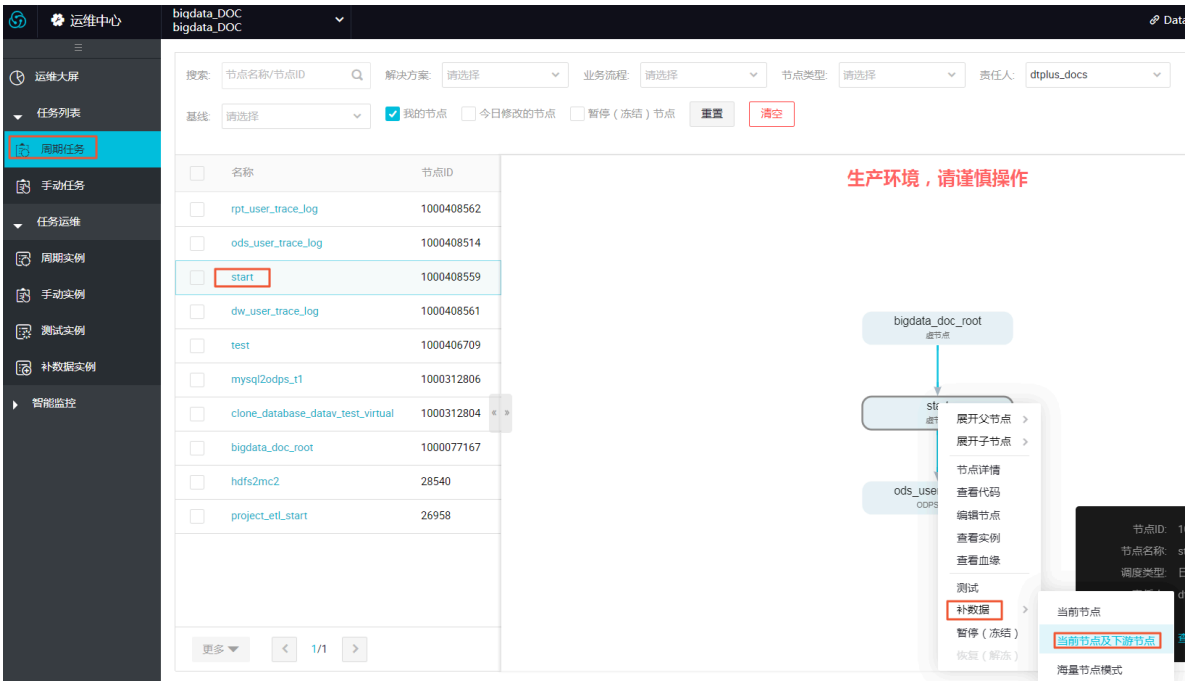
说明:

如果您在提交过程中报错，很可能是节点调度配置对输入输出的自动解析有误。您可以重新编辑节点的调度配置页面，选择自动解析为否后，手动删除错误的输入。对于VI虚拟节点，建议您勾选使用工作空间根节点。

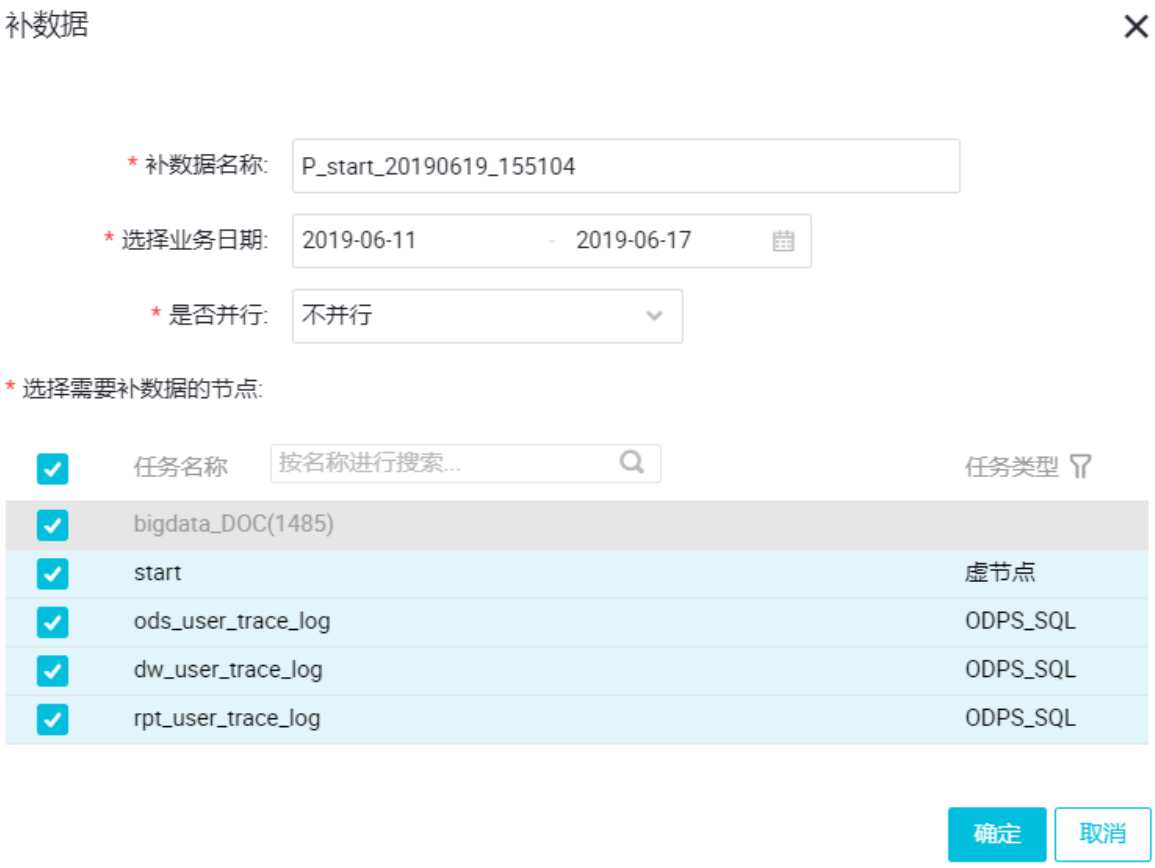


2. 单击右上角的运维中心进行界面切换。

3. 双击任务列表 > 周期任务中您的虚拟节点后，在右侧界面右键单击虚拟节点（本例中名为start）。在弹框中单击补数据 > 当前节点及下游节点。



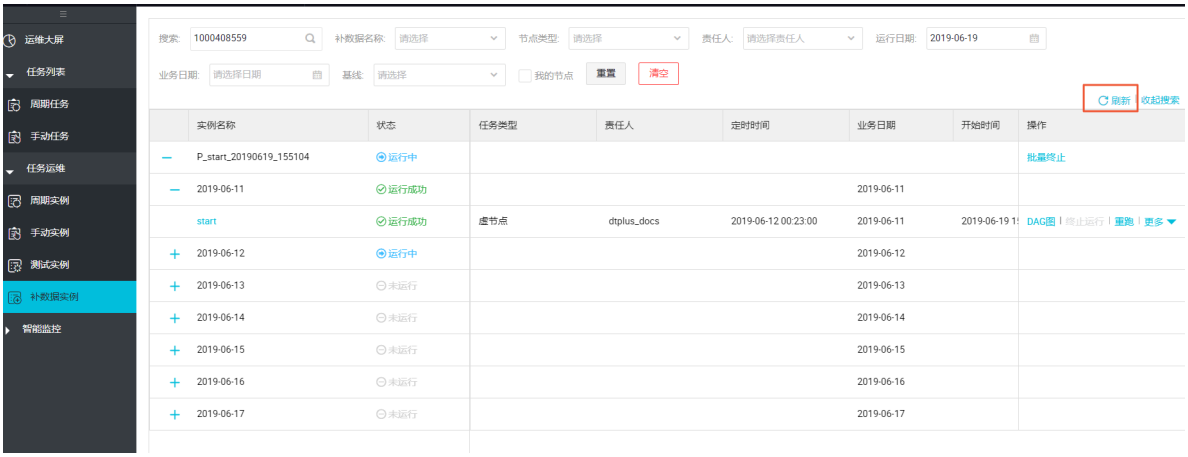
在弹框中勾选所有节点，选择业务日期为最近一周，单击确定。



说明:

关于补数据实例的详情请参见补数据实例。

4. 在补数据实例中，您可以查看补数据实例的运行情况，并通过单击刷新查看实时状态。

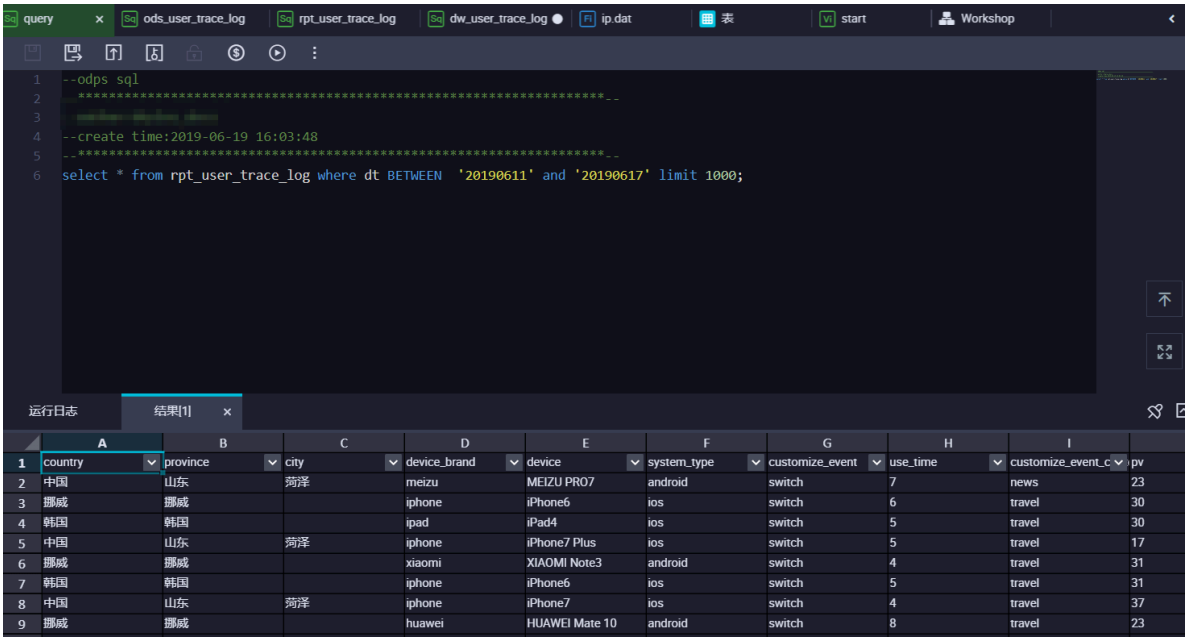


如果运行状态异常，您可以右键单击出错节点，单击查看运行日志进行排查。


生产环境，请谨慎操作
展开父节点
展开子节点
查看运行日志
查看代码
编辑节点
查看血缘
终止运行
重跑
重跑下游
置成功
暂停（冻结）

5. 待补数据实例运行完成后，您可以使用数据开发 > 新建数据开发节点 > ODPS SQL，在新建的ODPS SQL节点中写入下列SQL语句来确认数据是否成功写入rpt_user_trace_log表。

   SQL语句：`select * from rpt_user_trace_log where dt BETWEEN '20190611' and '20190617' limit 1000;`



	A	B	C	D	E	F	G	H	I	
1	country	province	city	device_brand	device	system_type	customize_event	use_time	customize_event_c	pv
2	中国	山东	菏泽	meizu	MEIZU PRO7	android	switch	7	news	23
3	挪威	挪威		iphone	iPhone6	ios	switch	6	travel	30
4	韩国	韩国		ipad	iPad4	ios	switch	5	travel	30
5	中国	山东	菏泽	iphone	iPhone7 Plus	ios	switch	5	travel	17
6	挪威	挪威		xiaomi	XIAOMI Note3	android	switch	4	travel	31
7	韩国	韩国		iphone	iPhone6	ios	switch	5	travel	31
8	中国	山东	菏泽	iphone	iPhone7	ios	switch	4	travel	37
9	挪威	挪威		huawei	HUAWEI Mate 10	android	switch	8	travel	23

2.5 数据可视化展现

数据表rpt_user_trace_log加工完成后，您可以通过Quick BI创建网站用户分析画像的仪表板，实现该数据表的可视化。

前提条件

在开始试验前，请确认您已经完成了环境准备和数据建模与开发的全部步骤。单击进入Quick BI控制台。

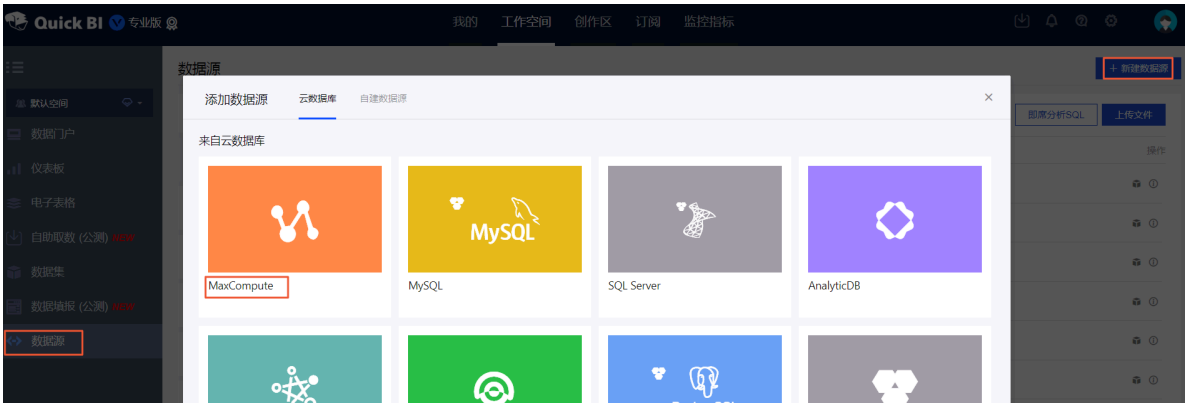
背景信息

rpt_user_trace_log表包含了country、province、city、device_brand、use_time、pv等字段信息。您可以通过仪表板展示用户的核心指标、周期变化、用户地区分布、分布和记录。

操作步骤

1. 单击进入默认空间，您也可以使用自己的个人空间。

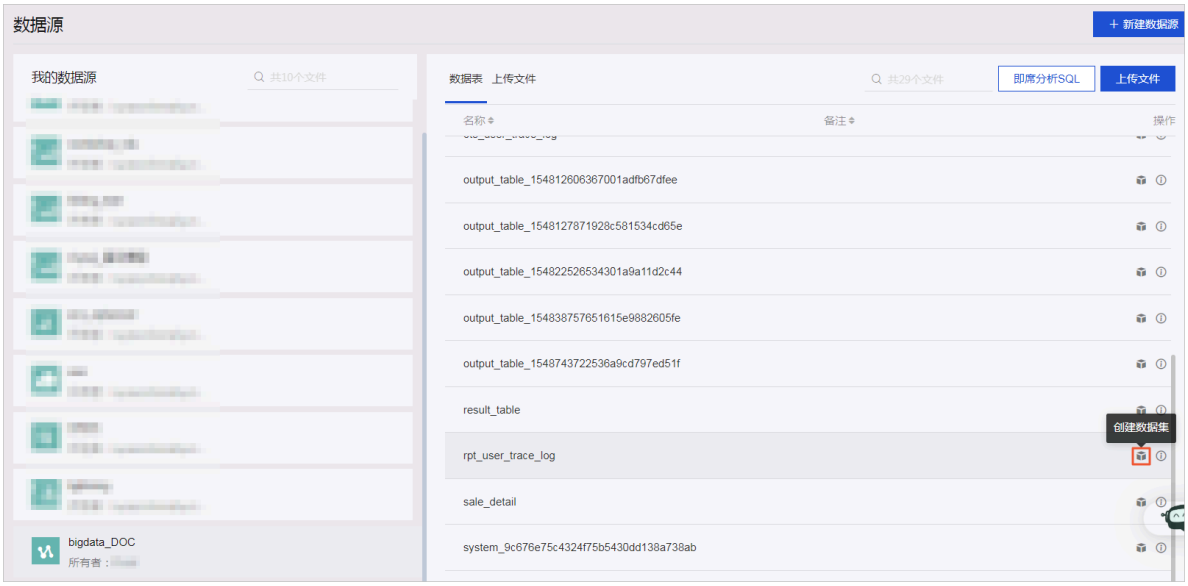
2. 选择数据源 > 新建数据源 > 云数据库 > MaxCompute。



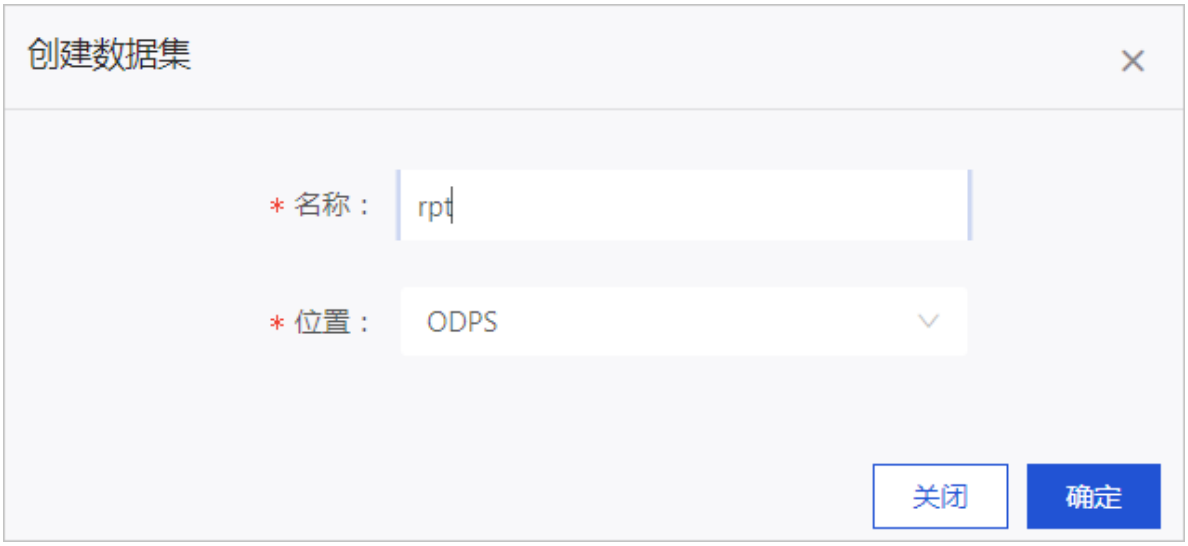
3. 输入您的MaxCompute项目名称以及您的AcessKey信息，数据库地址使用默认地址即可，关于数据库地址详情请参见配置Endpoint。

   完成填写后，单击连接测试，待显示数据源连通性正常后单击添加即可。

4. 找到您刚添加的数据源的rpt_user_trace_log表，单击创建数据集。



选择您想放置的数据集位置，单击确定。



5. 进入数据集列表页，单击您刚刚创建的数据集，对数据集进行编辑。

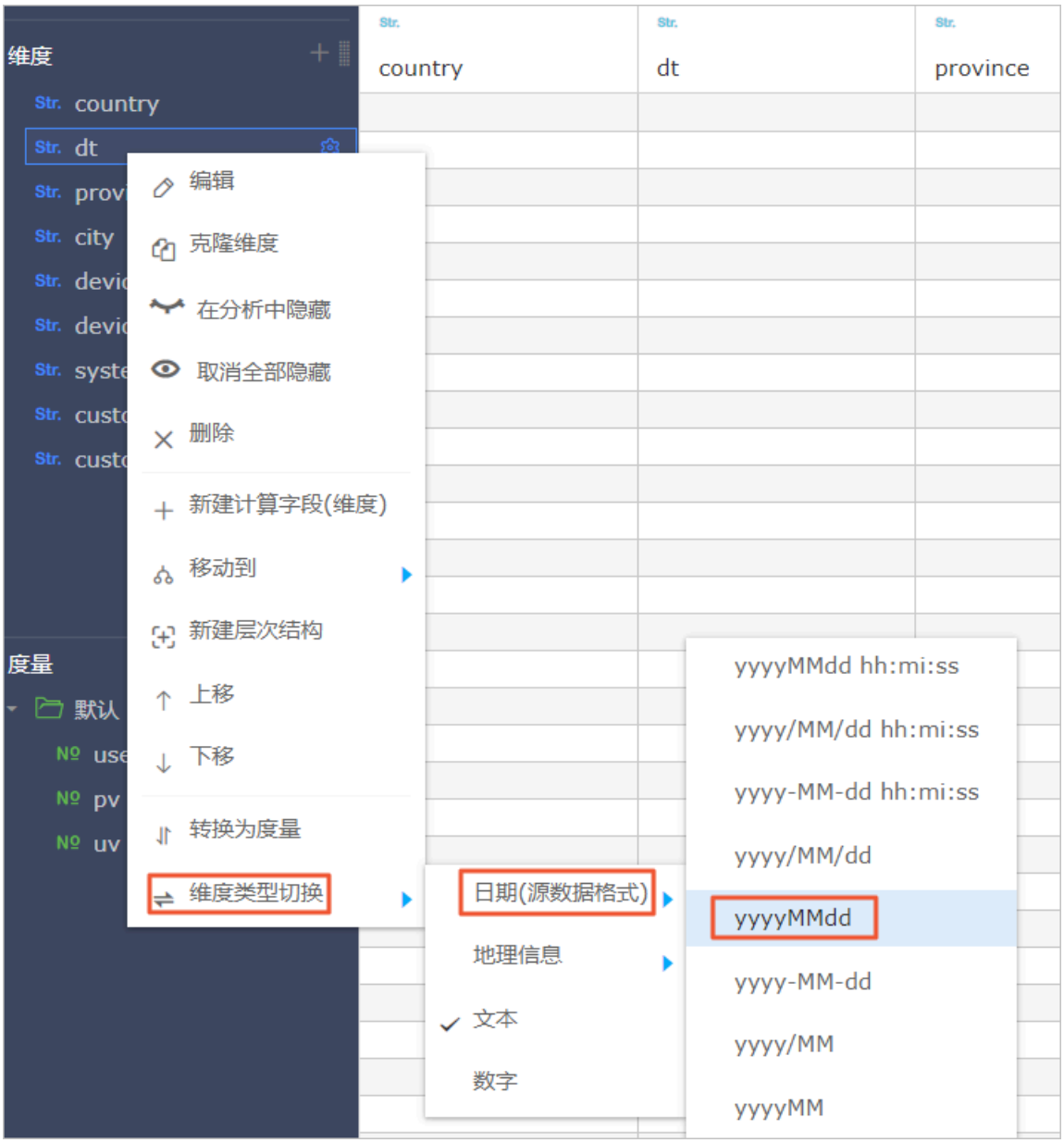
常见的数据集加工包括：维度、度量的切换、修改维度的类型、增加计算字段、创建层次结构、修改字段的数据类型、更改度量聚合方式、制作关联模型。

6. 进入数据集列表页，单击您刚刚创建的数据集。

7. 转换字段的维度类型

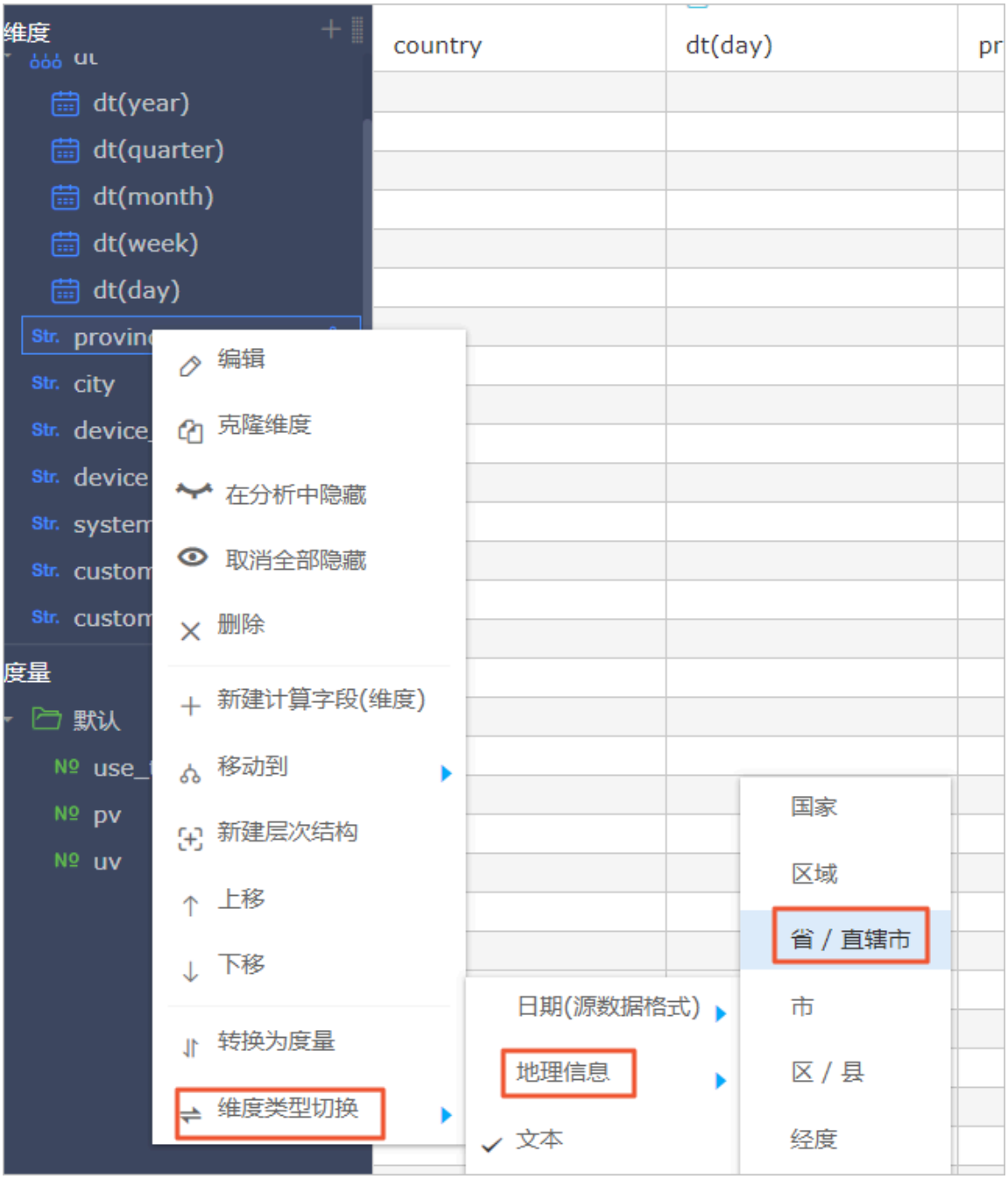
a) 转换日期字段的维度类型。

右键单击dt字段，选择维度类型切换 > 日期（源数据格式） > yyyyMMdd。

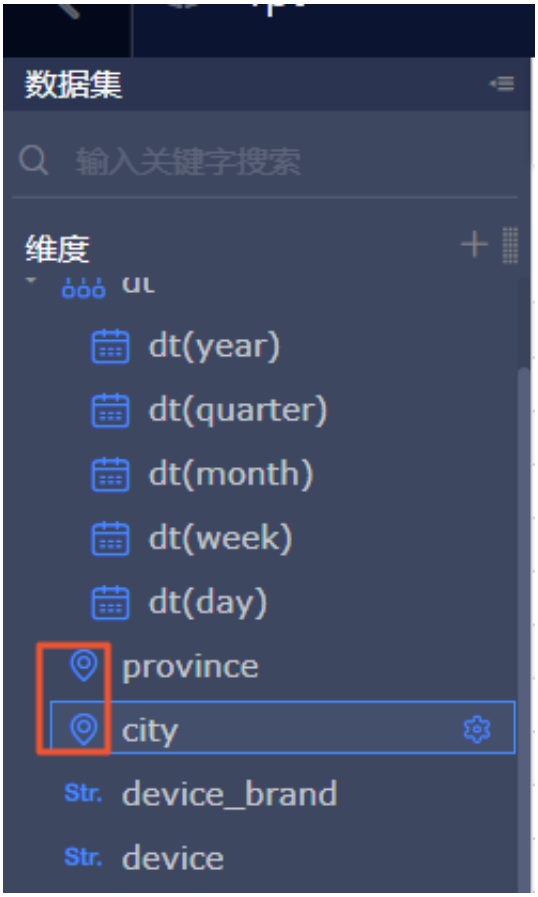


b) 转换地理信息字段的维度类型。

右键单击province字段，选择维度类型切换 > 地理信息 > 省/直辖市。

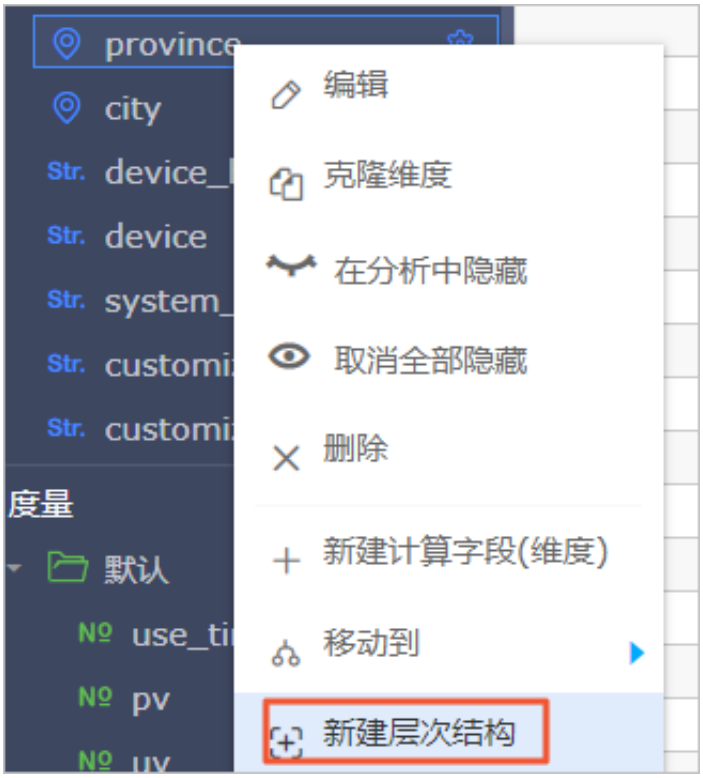


右键单击city字段，选择维度类型切换 > 地理信息 > 市。转换成功后，在左侧维度栏中会看到字段前多一个地理位置图标。



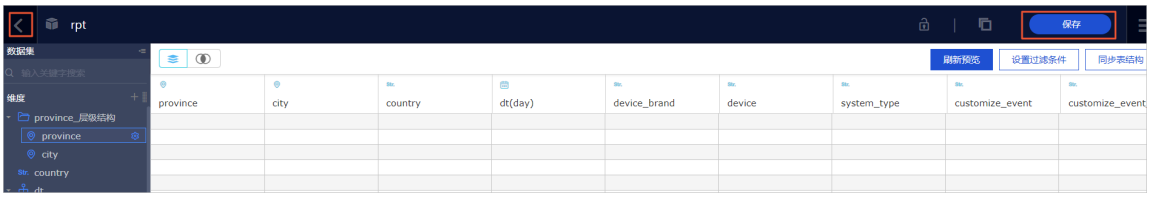
c) **新建层次结构。**

右键province，单击新建层次结构，在弹框中单击确定。



然后把city字段移到province层次结构的树下。

完成上述操作后，单击保存，返回数据集列表。



8. 制作仪表板。

随着数据的更新，让报表可视化地展现最新数据，这个过程叫制作仪表板。仪表板的制作流程如下：

a. 确定内容。

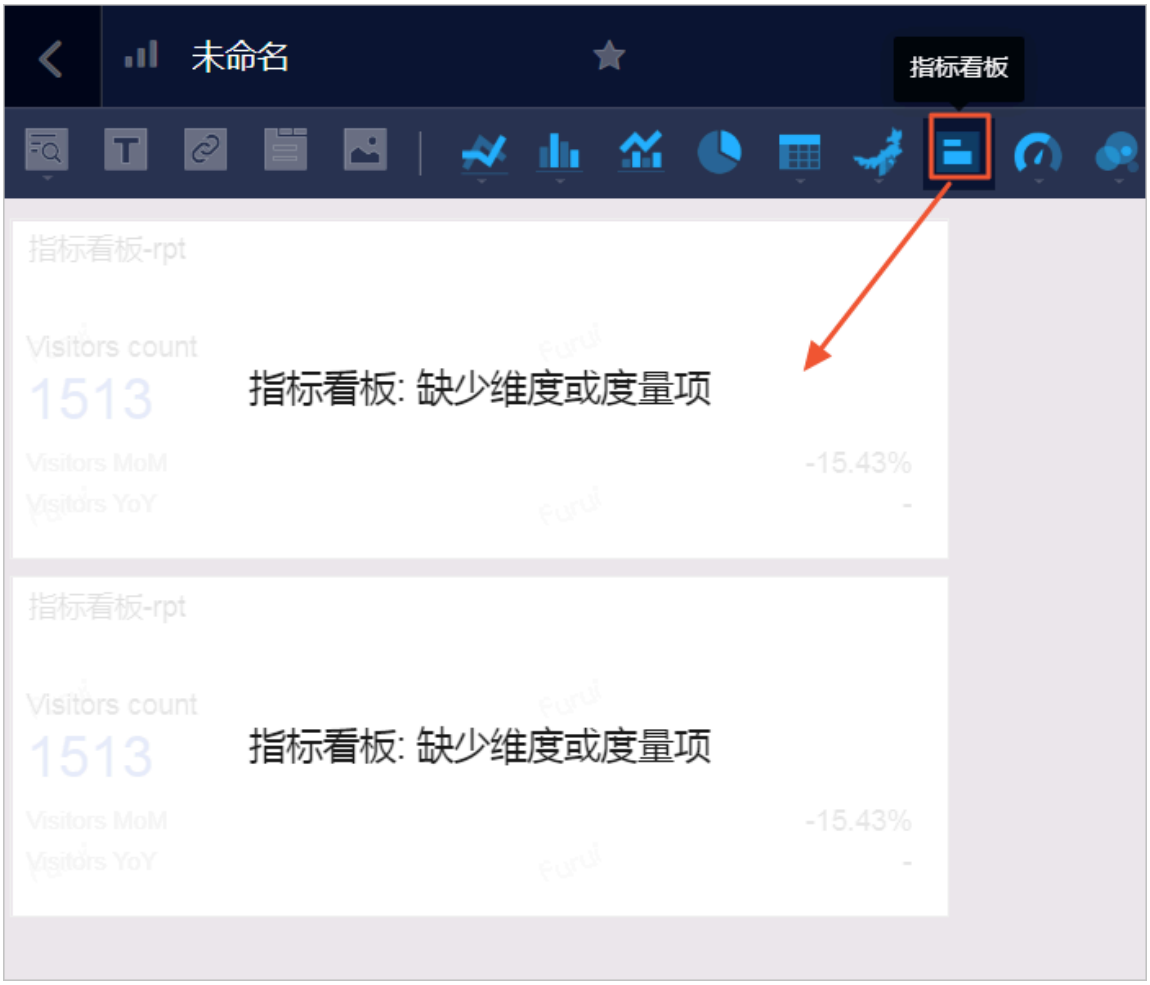
b. 确定布局和样式。

c. 制作图表。

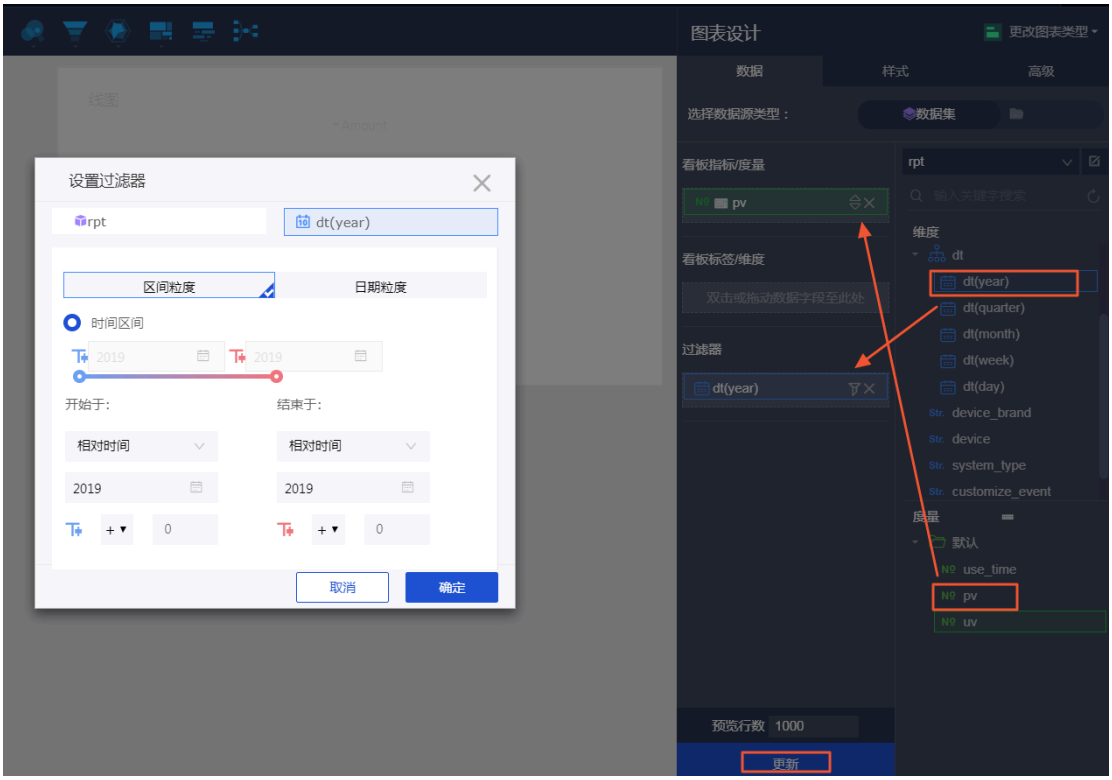
d. 实现动态联动查询。

a) 单击rpt数据集后的新建仪表板，选择常规模式，进入仪表板编辑页。

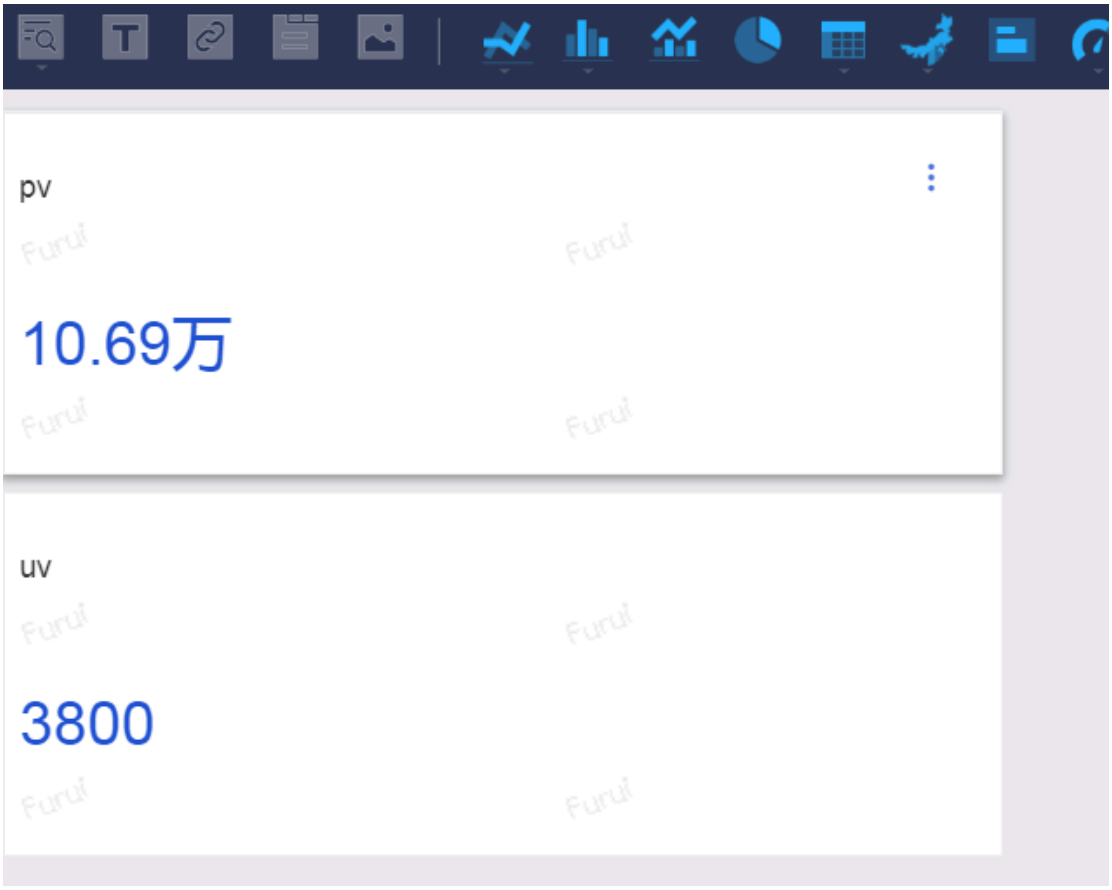
b) 从仪表板空间中向空白区拖入2个指标看板，调整布局成一排。



- 指标看板一：选择数据来源为数据集rpt，选择度量为pv。由于数据表rpt_user_trace_log为分区表，因此必须在过滤器处选择筛选的日期，本例中筛选为2019~2019年，完成设置后单击更新。



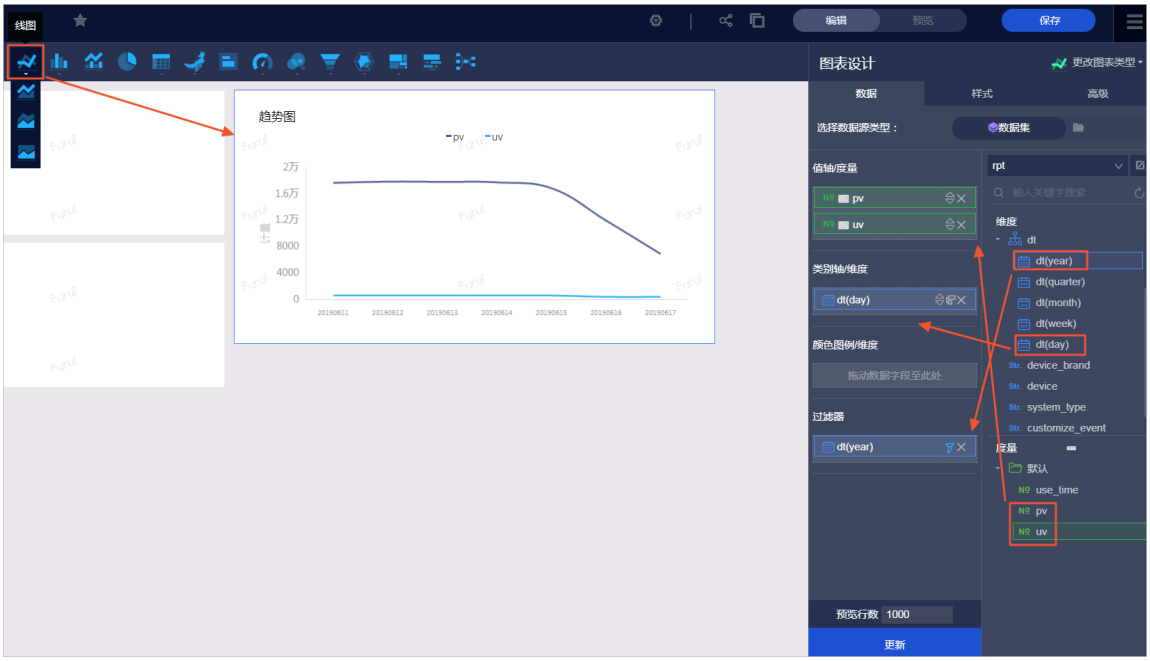
- 指标看板二：选择数据来源为来自数据集rpt，选择度量为uv，其他操作同上。完成设置后单击更新样式处设置指标看板显示的名称，显示效果如下。



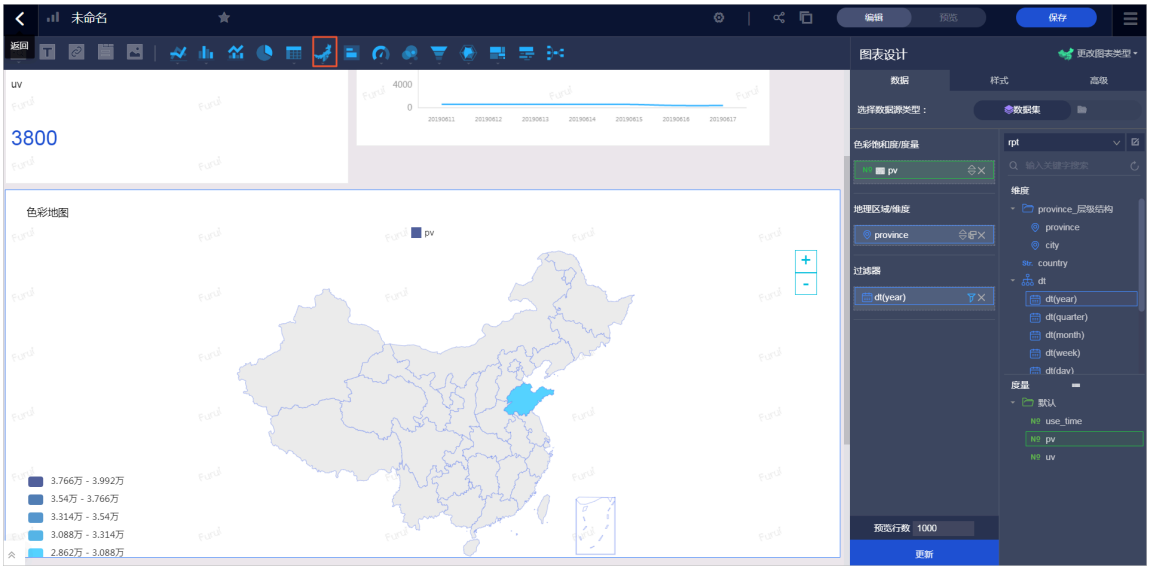
c) 制作趋势图：将图表区域内的线图拖拽到左侧画布。

参数配置如下，完成之后单击更新：

- 值轴/度量：pv、uv
- 类别轴/维度：dt（day）
- 过滤器：dt（year）



d) 制作色彩地图：单击图表区域内的色彩地图，并选择数据源来源为数据集rpt_user_trace_log，选择地理区域/维度为province（地区）、色彩饱和度/度量为pv，选择完成后单击更新，结果如下。



e) 完成配置后，单击保存及预览，即可看到展示效果。


RPT
预览
保存
pv
10.69万
uv
3800
趋势图
pv
uv
20190611
20190612
20190613
20190614
20190615
20190616
20190617
色彩地图
pv

3 数据质量保障教程

背景信息

本教程基于一份真实的网站日志数据集，数据来源于某网站上的HTTP访问日志数据。基于这份网站日志，您可以实现如下分析需求：统计并展现网站的浏览次数（PV）和独立访客（UV），并能够按照用户的终端类型（如Android、iPad、iPhone、PC等）和地域分别统计。

在整体数据链路的处理过程中，为保证最终产出数据的质量，您需要对数据仓库的ODS、CDM和ADS层的数据分别进行监控。数据仓库分层的定义请参见数仓分层。本教程基于教程《搭建互联网在线运营分析平台》，ods_user_trace_log、dw_user_trace_log、rpt_user_trace_log分别代表数据仓库的ODS、CDM和ADS层，详情请参见设计工作流。

如何衡量数据质量

3.1 数据质量教程概述

数据质量是数据分析结论有效性和准确性的基础。本文为您介绍数据质量保障教程的业务场景以及如何衡量数据质量的高低。

前提条件

在开始本教程前，请您首先完成教程《搭建互联网在线运营分析平台》，详情请参见业务场景与开发流程。

说明:

由于数据质量当前仅在华东2区域开放，请您在华东2区域创建DataWorks工作空间，完成教程。

业务场景

要保证业务数据质量，首先您需要明确数据的消费场景和加工链路。

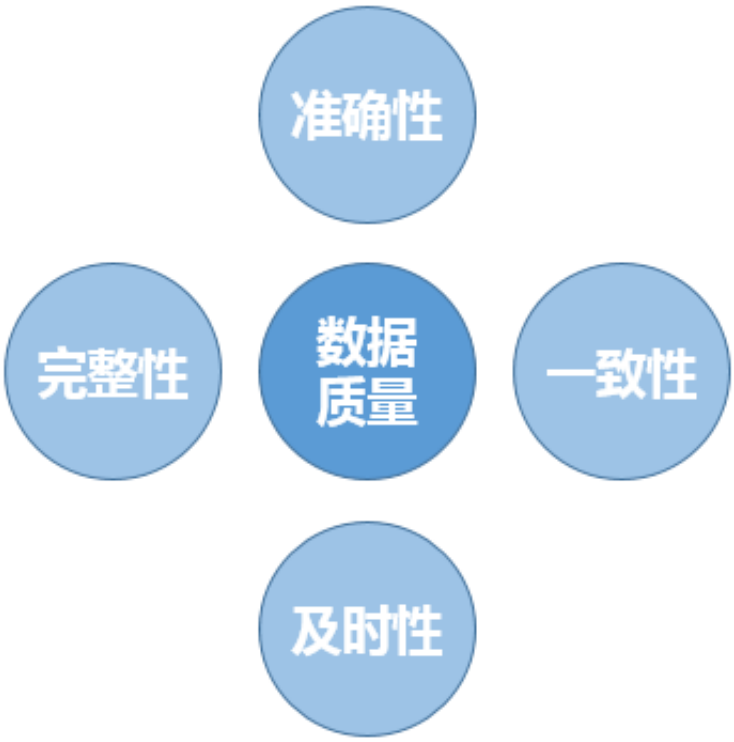
本教程基于一份真实的网站日志数据集，数据来源于某网站上的HTTP访问日志数据。基于这份网站日志，您可以实现如下分析需求：统计并展现网站的浏览次数（PV）和独立访客（UV），并能够按照用户的终端类型（如Android、iPad、iPhone、PC等）和地域分别统计。

在整体数据链路的处理过程中，为保证最终产出数据的质量，您需要对数据仓库的ODS、CDM和ADS层的数据分别进行监控。数据仓库分层的定义请参见数仓分层。本教程基于教程《搭建互联网在线运营分析平

台》，ods_user_trace_log、dw_user_trace_log、rpt_user_trace_log分别代表数据仓库的ODS、CDM和ADS层，详情请参见设计工作流。

如何衡量数据质量

数据质量可以从完整性、准确性、一致性和及时性共四个角度进行评估。详情请参见数据质量保障原则。



在本教程中，您将学会通过数据质量风险监控，保证数据的完整性、准确性。通过数据及时性监控，保证数据的及时性。

- 完整性

  完整性是指数据的记录和信息是否完整、不缺失。数据的缺失包括数据记录的缺失（表行数异常）和记录中某字段信息的缺失（字段出现空值）。在本教程中，您需要重点关注数据的生产环节（MaxCompute外部表引用的表格存储数据）和加工环节（数仓CDM及ADS层）中表行数是否大于0、表行数波动是否正常以及字段是否出现空值或重复的情况。

- 准确性

  准确性是指数据记录中信息和数据是否准确、不存在错误或异常。例如，在本教程中，如果UV、PV数值小于0，则明显是错误数据。

- 一致性

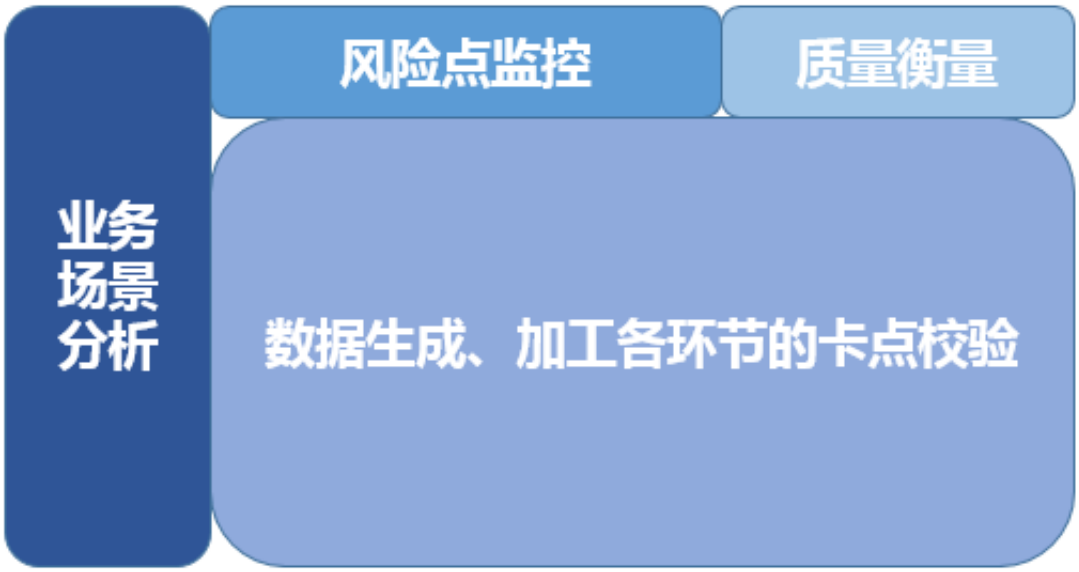
  对于不同的业务流程和节点，同一份数据必须保持一致性。例如表字段的province字段中如果有浙江、ZJ两种表述，在您`group by province`时会出现两条记录。

- 及时性

  及时性主要体现在最终ADS层的数据可以及时产出。为保证及时性，您需要确保整条数据加工链路上的每个环节都可以及时产出数据。本教程将利用DataWorks智能监控功能保证数据加工每个环节的及时性。

3.2 数据质量管理流程

数据质量的管理流程包括业务数据资产定级、加工卡点、风险点监控、及时性监控，您可以构建属于自己的数据质量保障体系。



数据质量的管理流程如下：

1. 数据资产定级。
2. 离线数据加工卡点。
3. 数据质量风险监控。
4. 数据及时性监控。

3.3 数据资产定级

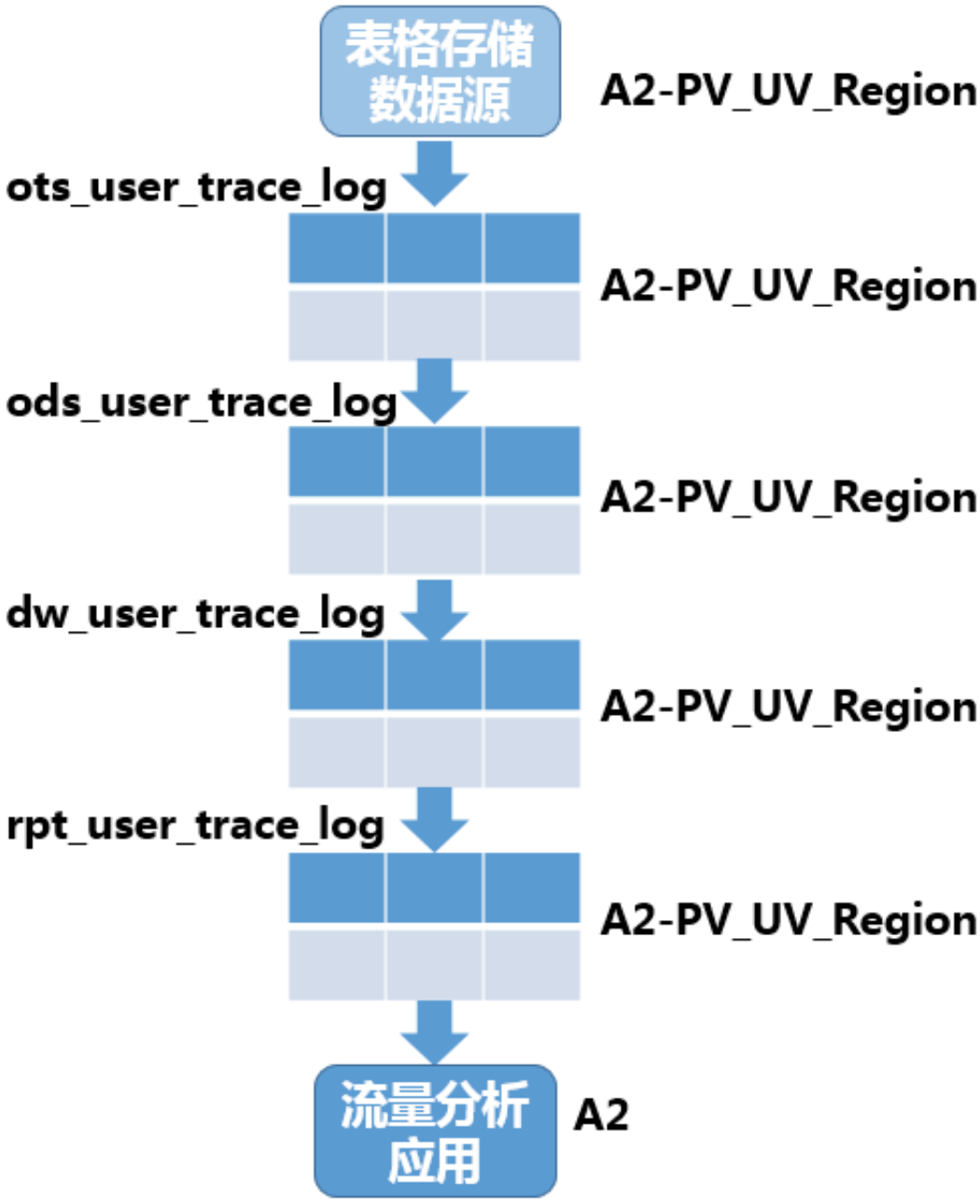
数据的资产等级，可以根据数据质量不满足完整性、准确性、一致性、及时性后对业务的影响程度进行划分。

数据等级定义如下：

- 毁灭性质：数据一旦出错，将会引起重大资产损失，面临重大收益损失等。
- 全局性质：数据直接或间接用于企业级业务和效果评估、重要决策等。
- 局部性质：数据直接或间接用于某些业务线的运营、报告等，若出现问题会给业务线造成一定的影响或造成工作效率降低。
- 一般性质：数据主要用于日常数据分析，出现问题带来的影响极小。
- 未知性质：无法明确数据的应用场景。

资产等级可以用Asset进行标记：毁灭性质为A1，全局性质为A2，局部性质为A3，一般性质为A4，未知性质为Ax。重要程度为：A1>A2>A3>A4>Ax。

在数据流转链路上，您需要整理各个表是被哪些应用业务消费。通过给这些应用业务划分数据资产等级，结合数据的上下游依赖关系，将整个链路打上某一类资产等级的标签。在本教程中，互联网在线运营分析平台只存在一个应用：统计并展现网站的PV和UV，并能够按照用户的终端类型和地域进行统计，命名为PV_UV_Region。假设该应用会直接影响整个企业的重要业务决策，您可以定级应用为A2，从而整个数据链路上的表的数据等级，都可以标记为A2-PV_UV_Region。



说明:

当前MaxCompute暂无配套资产等级打标工具，您可以使用其他工具完成打标。

3.4 离线数据加工卡点

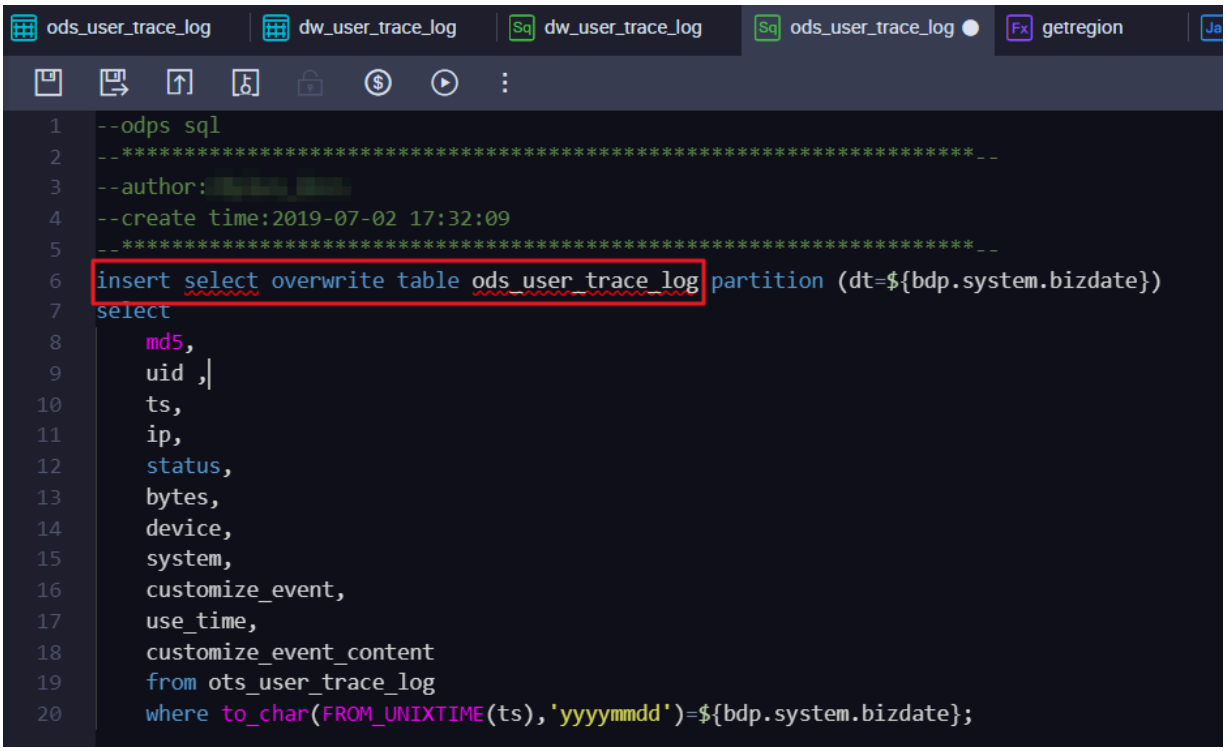
离线数据加工卡点，主要指在业务系统的数据生成过程中进行的卡点校验。

代码提交的卡点校验

代码提交卡点主要包括您在提交代码时，手动或自动进行SQL扫描，检查您的SQL逻辑。校验规则分类如下：

- 代码规范类规则，如表命名规范、生命周期设置、表注释等。
- 代码质量类规则，如分母为0提醒、NULL值参与计算影响结果提醒、插入字段顺序错误等。
- 代码性能类规则，如分区裁剪失效、扫描大表提醒、重复计算检测等。

您在使用DataWorks数据开发功能时，如果代码中有语法错误，会出现如下红色波浪线提示。



关于SQL代码、表命名、生命周期、注释的其他规范，请参见表设计规范及SQL代码编码原则与规范。

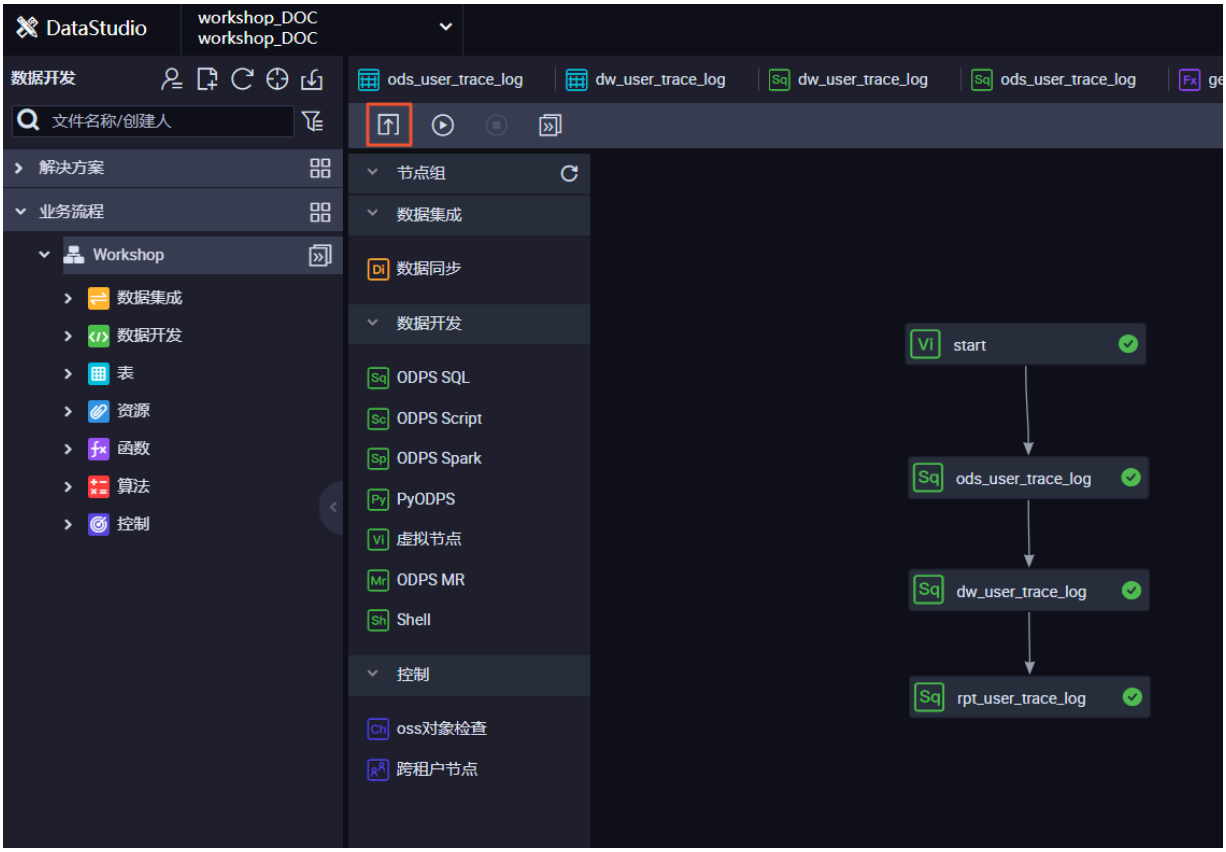
任务发布卡点

为保证线上数据的准确性，每次变更都需要经过测试再发布到线上生产环境，且生产环境测试通过后才算发布成功。发布上线前的测试包括代码审查和回归测试。对于资产等级较高的应用，必须在完成回归测试之后，才允许任务发布，本教程中应用为A2等级，属于高资产级别应用。

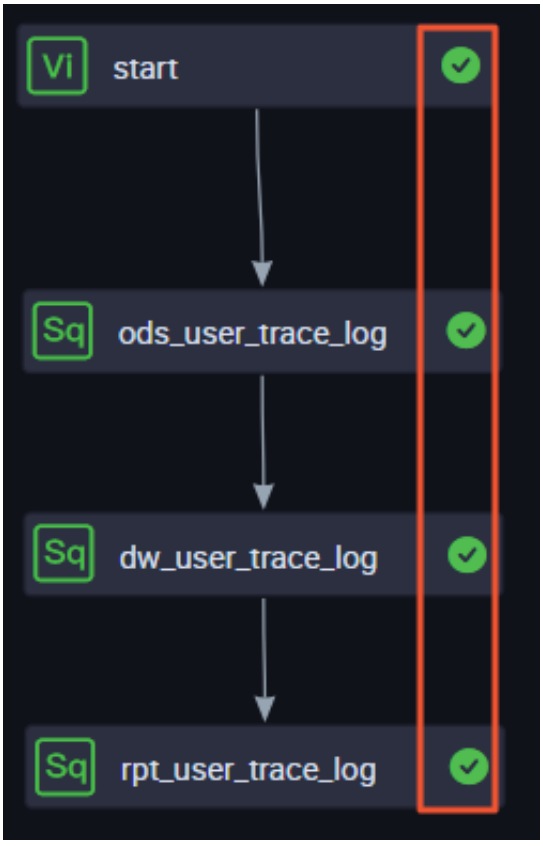
回归测试需保证您能充分模拟真实环境进行测试：

- 对于标准模式项目，您可使用SQL语句将数据从生产环境复制开发环境，运行业务流程。
- 对于简单模式的项目，您可以直接运行业务流程，观察是否有报错，详情请参见简单模式和标准模式的区别。

在本教程中，由于使用简单模式，您只需提交任务。



完成运行后，如果所有节点都显示绿色图标，则表示业务流程测试通过。



相关人员通告

在进行更新操作前，需要通知下游变更原因、变更逻辑、变更时间等信息。下游对此次变更没有异议后，再按照约定时间执行发布变更，将变更对下游的影响降到最低。例如，在本教程中，如果表格存储数据源的表结构发生了变更，您需要通知ots_user_trace_log、ods_user_trace_log、dw_user_trace_log、rpt_user_trace_log表的责任人，及时更新表结构。

3.5 数据质量风险监控

数据质量风险监控主要针对数据的准确性、一致性和完整性。本教程使用DataWorks数据质量（DQC）功能，完成数仓各层次的数据质量监控。

前提条件

您需要首先完成教程《搭建互联网在线运营分析平台》，并保证您的DataWorks工作空间创建区域为华东2上海，详情参见业务场景与开发流程。您需要完成数据资产定级，本教程中定义为A2，详情请参见数据资产定级。

说明:

数据质量风险监控理论规范，请参见数据风险点监控。

背景信息

数据质量监控和数据资产等级对应，您可以根据以下因素细化您的监控配置，数据质量使用详情请参见数据质量概述。

- 监控分类：数据量、主键、离散值、汇总值、业务规则和逻辑规则。
- 监控粒度：字段级别、表级别。
- 监控层次：ODS、CDM、ADS三层数据，其中ODS和DWD层主要偏重数据的完整和一致性。DWS和ADS层数据量较小、逻辑复杂，偏重数据的准确性。

以下为不同数据资产等级和数仓层次数据的数据质量监控建议，仅供参考。

数据质量DQC监控规范									
监控分类					数据量	主键	离散值	汇总值	业务逻辑、规则
适合场景					所有非临时表都建议配置该项监控。	对于存在业务主键、逻辑主键的表需配置该监控。	维表、事实表中的维度值、状态值、可枚举的值需配置该监控。	汇总统计表中的汇总值需配置该监控。	1、重要指标的异常值监控。例如，正常UID长度是否为32位。 2、字段间的平衡值监控。例如，字段a与字段b满足一一对应关系等。 3、多表关联监控。例如两张表左关联，关联不上的记录数应等于0。
监控粒度					表级数据量监控	字段级	字段级	字段级	字段级/表级
常用监控规则					表行数波动/自助规则表行数>固定值	模板规则的字段空值、重复值/自定义规则监控联合主键空值、重复值情况	离散值分组个数/离散值分组个数波动/离散值状态值波动	模板规则的单字段大于0/自定义规则判断字段等于0所占的比例等	自定义规则
层次	表类型				规则配置				
ODS/DWD	离线表	A2	增量表	有周期规律	模板表行数波动率	空值、重复值唯一性	需监控	不涉及	需监控
				无周期规律	自助表行数>固定值	空值、重复值唯一性	需监控	不涉及	需监控
			全量表		模板表行数波动率	空值、重复值唯一性	需监控	不涉及	需监控
		A3	增量表	有周期规律	模板表行数波动率	空值、重复值唯一性	需监控	不涉及	不涉及
				无周期规律	自助表行数>固定值	空值、重复值唯一性	需监控	不涉及	不涉及
			全量表		模板表行数波动率	空值、重复值唯一性	需监控	不涉及	不涉及
		A4	增量表	有周期规律	模板表行数波动率	空值、重复值唯一性	需监控	不涉及	不涉及
				无周期规律	自助表行数>固定值	空值、重复值唯一性	需监控	不涉及	不涉及
			全量表		模板表行数波动率	空值、重复值唯一性	需监控	不涉及	不涉及
		Ax	增量表	有周期规律	模板表行数波动率	空值、重复值唯一性	不涉及	不涉及	不涉及
				无周期规律	自助表行数>固定值	空值、重复值唯一性	不涉及	不涉及	不涉及
			全量表		模板表行数波动率	空值、重复值唯一性	不涉及	不涉及	不涉及
DWS/ADS	离线表	A2	增量表	有周期规律	模板表行数波动率	空值、重复值唯一性	需监控	需监控	需监控
				无周期规律	自助表行数>固定值	空值、重复值唯一性	需监控	需监控	需监控
			全量表		模板表行数波动率	空值、重复值唯一性	需监控	需监控	需监控
		A3	增量表	有周期规律	模板表行数波动率	空值、重复值唯一性	需监控	需监控	不涉及
				无周期规律	自助表行数>固定值	空值、重复值唯一性	需监控	需监控	不涉及
			全量表		模板表行数波动率	空值、重复值唯一性	需监控	需监控	不涉及
		A4	增量表	有周期规律	模板表行数波动率	空值、重复值唯一性	需监控	不涉及	不涉及
				无周期规律	自助表行数>固定值	空值、重复值唯一性	需监控	不涉及	不涉及
			全量表		模板表行数波动率	空值、重复值唯一性	需监控	不涉及	不涉及
		Ax	增量表	有周期规律	模板表行数波动率	空值、重复值唯一性	不涉及	不涉及	不涉及
				无周期规律	自助表行数>固定值	空值、重复值唯一性	不涉及	不涉及	不涉及
			全量表		模板表行数波动率	空值、重复值唯一性	不涉及	不涉及	不涉及

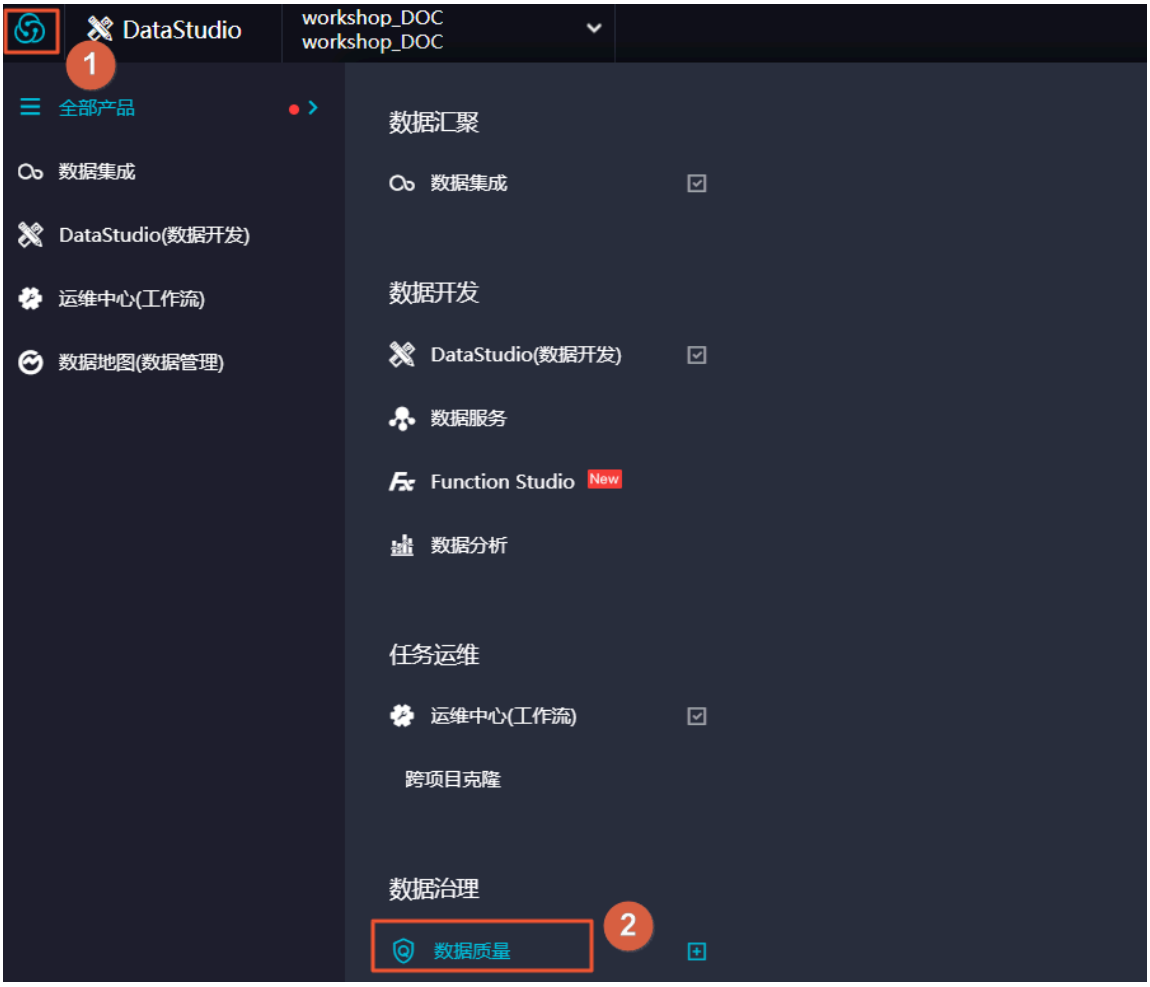
操作步骤

1. ODS层数据质量监控

ODS层表里的数据来源于OSS上的日志文件，作为源头表，您需要尽早判断此表分区中是否有数据。如果这张表中没有数据，后续任务运行无意义，则需要阻止后续任务运行。

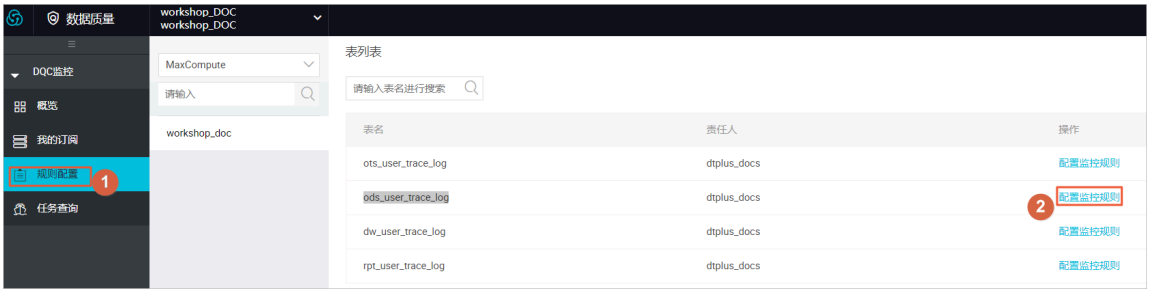
a) 进入数据质量。

在您的数据开发页面，单击左上角图标，选择数据质量。



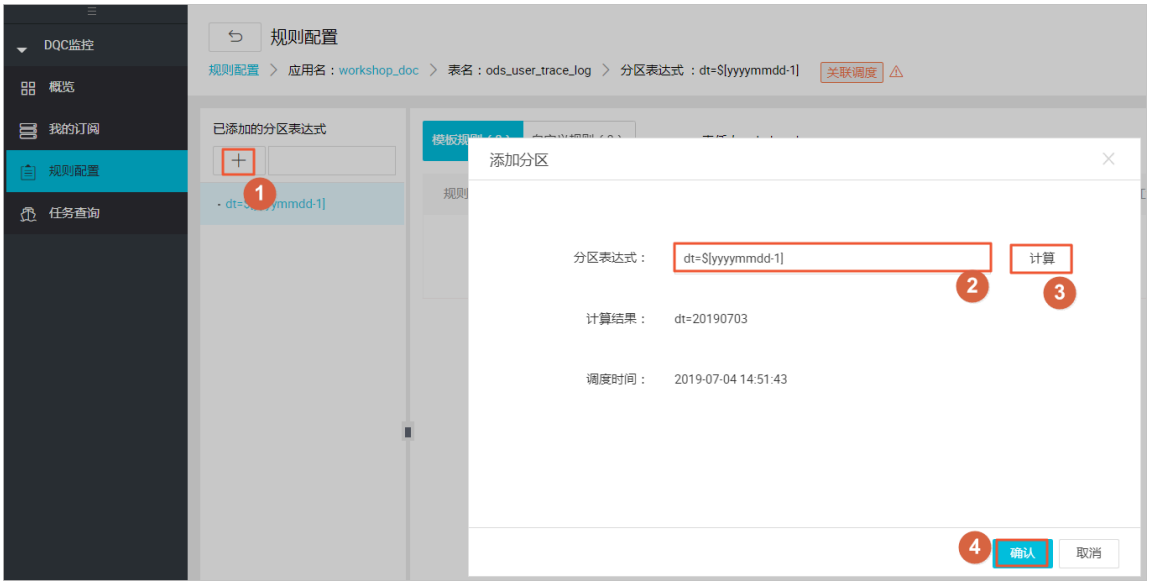
b) 进入ods_user_trace_log规则配置页面。

在规则配置页面找到代表外部数据源的表ods_user_trace_log，单击配置监控规则。



c) 添加分区。

单击+，在分区表达式一栏输入 `dt=$[yyyymmdd-1]`，对应表ods_user_trace_log的分区格式`${bdp.system.bizdate}`（获取到前一天的日期）。分区表达式的详细信息请参见参数配置，若表中无分区列，可以配置无分区。



d) 单击创建规则。

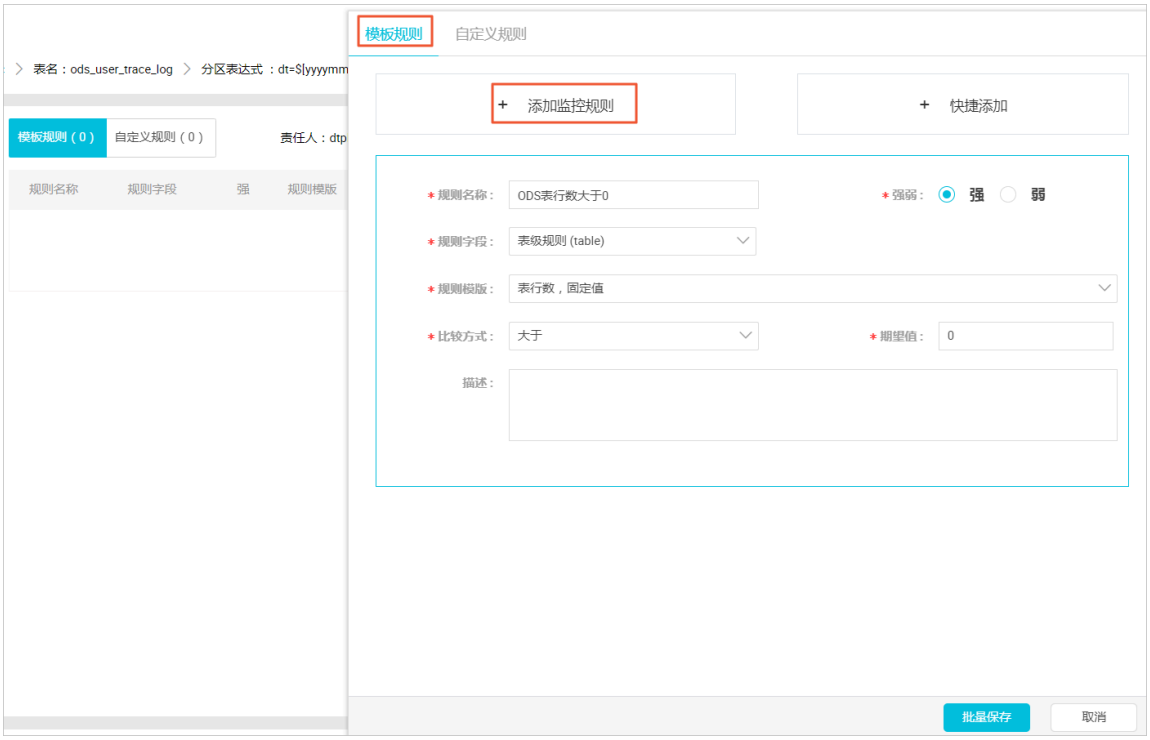
e) 监控表行数大于0。

单击模板规则 > 添加监控规则。

在配置规则时，选择规则模板为表行数、固定值，将规则的强度设置为强，比较方式设置为期望值大于0。目的为保证ODS层分区内存在表数据。

说明:

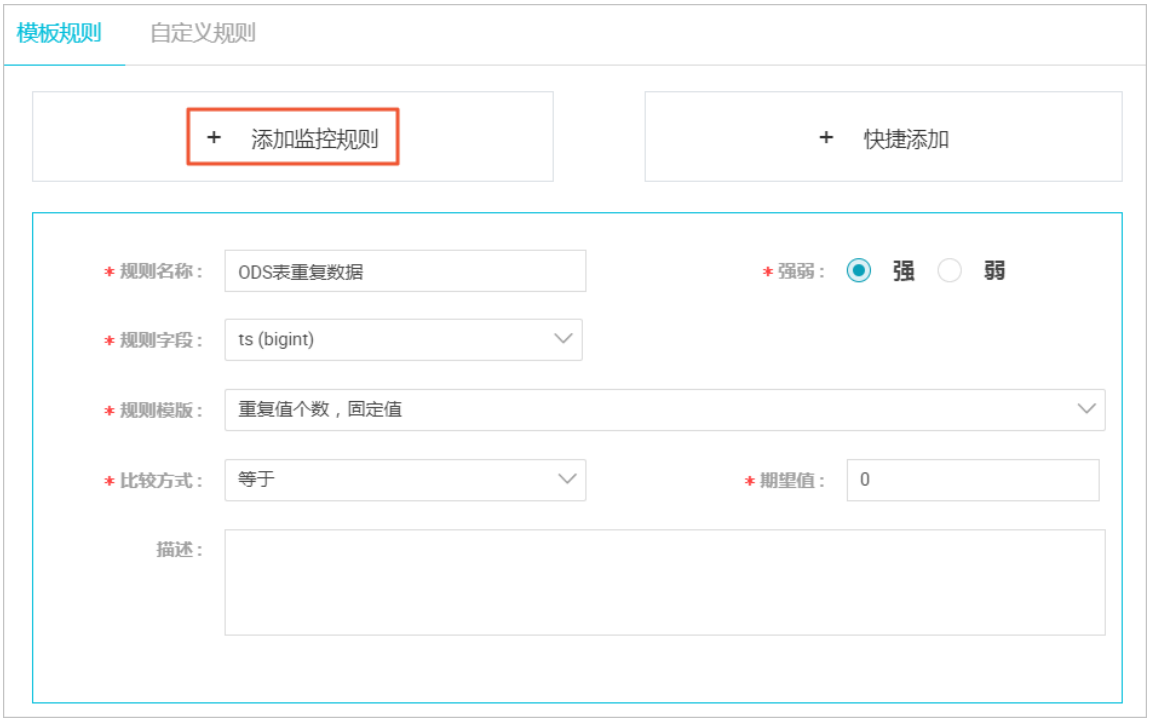
只有强规则下红色报警会导致任务阻塞，阻塞会将任务的实例状态置为失败。



f) 监控重复数据。

单击添加监控规则。

配置规则时，选择规则字段为ts(bigint)，规则模板为重复值个数、固定值，将规则的强度设置为强，比较方式设置为期望值等于0。ts(bigint)值为用户时间戳，目的是避免ODS层出现重复的数据。



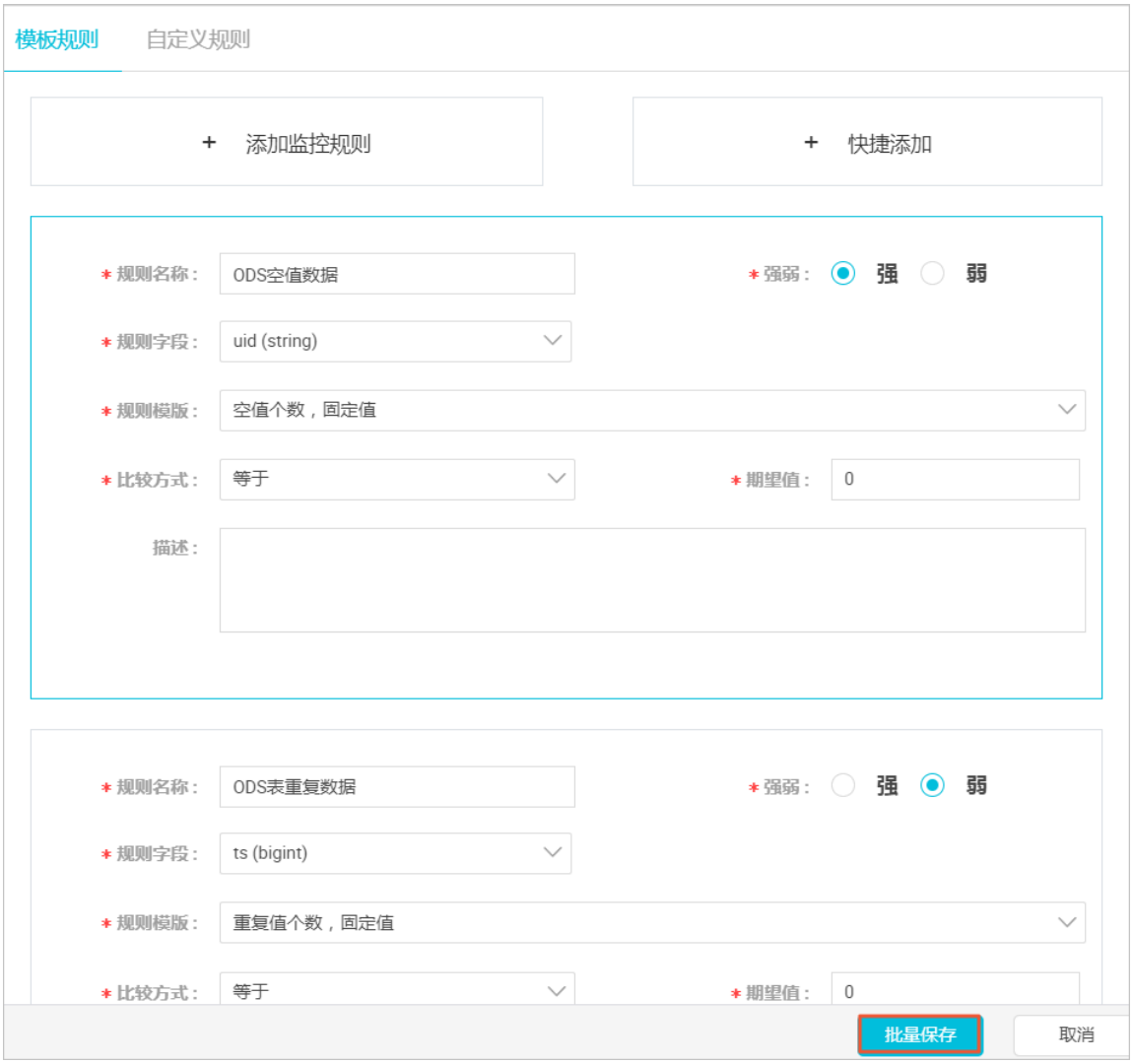
g) 监控空值数据。

单击添加监控规则。

配置规则时，选择规则字段为uid(string)，规则模板为空值个数、固定值，将规则的强度设置为强，比较方式设置为期望值等于0。uid(string)值为用户ID，目的是避免出现用户ID为空值的脏数据。

h) 批量保存规则。

完成上述操作后，单击批量保存。

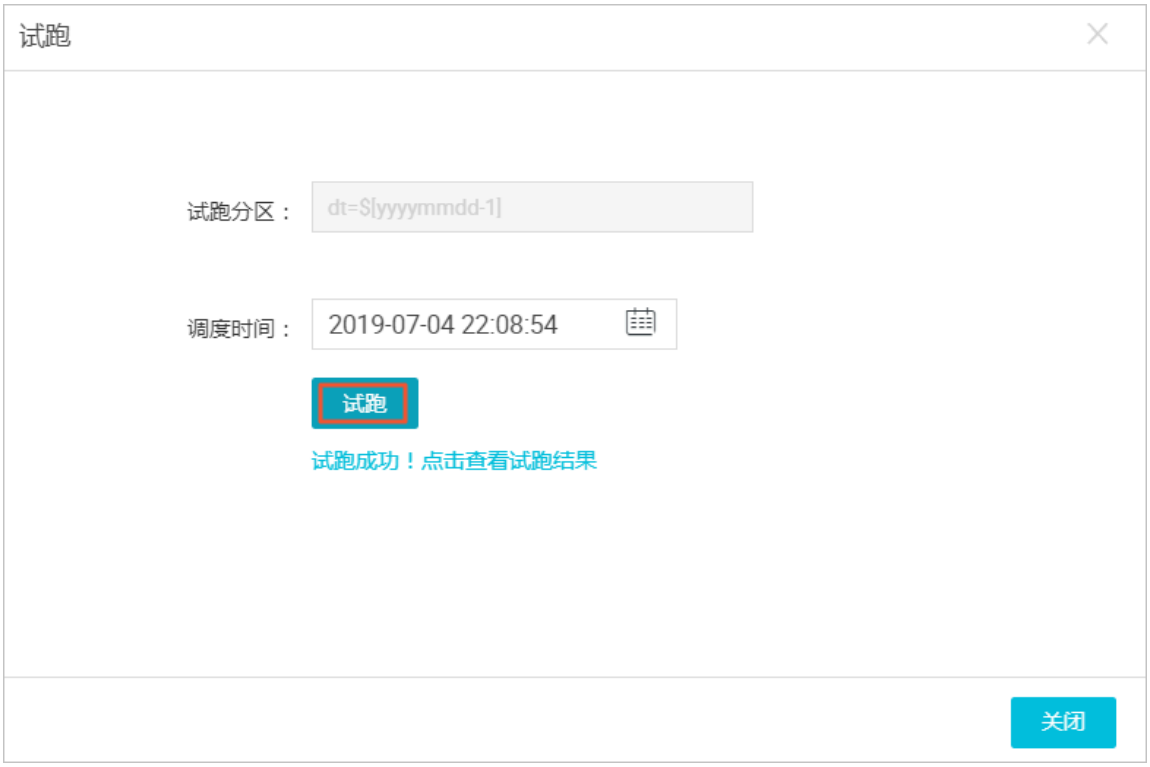


i) 规则试跑。

右上角有一个节点试跑的按钮，可以在规则配置完毕后，进行规则校验。单击试跑按钮，可立即触发数据质量的校验规则。

j) 查看试跑结果。

单击试跑后，您可以单击试跑成功！点击查看试跑结果。



在弹出的页面中，您可以查看表数据是否已符合您的规则。

说明：

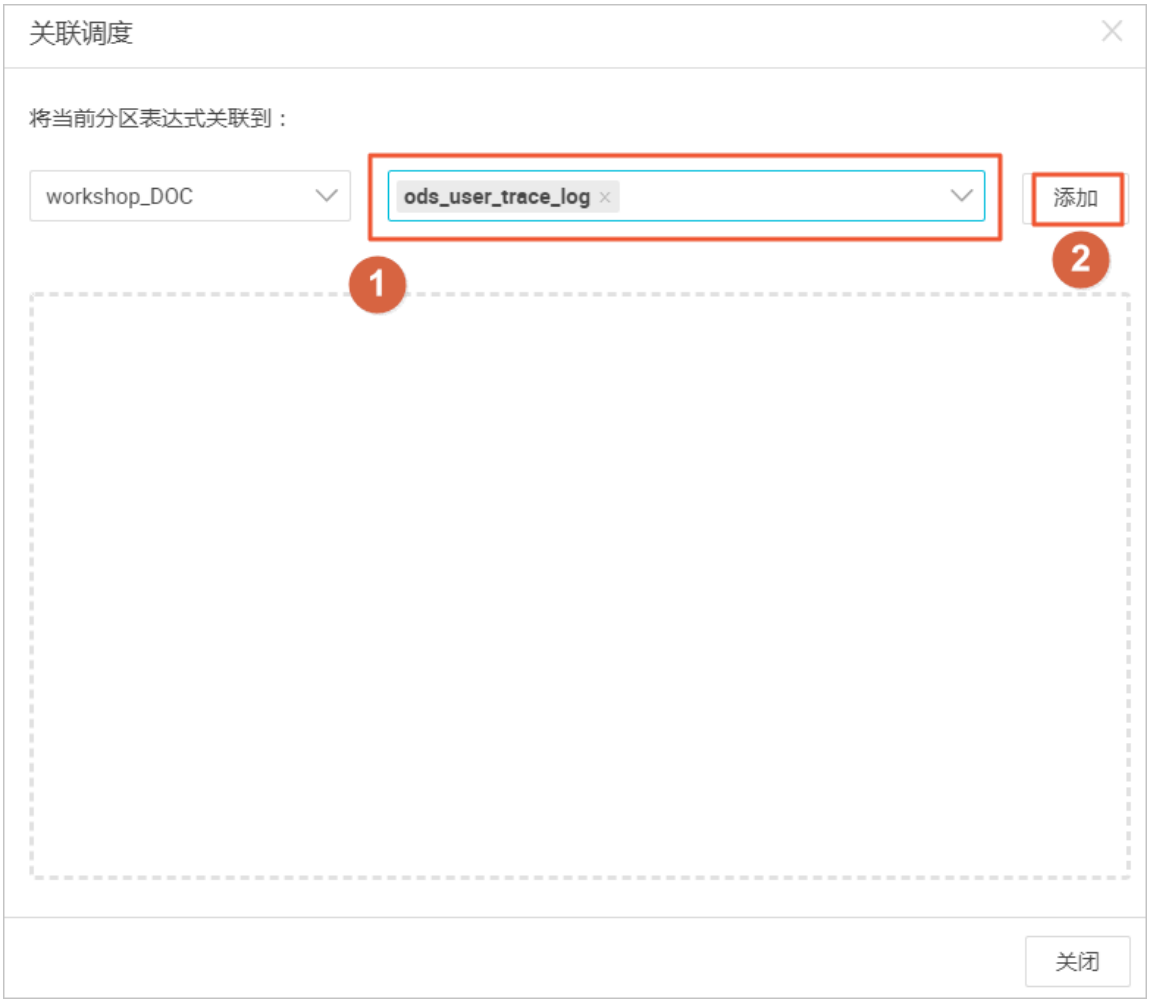
可根据试跑结果，来确认此次任务产出的数据是否符合预期。建议每个表规则配置完毕后，都进行一次试跑操作，以验证表规则的适用性。

k) 关联调度。

在规则配置完毕，且试跑又都成功的情况下。您需要将表和其产出任务进行关联，这样每次表的产出任务运行完毕后，都会触发数据质量规则的校验，以保证数据的准确性。在表规则和调度任务绑定后，任务实例运行完毕，都会触发数据质量的检查。

在表规则配置界面，单击关联调度，配置规则与任务的绑定关系。

在弹框中输入您需要关联的任务节点名称，单击添加。



关联调度后，表名后的小图标会变成蓝色。

l) 配置任务订阅。

关联调度后，每次调度任务运行完毕，都会触发数据质量的校验。数据质量支持设置规则订阅，可以针对重要的表及其规则设置订阅，设置订阅后会根据数据质量的校验结果进行告警，从而实现对校验结果的跟踪。若数据质量校验结果异常，则会根据配置的告警策略进行通知。

单击订阅管理，设置接收人以及订阅方式，目前支持邮件通知、邮件和短信通知、钉钉群机器人和钉钉群机器人@ALL四种方式。

订阅管理设置完毕后，可以在我的订阅中进行查看及修改，建议您订阅所有规则。

2. CDM层数据质量监控

CDM层数据质量监控配置方法与ODS层相同，区别在于监控规则的配置。

a) 添加分区表达式。

进入dw_user_trace_log表的规则配置页面，同样配置分区为dt=$[yyyymmdd-1]，完成添加后您可以在界面中看到已添加的分区表达式。

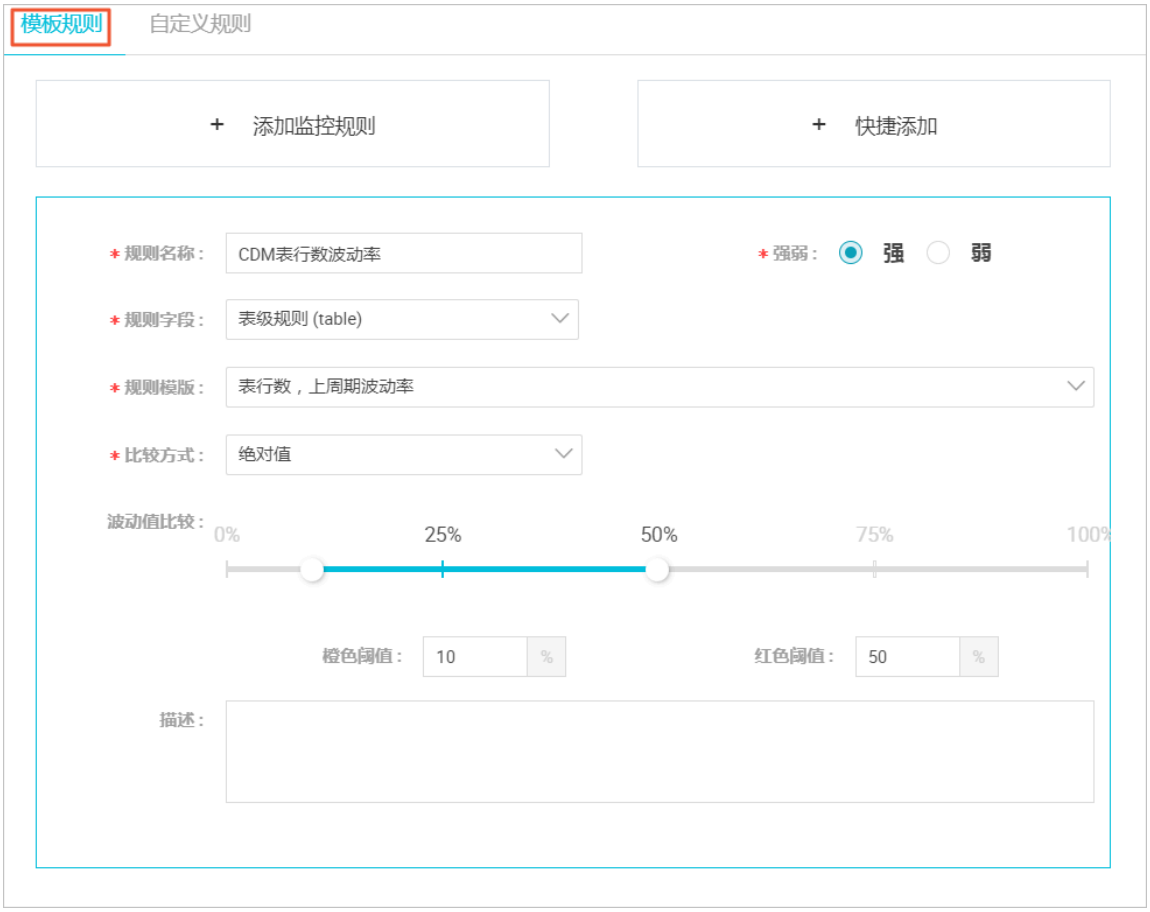
b) 监控表行数及空值数据。

表行数和空值数据的监控规则配置与ODS层相同，完成配置后如下图所示。

c) 监控表行数波动率。

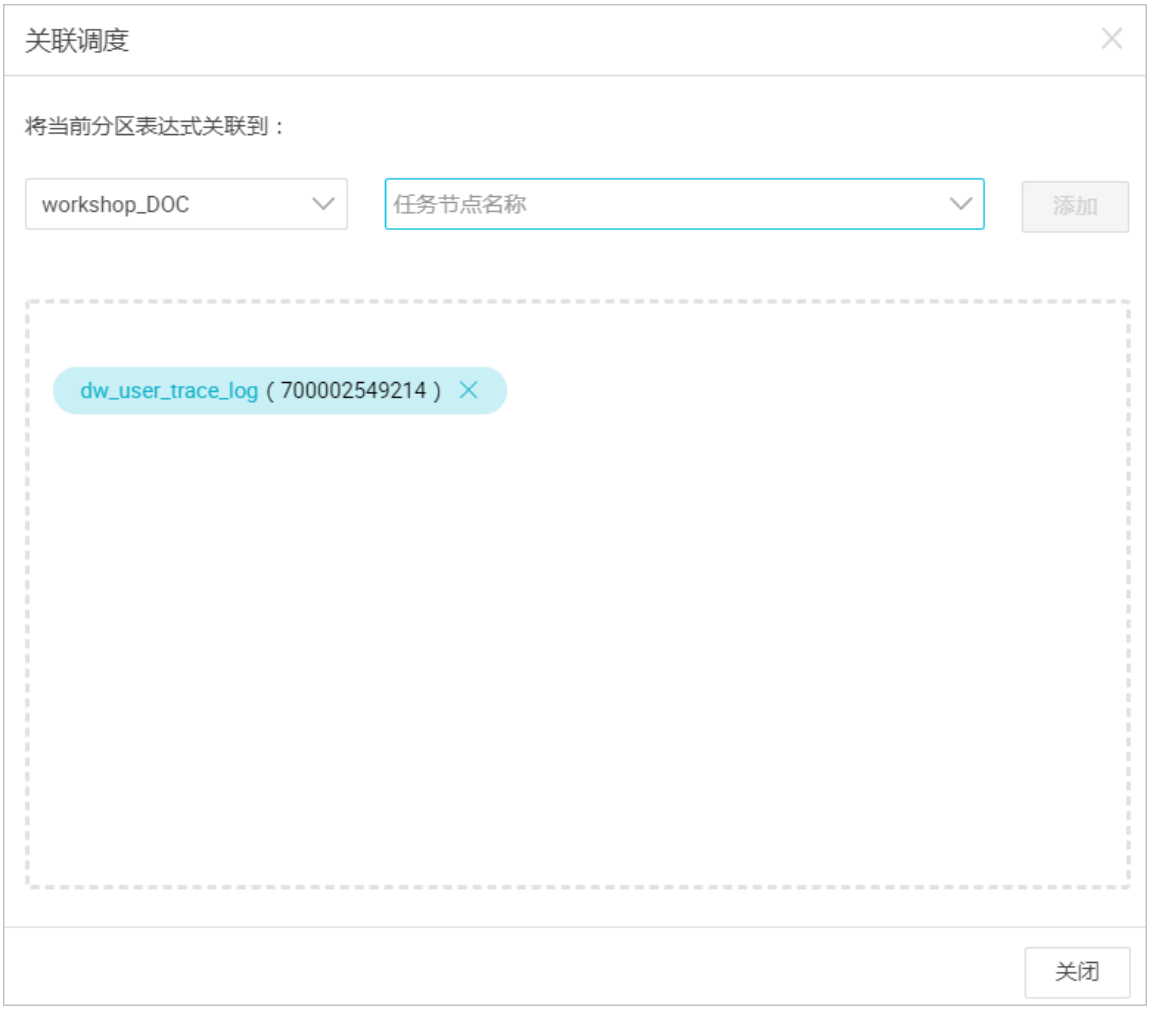
监控表行数波动率主要是为了避免出现突发的大量脏数据的污染。配置规则时，选择规则字段为表级规则(table)，规则模板为表行数、上周期波动率，将规则的强度设置为强，比较方

式设置为绝对值。橙色阈值为10，红色阈值为50，代表当表行数波动率到达50%时，会产生红色报警。



d) 规则试跑并关联调度。

方法同ODS层。


关联调度
将当前分区表达式关联到：
workshop_DOC
任务节点名称
添加
dw_user_trace_log (700002549214)
关闭

3. ADS层数据质量监控

ADS层数据质量监控配置方法与ODS层相同，区别在于监控规则的配置。

a) 添加分区表达式。

进入rpt_user_trace_log表的规则配置页面，同样配置分区为dt=$[yyyymmdd-1]，完成添加后您可以在界面中看到已添加的分区表达式。

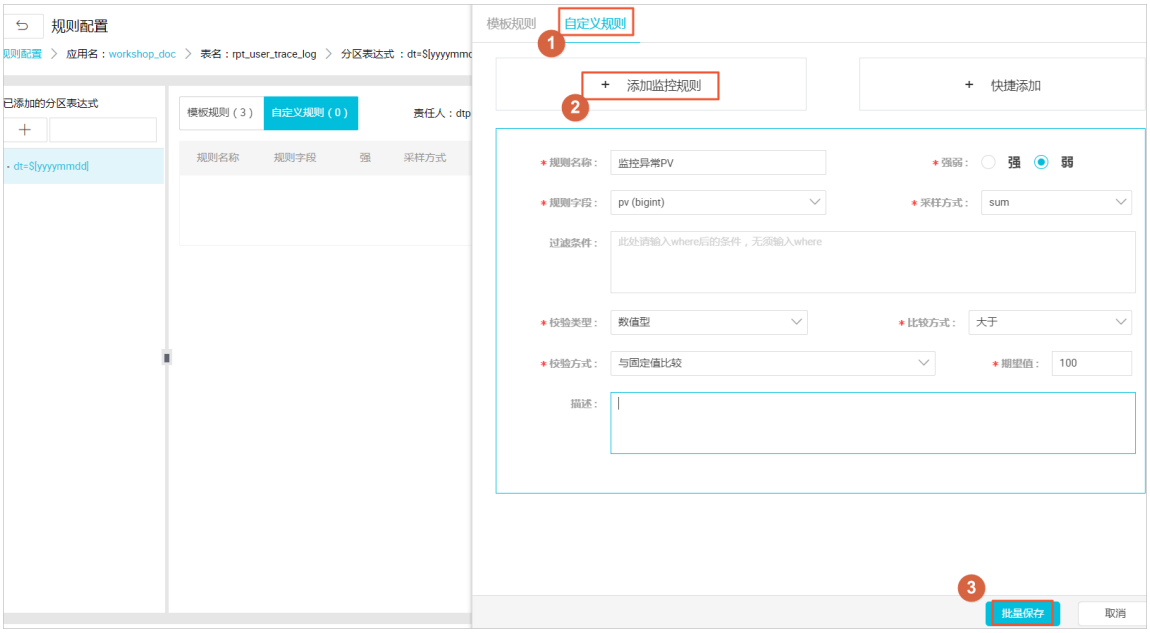
b) 监控表行数、波动率及空值数据。

监控表行数、波动率和空值数据的监控规则配置与CDM层相同。由于在数仓分层中，越靠近应用层数据越少、约束性越低，强弱选择为弱，完成配置后如下图所示。

c) 监控表异常PV。

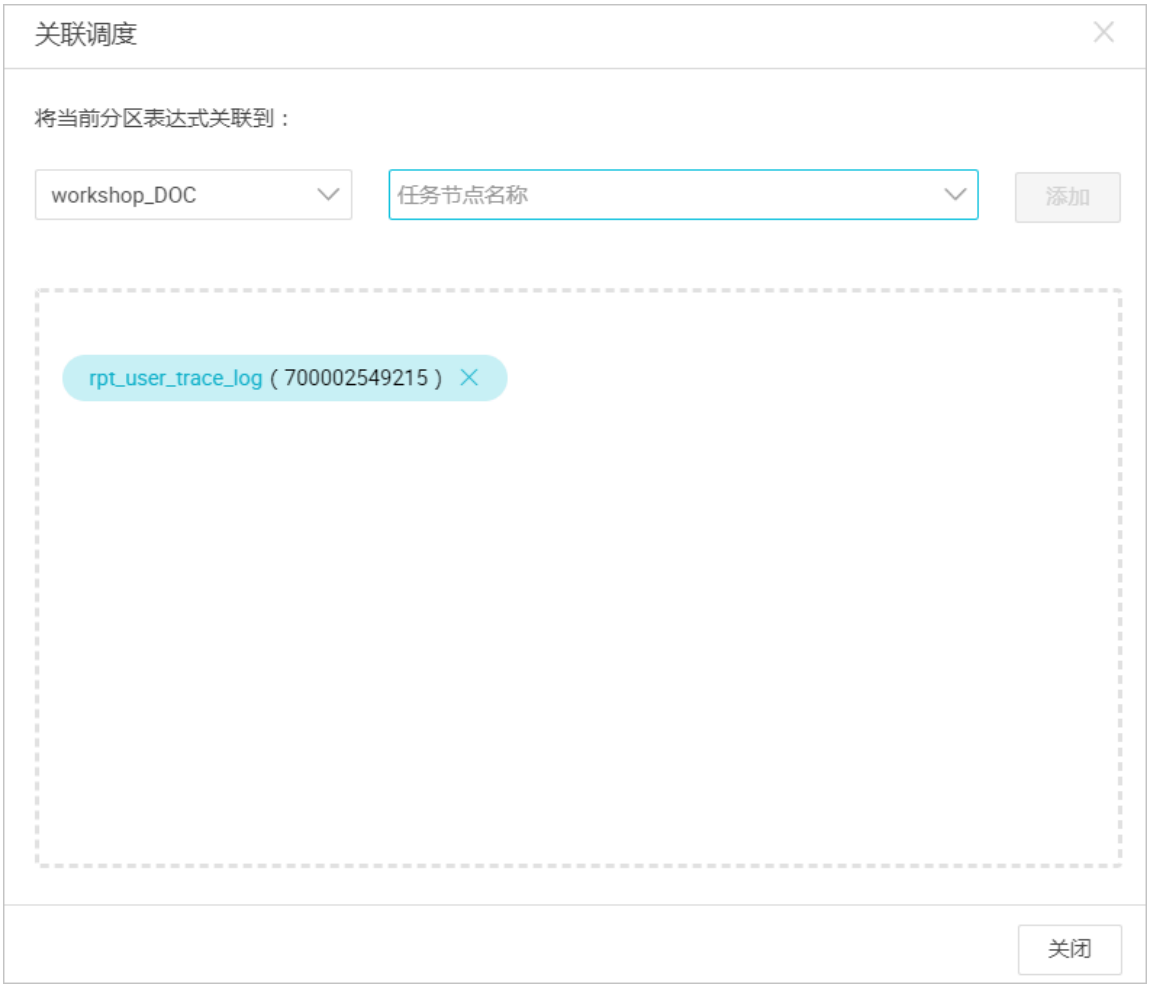
您可以利用自定义规则功能监控ADS层的应用数据。配置规则时，选择规则字段为pv(bigint)，采样方式为sum，将规则的强度设置为弱，比较方式设置为大于期望

值100。这样，当PV和异常锐减到100时，您可以及时收到告警。完成配置后，单击批量保存。



d) 规则试跑并关联调度。

方法同ODS层。



3.6 数据及时性监控

基于MaxCompute的离线任务会对数据产出有时间要求，在确保数据准确性的前提下，您需要进一步让数据能够及时提供服务。本教程为您演示如何使用DataWorks智能监控功能完成数据及时性的监控。

前提条件

本教程为您演示基础版DataWorks的基本智能监控功能：规则管理。如果您想使用完整的智能监控功能，需至少购买标准版DataWorks，详情请参见DataWorks增值版本功能对比。关于DataWorks智能监控功能详情请参见智能监控概述。

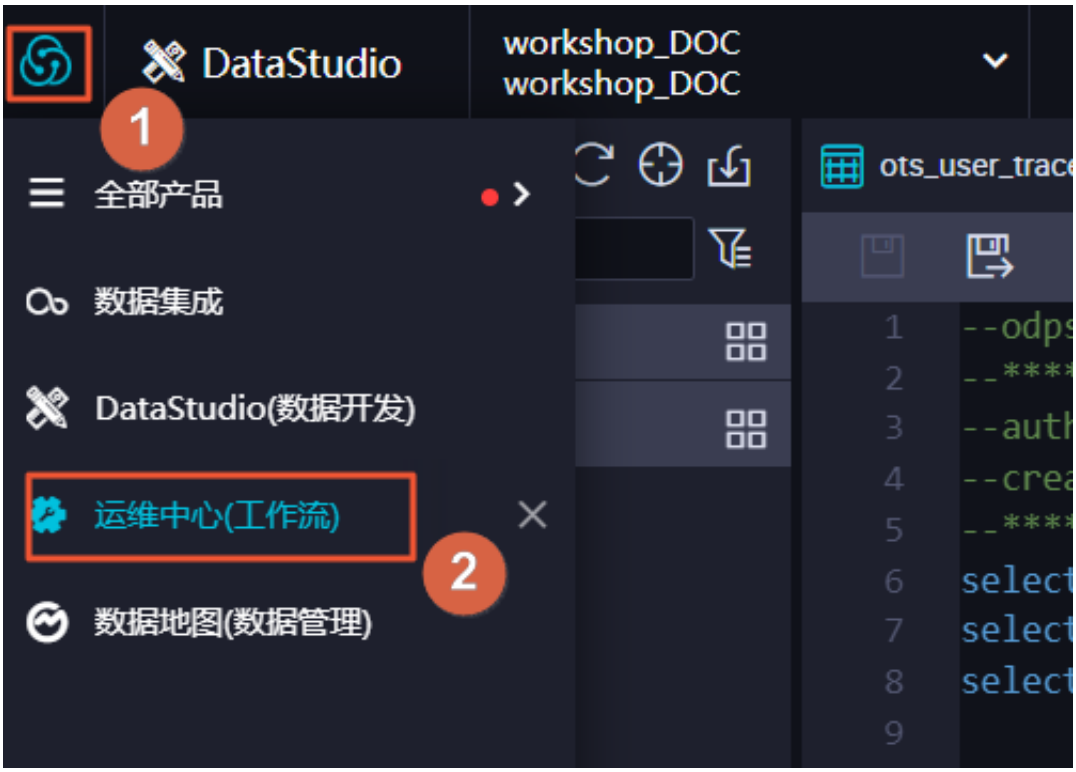
背景信息

在对数据产出及时性监控前，您需要首先确定调度任务的优先级。数据资产等级越高的任务节点，优先级越高，您可以给予更加严格的数据及时性监控和告警规则。

操作步骤

1. 进入规则管理页面。

   在DataStudio页面单击运维中心（工作流）。

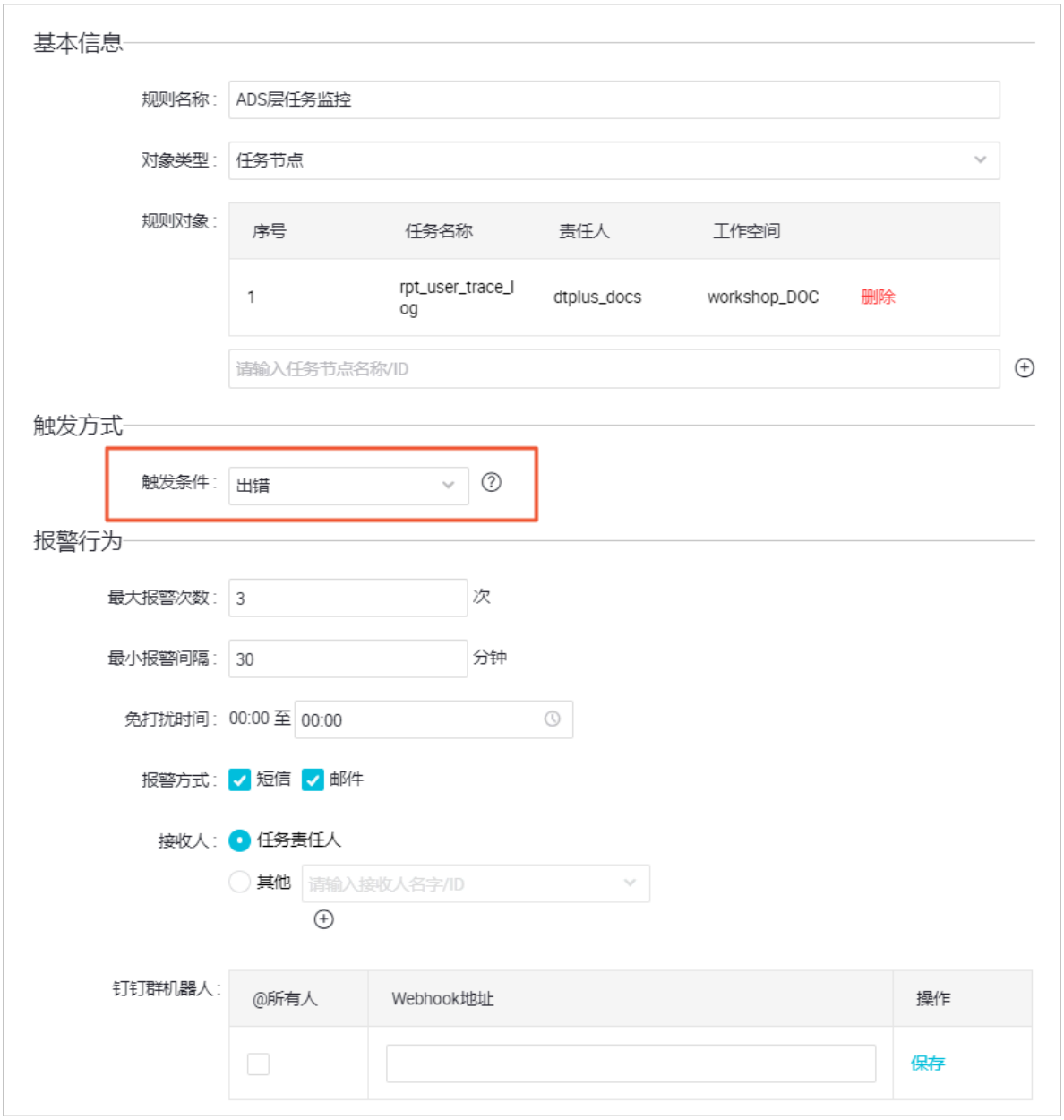


   单击智能监控 > 规则管理。关于规则管理的详情请参见规则管理。

2. 新建自定义规则。

单击新建自定义规则，输入参数后单击确定即可，如下图所示。

在本例中，监控整个业务流程每次运行时间不可超过30分钟。如果运行时间超过30分钟，即开始每30分钟产生告警1次，连续告警3次，告警方式为邮件及短信。对于重要的任务节点，您还可以单独设置任务节点规则，并定义其他触发条件如下图所示。


基本信息
规则名称：
ADS层任务监控
对象类型：
任务节点
规则对象：
序号
任务名称
责任人
工作空间
1
rpt_user_trace_log
dtplus_docs
workshop_DOC
删除
请输入任务节点名称/ID
触发方式
触发条件：
出错
报警行为
最大报警次数：
3
次
最小报警间隔：
30
分钟
免打扰时间：
00:00 至
00:00
报警方式：
短信
邮件
接收人：
任务责任人
其他
请输入接收人名字/ID
钉钉群机器人：
@所有人
Webhook地址
操作
保存

3. 数据及时性优化。

通常，影响数据按时产出的主要原因有：

- 计算资源不足：
  - 资源总量不足。例如，资源上限为500，但您提交了需要1000资源的任务。
  - 资源分配不合理，重要任务未优先分配资源。
- 代码执行效率低：
  - 代码冗余，例如扫描所有分区。
  - 节点任务配置不合理，例如出现长尾问题。
- 缺少问题紧急预案，运维人员无法应对。

面对上述问题，您可以考虑从以下方面优化，提升数据及时性：

- 扩容计算资源，或让您的核心计算任务独占资源。
- 分级错峰：高峰时段让低优先级任务延迟启动。
- 在任务正式运行前，进行充分的压力测试。