ALIBABA CLOUD

阿里云

大数据计算服务
使用教程

文档版本：20211222

阿里云

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
危险	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。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会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，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	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。	单击**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。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	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。	`switch {active|stand}`

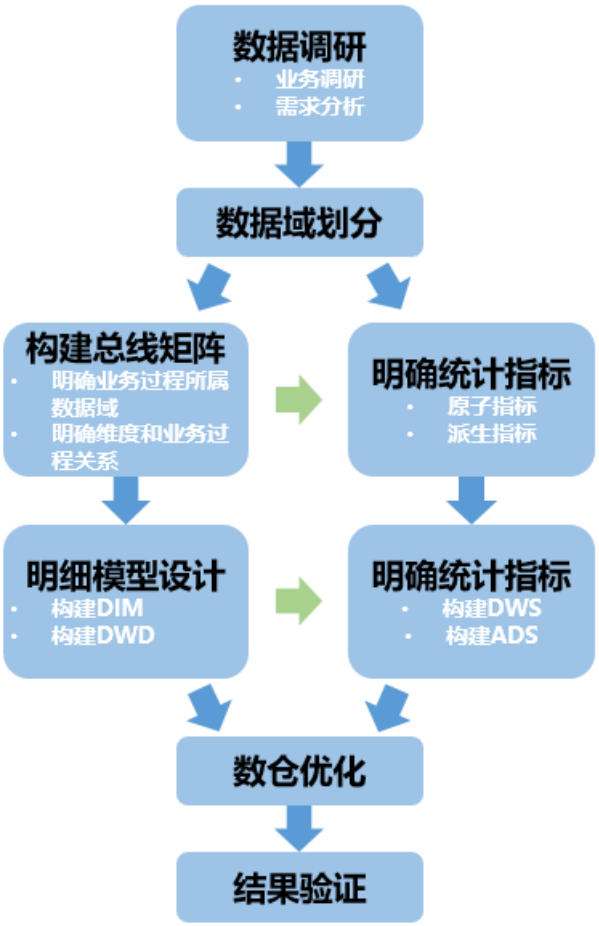
目录

1.构建与优化数据仓库

1.1. 数仓构建流程

本文为您介绍构建MaxCompute数据仓库的流程。

构建MaxCompute数据仓库的整体流程如下。



基本概念

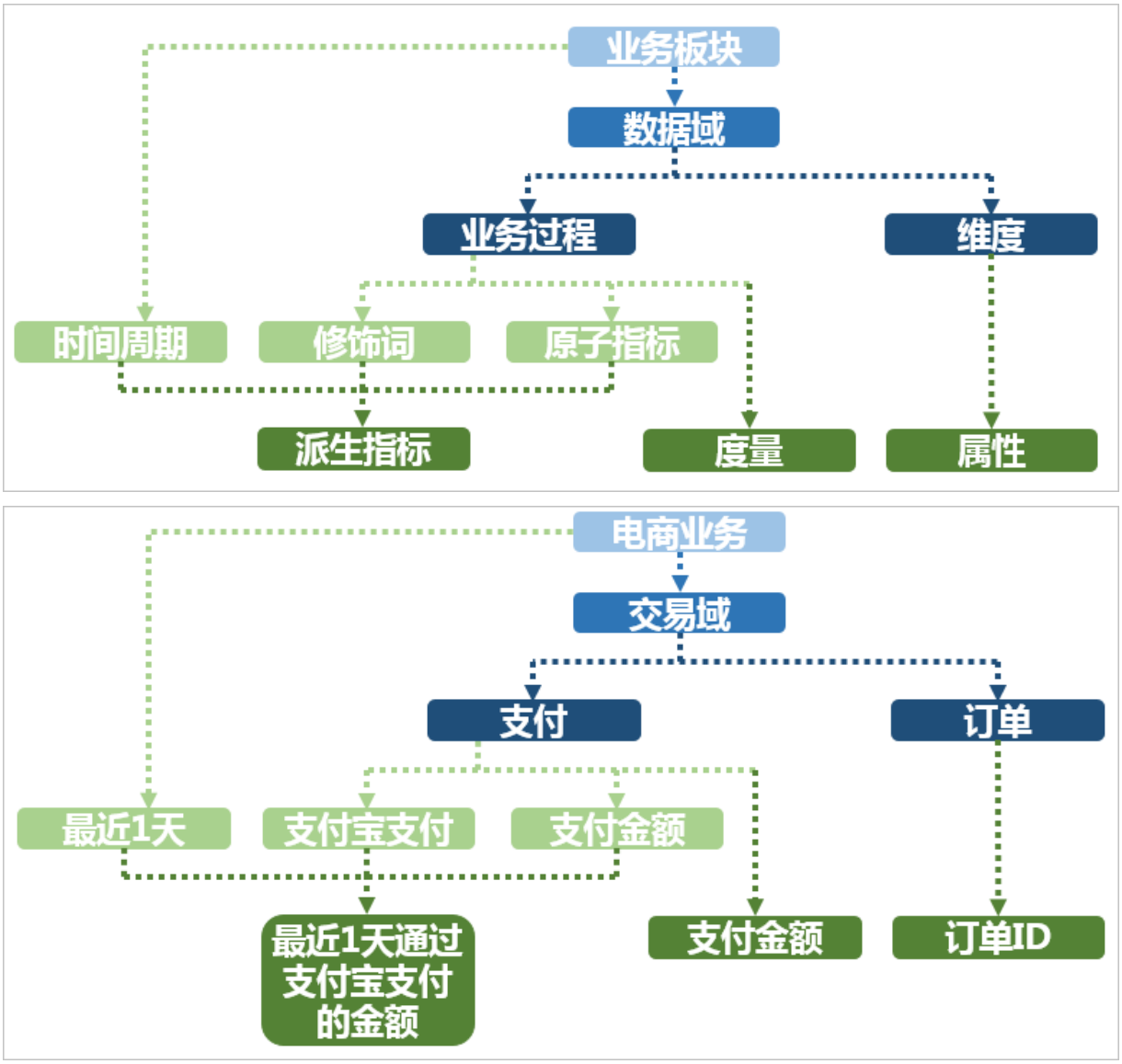
在正式学习本教程之前，您需要首先理解以下基本概念：

- 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，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

您在构建数据仓库之前，首先需要确定构建数据仓库的目标与需求，并进行全面的业务调研。您需要了解真实的业务需求，以及确定数据仓库要解决的问题。

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和运营人员的需求，沉淀出相关文档。

您可以通过调查表和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、运营和维护部门人员，各个部门对数据仓库的需求不同，您需要对不同部门分别进行调研。

2. 用户的整体业务架构，各个业务板块之间的联系和信息流动的流程。

   您需要梳理出整体的业务数据框架。

3. 各个已有的业务板块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、营销和服务四个模块，每个板块的需求和数据应用都不同。您在构建数据仓库之前，首先需要明确构建数据仓库的业务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数据功能模块。数据功能模块通常和业务板块紧耦合，对应一个或多个表，可以作为构建数据仓库的数据源。下表展现的是一个营销业务板块的数据功能模块。

数据功能模块	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代表包含该数据功能模块，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销售额商品名称和各省客户购买力分布（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数据功能模块。

需求分析

在未考虑数据分析师和业务运营人员的数据需求的情况下，单纯根据业务调研结果构建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：

- 根据与分析师和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，订单数是成交量的度量。
- 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？
- 数据是否需要冗余或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及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。

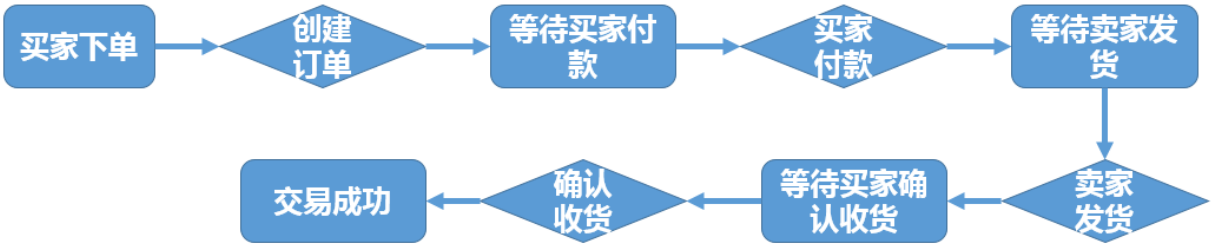
您可以采用过程分析法，将整个业务过程涉及的每个环节一一列清楚，包括技术、数据、系统环境等。在分析企业的工作职责范围（部门）后，您也可以借助工具通过逆向工程抽取业务系统的真实模型。您可以参考业务规划设计文档以及业务运行（开发、设计、变更等）相关文档，全面分析数据仓库涉及的源系统及业务管理系统：

- 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、当前状态还是事件流转效率。

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（交易成功）步骤即可。

在明确用户的业务过程之后，您可以根据需要对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，目的是便于管理和应用数据。

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，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、单击
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。从商品维度我们可以定义出以下维度的属性：

- 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 删除 2 下架 3 从未上架
- 商品所在城市
- 商品所在省份

从地域维度，我们可以定义出以下维度的属性：

- 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）

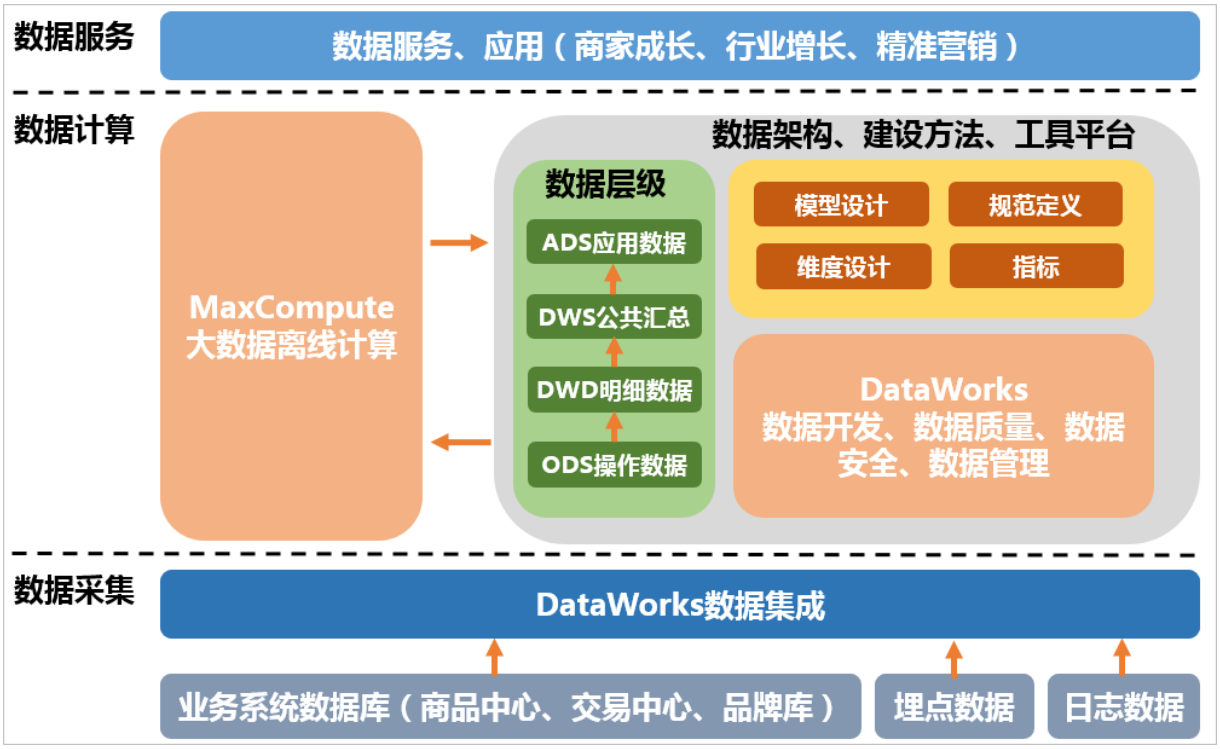
最近一天全省厨具类目各商品销售总额进行降序排序后取前10名的名称，即可得到该类目Top10销售额商品名称。

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。

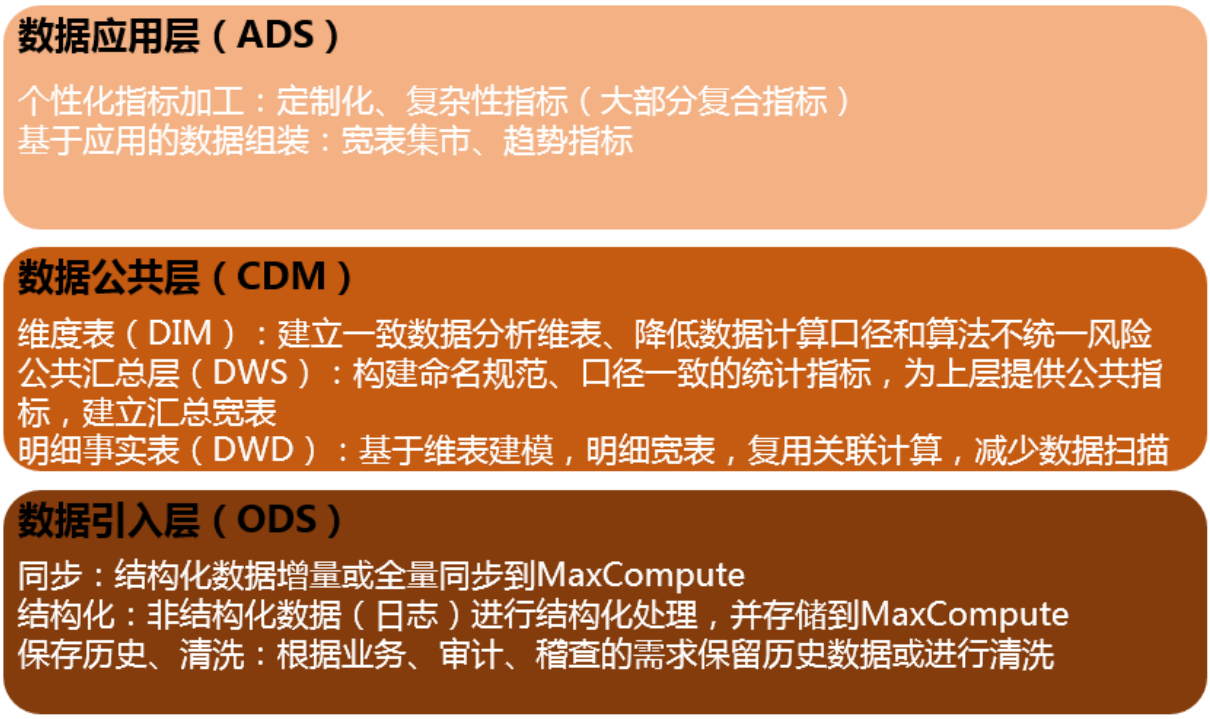
完整的技术架构图如下图所示。其中，DataWorks的数据集成负责完成数据的采集和基本的ETL。MaxCompute作为整个大数据开发过程中的离线计算引擎。DataWorks则包括数据开发、数据质量、数据安全、数据管理等在内的一系列功能。



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。



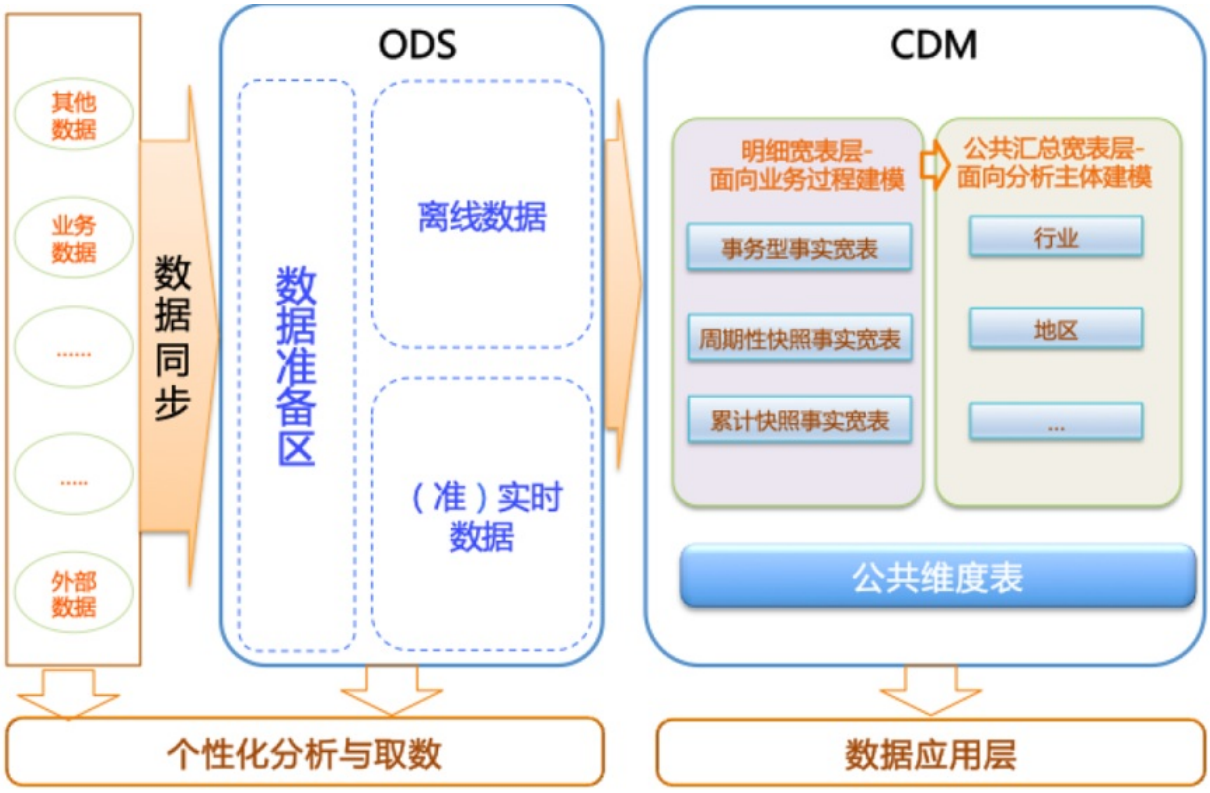
- 数据引入层ODS（Operation Data Store）：存放未经过处理的原始数据至数据仓库系统，结构上与源系统保持一致，是数据仓库的数据准备区。主要完成基础数据引入到MaxCompute的职责，同时记录基础数据的历史变化。
- 数据公共层CDM（Common Data Model，又称通用数据模型层），包括DIM维度表、DWD和DWS，由ODS层数据加工而成。主要完成数据加工与整合，建立一致性的维度，构建可复用的面向分析和统计的明细事实表，以及汇总公共粒度的指标。
  - 公共维度层（DIM）：基于维度建模理念思想，建立整个企业的一致性维度。降低数据计算口径和算法不统一风险。

    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，即宽表化处理。

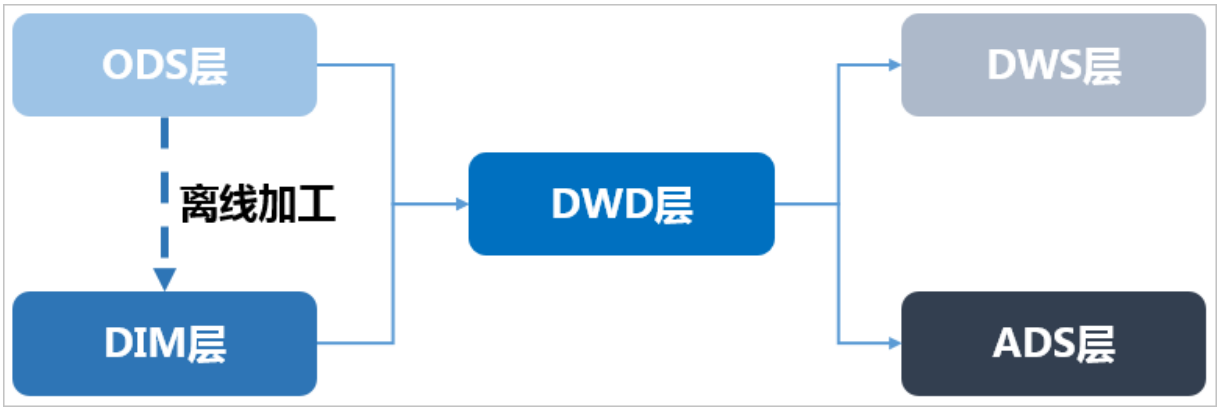
    明细粒度事实层的表通常也被称为逻辑事实表。
- 数据应用层ADS（Application Data Service）：存放数据产品个性化的统计指标数据。根据CDM与ODS层加工生成。

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. 数据模型

1.3.3.1. 数据引入层（ODS）

ODS（Operational Data Store）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支持的数据类型版本说明。

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，可以通过添加_col后缀解决。

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。更多建表信息，请参见表操作。

```
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 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 '商品正常购买ODS'
PARTITIONED BY (ds      STRING COMMENT '格式：YYYYMMDD')
LIFECYCLE 400;
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'
```

```
    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。例如，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 `start_dt<=20160101` 并且 `end_dt>20160101` 。

缓慢变化维度

MaxCompute不推荐使用代理键，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。

在不使用代理键的情况下，缓慢变化维度可以通过快照方式处理。

快照方式下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，每天保留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空间的极大浪费。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。

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到MaxCompute一次，保持表结构的一致性。
- 数据集成提供数据同步解决方案，您可以通过配置同步规则，实现离线数据全量及增量同步、增量数据实时写入、增量数据和全量数据定时自动合并写入新的全量表分区。详情请参见同步解决方案。
- 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。

1.3.3.2. 公共维度汇总层（DIM）

公共维度汇总层DIM（Dimension）基于维度建模理念，建立整个企业的一致性维度。

公共维度汇总层（DIM）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将数据平台上构建的物理化的表，采用宽表设计的原则。因此，公共维度汇总层（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。本教程以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

  0表示全新，1表示闲置，2表示二手。
- 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表示正常，1表示用户删除，2表示下架，3表示从未上架。
- 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，您可以对维度进行补充，进而生成维表。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。
- 维表是否需要垂直拆分。

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查询变慢，则需要考虑对字段进行拆分，创建多个维表。

- 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。

- 核心维表的产出时间。通常有严格的要求。

设计维表的主要步骤如下：

1. 初步定义维度。

   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、卖家和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）、类目、卖家和店铺等相关维表中选择维度属性或生成新的维度属性。维度属性的设计需要注意：

   - 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}]，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。例如，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;
```

1.3.3.3. 明细粒度事实层（DWD）

明细粒度事实层DWD（Data Warehouse Detail）以业务过程驱动建模，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描述业务过程，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，作为事实逻辑表的依据。事实逻辑表的描述信息是事实属性，事实属性中的外键字段通过对应维度进行关联。

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：dwd_asale_trd_mbr_di。
- 交易订单信息事实表：dwd_asale_trd_ord_di。

DWD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 dwd_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;
```

1.3.3.4. 公共汇总粒度事实层（DWS）

公共汇总粒度事实层DWS（Data Warehouse Summary）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至少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;
```

1.3.3.5. 附录：ODS层示例数据

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

1.3.4. 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）。

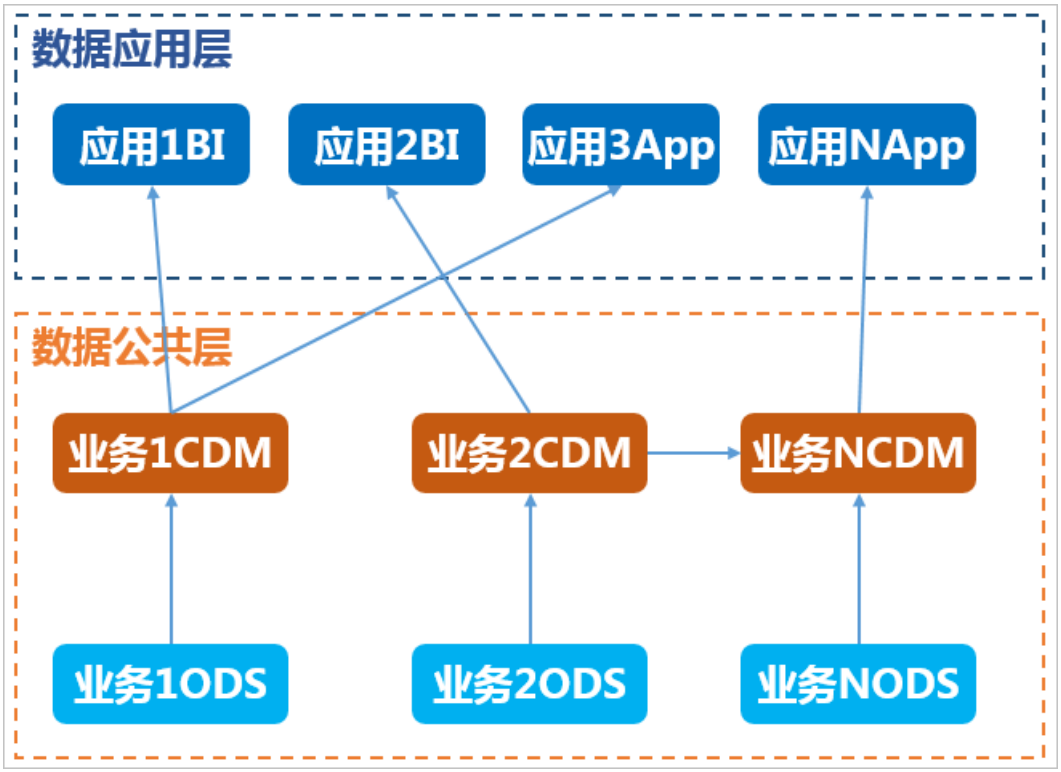
- 一个计算刷新任务只允许一个输出表，特殊情况除外。
- 如果多个任务刷新输出一个表（不同任务插入不同的分区），DataWorks上需要建立一个虚拟任务，依赖多个任务的刷新和输出。通常，下游应该依赖此虚拟任务。
- CDM汇总层优先调用CDM明细层，可累加指标计算。CDM汇总层尽量优先调用已经产出的粗粒度汇总层，避免大量汇总层数据直接从海量的明细数据层中计算得出。
- CDM明细层累计快照事实表优先调用CDM事务型事实表，保持数据的一致性产出。
- 有针对性地建设CDM公共汇总层，避免应用层过度引用和依赖CDM层明细数据。

1.4. 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` 结尾，例如 `asale_ods` 。
- 对于CDM层项目，建议以 `cdm` 结尾，例如 `asale_cdm` 。
- ADS应用层数据分为两类：
  - 数据报表、数据分析等以 `bi` 结尾，例如 `asale_bi` 。
  - 数据产品应用以 `app` 结尾，例如 `asale_app` 。

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产环境项目安全，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，建议您在优化性能之前首先建立性能基准。

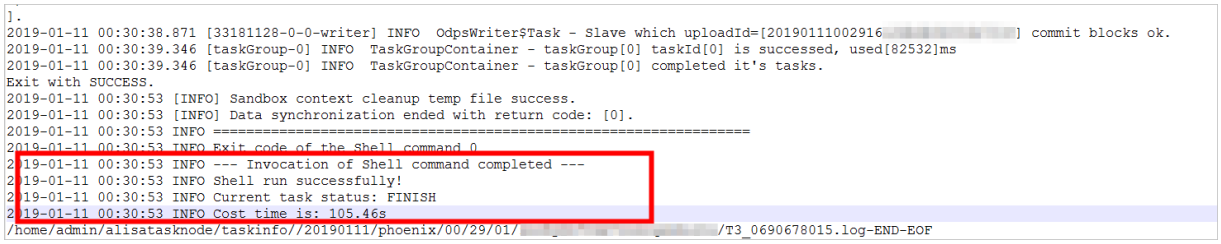
> **说明** MaxCompute表设计规范详情请参见表设计规范。

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。如果您使用的是包年包月方式购买的MaxCompute项目资源，还需要记录购买数。

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

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**页面右键查看用户任务运行时间，如下图所示。



记录占用存储大小

登录DataWorks控制台。

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。

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

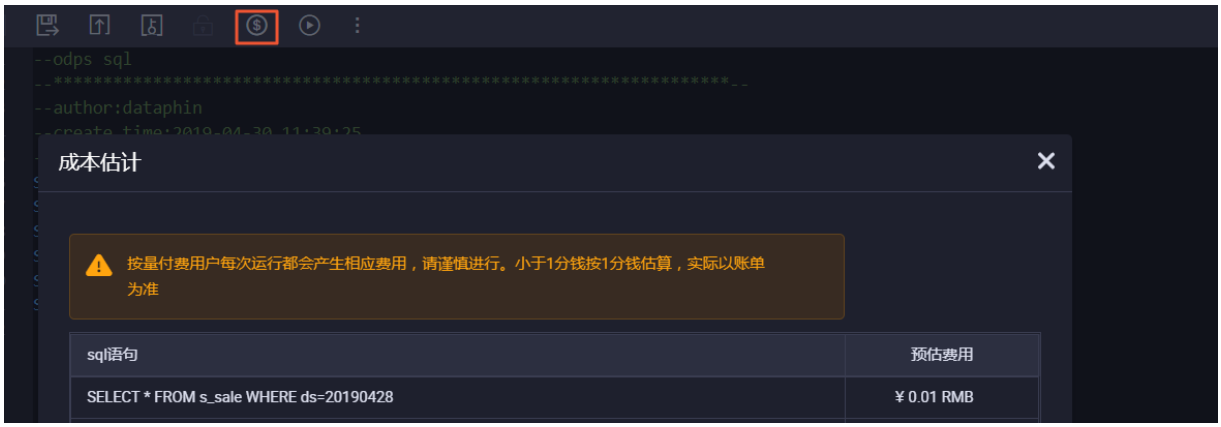
运行日志

```
+------------------------------------------------------------------------------------+
| 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                                     |
+------------------------------------------------------------------------------------+
```

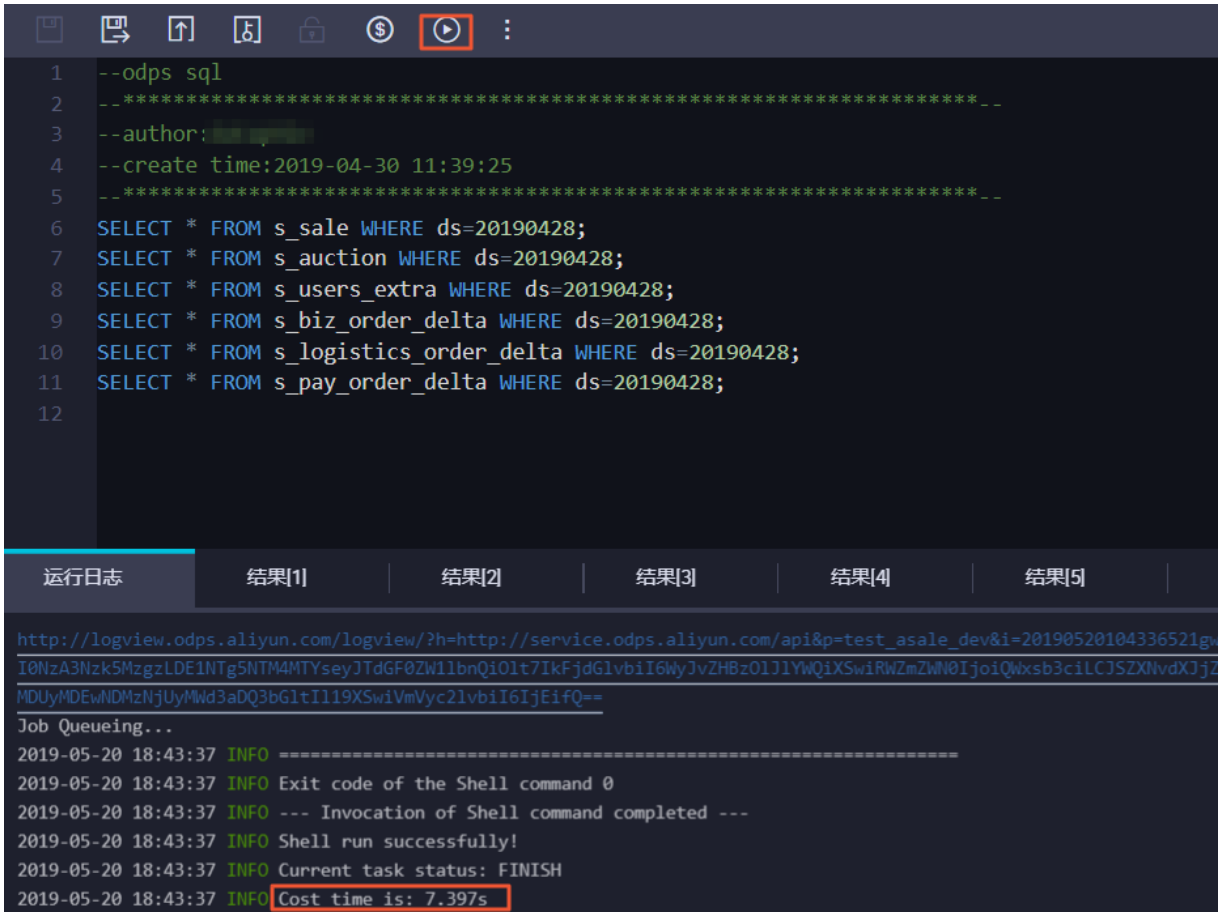
记录查询执行时间及预估费用

登录DataWorks控制台，进入**数据开发**页面，创建**ODPS sql**节点。

您可以在运行任务时或通过单击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表操作。

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，您可以将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表的高级功能。

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，性能基准详情请参见建立性能基准。

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。本教程帮助您了解并操作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

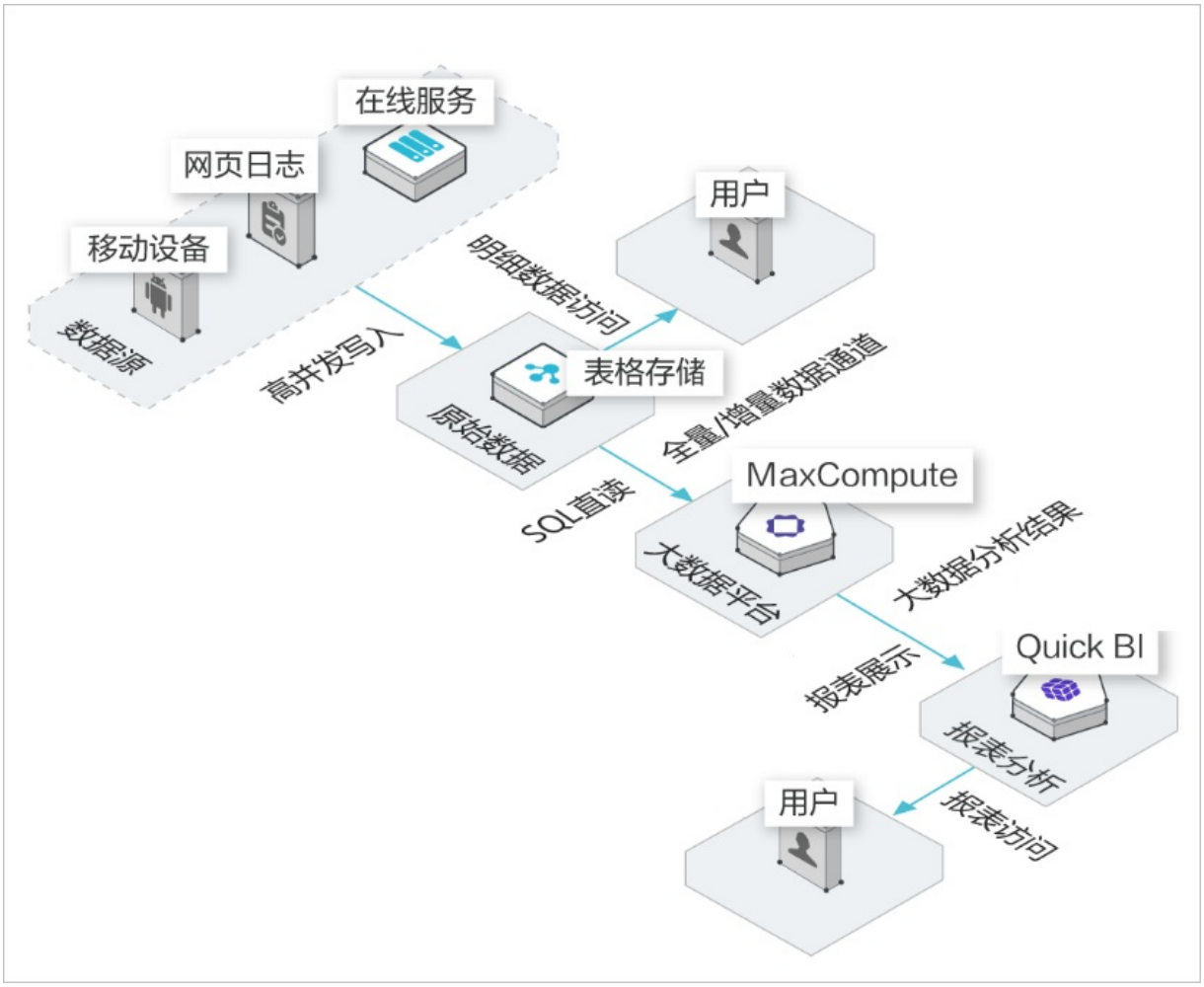
本文的示例基于真实的网站日志数据集，数据来源于某网站上的HTTP访问日志数据。基于这份网站日志，您可以实现如下分析需求：

- 统计并展现网站的PV和UV，并能够按照用户的终端类型（例如，Android、iPad、iPhone和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（UV）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：数据可视化展示。

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

本文为您介绍开始本教程前的环境准备工作，需要开通表格存储（Tablestore）、大数据计算服务（MaxCompute）、一站式大数据开发治理平台（DataWorks）和数据可视化分析平台（Quick BI）。

前提条件

- 已注册阿里云账号。如果您还没有注册阿里云账号，请进入阿里云官网，单击**立即注册**，即可进入阿里云账号注册页面创建新的阿里云账号。
- 已实名认证。如果您还没有实名认证，请进入实名认证页面对账号进行实名认证。

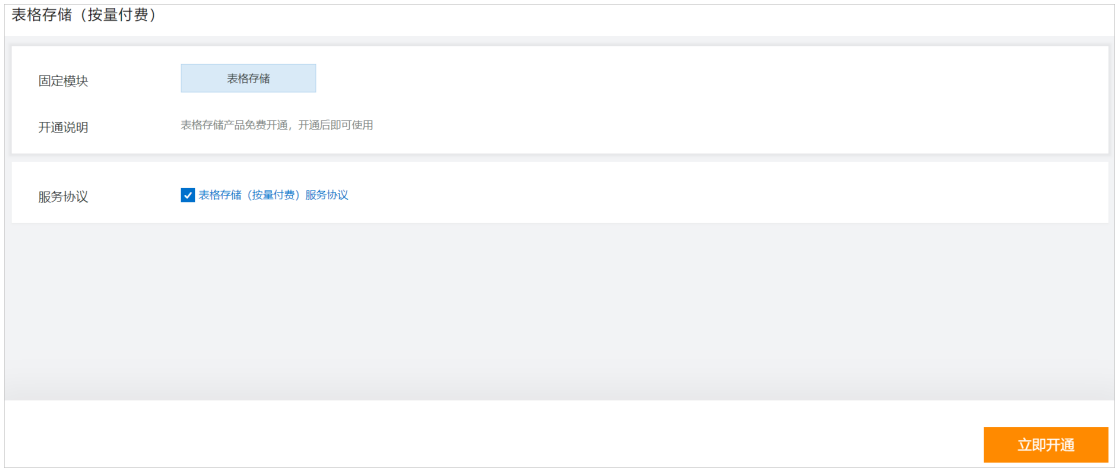
背景信息

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

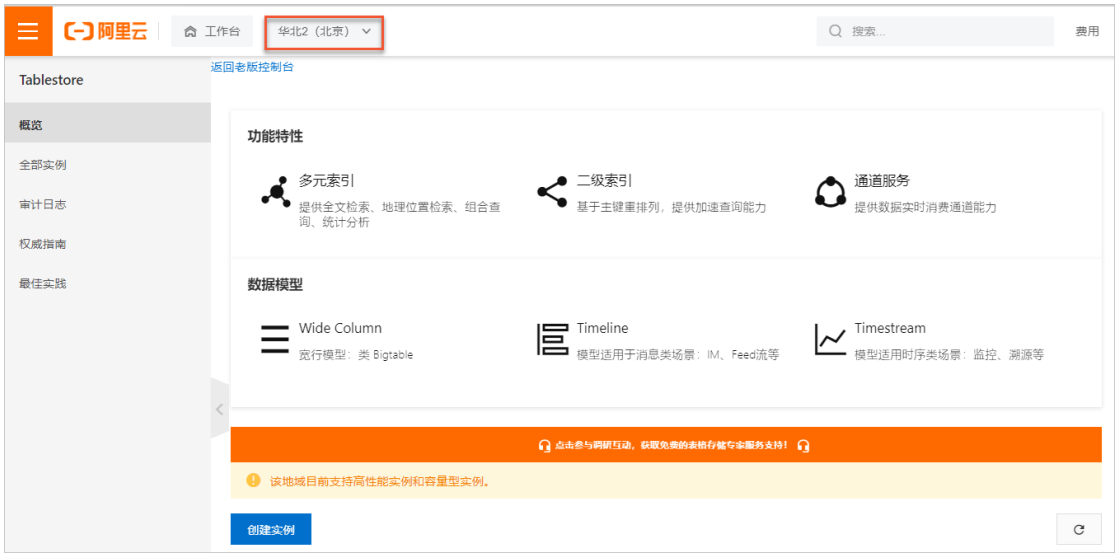
操作步骤

1. 创建表格存储实例。
   i. 进入表格存储Tablestore产品详情页，单击**立即开通**。
   ii. 在**表格存储（按量付费）**页面，勾选**表格存储（按量付费）服务协议**并单击**立即开通**。



   iii. 单击**管理控制台**。

iv. 在**管理控制台**页面的**概述**页签，选择地域为**华北2（北京）**，单击**创建实例**。



> **说明** 在本教程中，表格存储服务选择华北2（北京）。您可以根据需要选择其他地域。

v. 在**购买方式**对话框选择**按量模式**，填写**实例名称**，**实例规格**请选择**容量型实例**，单击**确定**。

> **说明**
> - 实例名称在表格存储同一个地域内必须全局唯一，建议您选用自己可辨识且符合规则的名称。
> - 实例名称在MaxCompute数据处理中也会被使用，本例中为workshop-bj-mc，关于实例的详细解释请参见实例。

vi. 完成创建后，单击左侧导航栏**全部实例**可以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。
   i. 进入MaxCompute产品详情页，单击**立即购买**。
   ii. 选择**按量计费**，选择地域为**华东2（上海）**，**规格类型**为默认的**标准版**，单击**立即购买**。

> **说明** MaxCompute地域与表格存储地域相同可以节省您的流量费用，因此您可以选择地域为**华北2（北京）**。本教程中MaxCompute地域选择为**华东2（上海）**，以便为您展示跨地域的外部表使用过程。

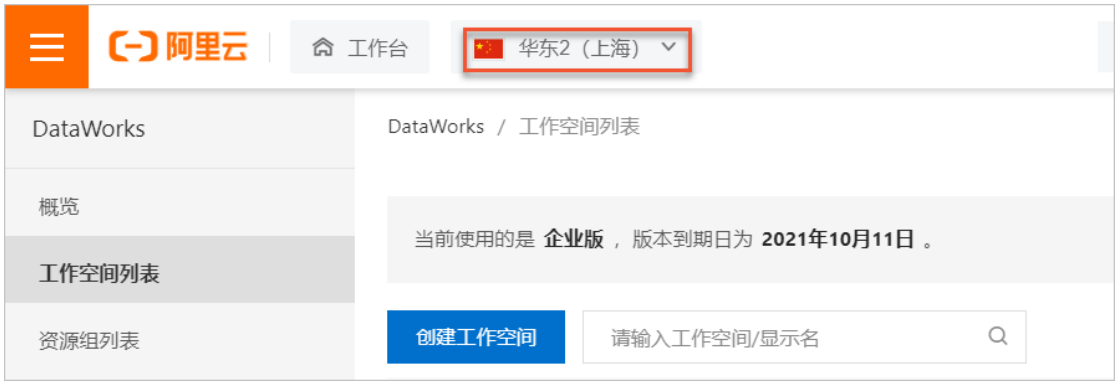
3. 开通DataWorks。
   i. 进入DataWorks产品详情页，单击**立即购买**。

ii. 选择地域为**华东2（上海）**，单击**立即购买**。

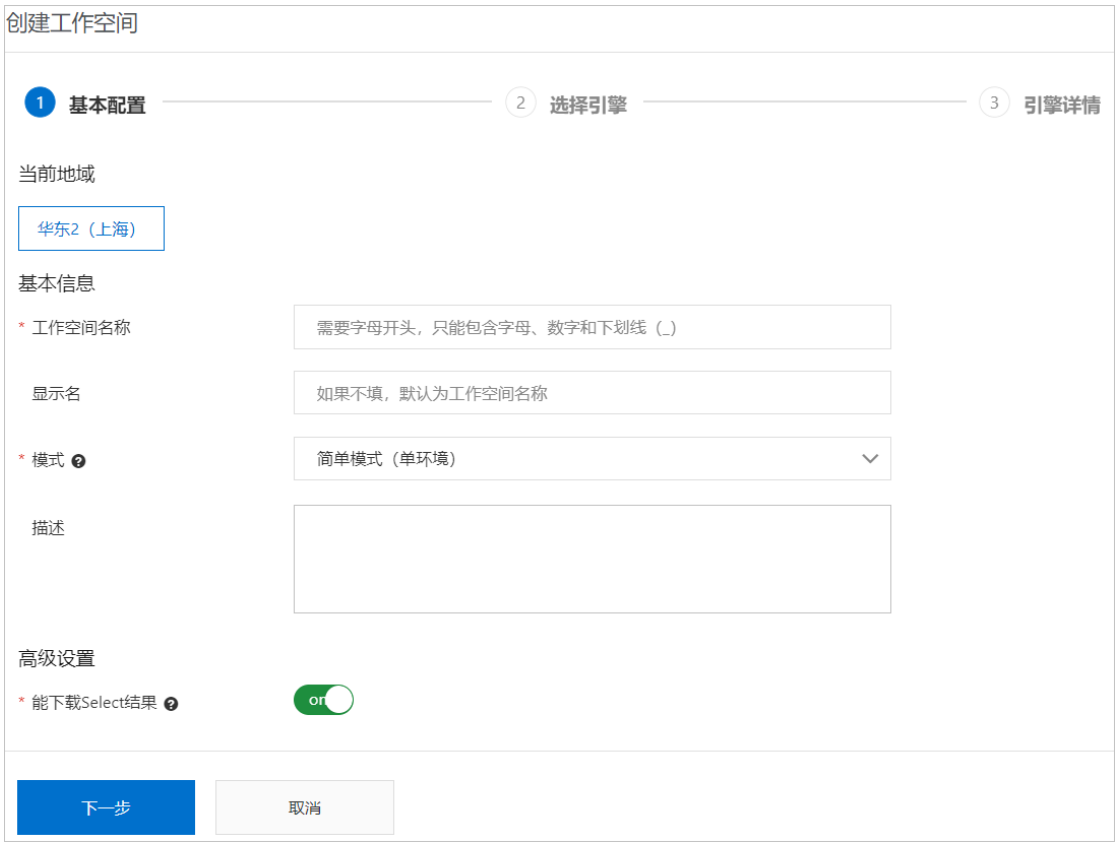
> **说明** MaxCompute地域与表格存储地域相同可以节省您的流量费用，因此您也可以选择地域为**华北2（北京）**。本教程中MaxCompute地域选择为**华东2（上海）**，以便为您展示跨地域访问数据的使用过程。

4. 创建DataWorks工作空间。

i. 进入DataWorks工作空间列表，选择地域为**华东2（上海）**，单击**创建工作空间**。



ii. 在**创建工作空间**面板中，填写**基本配置**相关内容，单击**下一步**。

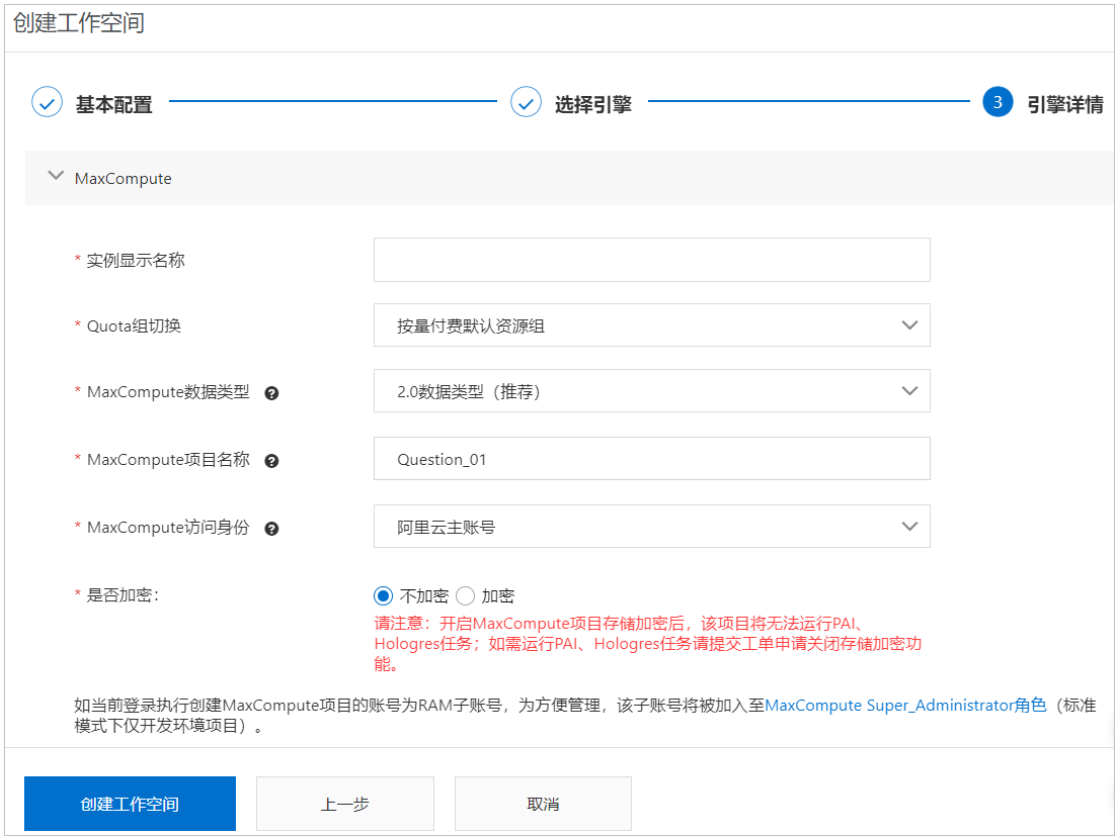


> **说明**
>
> - 工作空间名称全局唯一，建议您使用易于区分的名称。
> - 为方便使用，本教程中DataWorks工作空间模式为**简单模式（单环境）**。在简单模式下，DataWorks工作空间与MaxCompute项目一一对应，详情请参见简单模式和标准模式的区别。

iii. 进入**选择引擎**界面，选择**计算引擎服务**下的**MaxCompute**，**按量付费**，单击**下一步**。

iv. 进入**引擎详情**页面，填写选购引擎的配置，单击**创建工作空间**。

工作空间创建成功后，即可在**工作空间列表**页面查看相应内容。



分类	配置	说明
MaxCompute	**实例显示名称**	实例名称长度需要在3-28个字符，仅支持字母开头，仅包含字母、下划线和数字。
	Quota组切换	Quota用来实现计算资源和磁盘配额。
	MaxCompute数据类型	MaxCompute项目的数据类型版本。
	MaxCompute项目名称	默认与DataWorks工作空间的名称一致。
	MaxCompute访问身份	MaxCompute访问身份包括**阿里云主账号**和**任务责任人**。**阿里云主账号**即阿里云账号
	是否加密	根据实际情况选择是否需要加密当前实例。

5. 开通Quick BI。详情请参见Quick BI购买、升级、降级、续费、欠费。

> **说明** Quick BI免费试用信息，请参见30天免费试用说明。

2.3. 数据准备

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

前提条件

- 创建华北2（北京）区域的表格存储实例，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，即可获得实例访问地址。对于跨区域的访问，建议您使用公网地址。详细操作请参见环境准备。

- 使用阿里云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，解压下载包，编辑conf文件夹内的app.conf文件。

   app.conf文件内容示例如下。

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

   其中，需要配置的参数如下：

   - endpoint：表格存储实例的访问网络地址，建议您使用公网地址。您可以在Tablestore控制台，单击实例名称，在**实例详情**页签的**实例访问地址**区域获取。
   - instanceName：表格存储实例的名称。您可以在Tablestore控制台的概览页面获取。
   - accessKeyId和accessKeySecret：访问阿里云的密钥。

3. 启动Demo准备测试数据。

i. 启动Windows CMD命令行工具，进入您解压缩Demo包的路径，执行如下语句查看Demo包命令用法。

```
workshop_demo.exe -h
```

该命令会列出该demo的相关命令，如下。

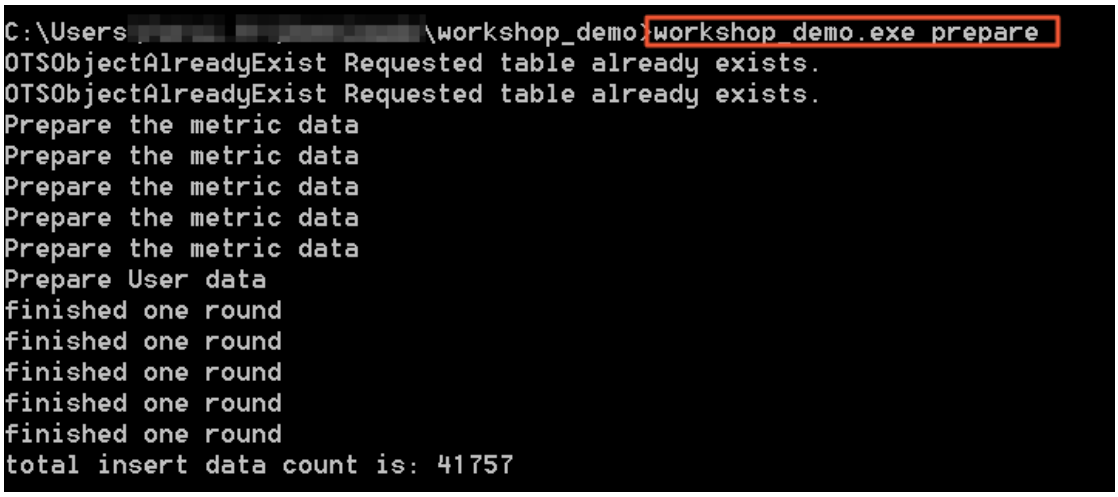
```
...\workshop_demo>workshop_demo.exe -h
prepare will prepare the data
raw 00005 "2017-12-19" will query user data by user id and time
new/day_active/month_active/day_pv/month_pv will query metrics data
```

- `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`：`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`：在结果表中按照如下几种类型查询报表数据：
  - 新增：new
  - 日活：day_active
  - 月活：month_active
  - 日PV：day_pv
  - 月PV：month_pv

ii. 执行如下命令生成准备数据。

```
...\workshop_demo>workshop_demo.exe prepare
```

结果如下。



在此过程中，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。

列名	类型	说明
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。
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

     通过如下语句查询指定用户在某一日期指定条数的明细数据。表格数据的日期对应于您创建表格的时间。

     ```
     raw ${userid} ${date} ${Top条数}
     ```

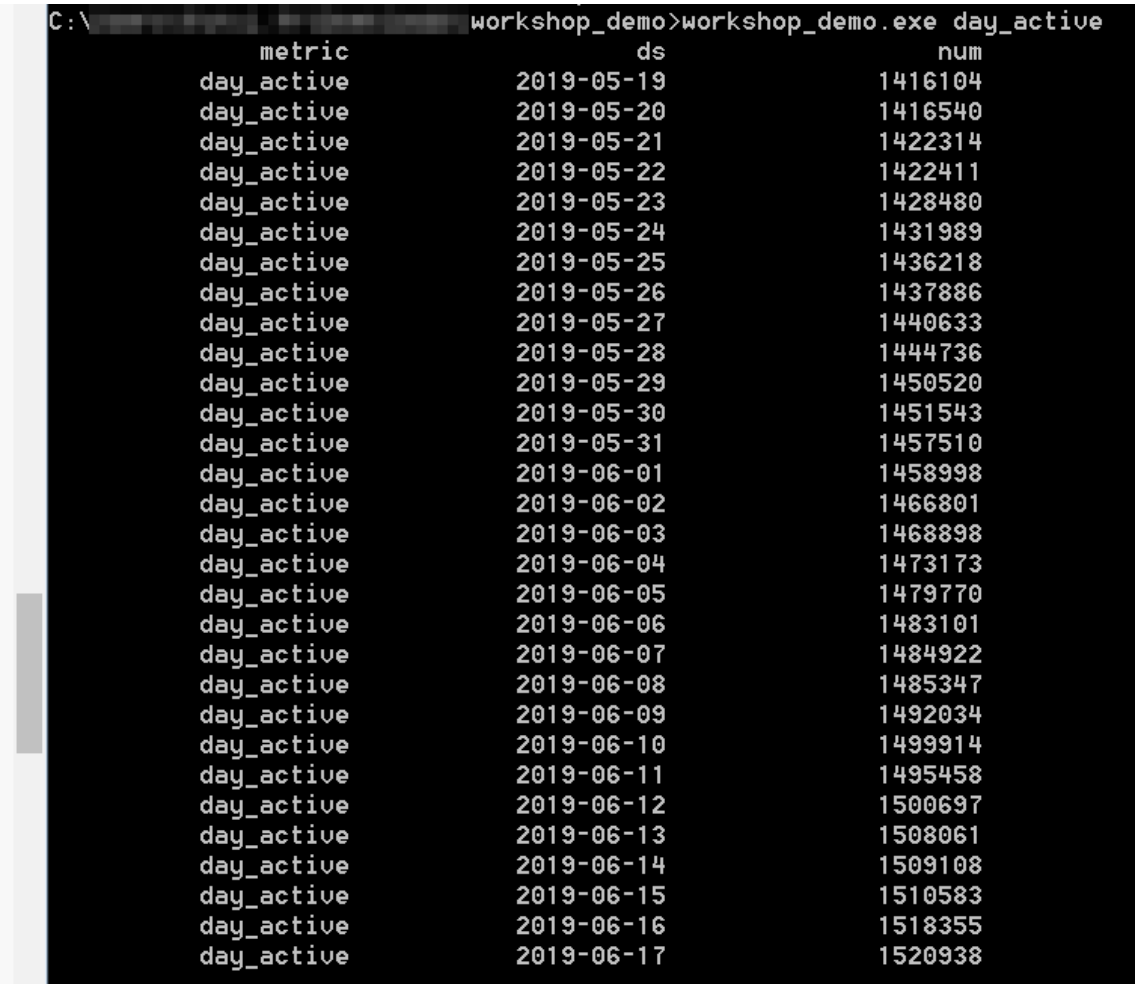
其中，${userid}为用户ID，${date}为指定日期，${Top条数}为指定查询条数。例如，您创建数据时间为2019年6月15日，则可以使用 `workshop_demo.exe raw 00010 "2019-06-15" 20` 查看20条用户明细数据。

```
C                    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、内容进行对齐显示。

- 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:\[redacted]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

本文为您介绍如何在MaxCompute上建立数据表，用于存储原始数据及加工后的数据。

前提条件

- 已开通MaxCompute服务并创建DataWorks工作空间（本教程使用为简单模式工作空间），详情请参见环境准备。
- 已具备访问Tablestore数据的权限。当MaxCompute和Tablestore的所有者是同一个账号时，您可以单击此处一键授权。如果不是，您可以自定义授权，详情请参见OTS外部表。

操作步骤

1. 进入DataWorks数据开发界面。
   i. 进入DataWorks工作空间列表，选择区域为**华东2（上海）**。
   ii. 单击已创建好的工作空间后的**进入数据开发**，进入工作空间的数据开发界面。
2. 新建业务流程。

i. 右键单击**业务流程**，选择**新建业务流程**。

ii. 填写**业务名称**和**描述**，单击**新建**。本教程中，业务流程名为Workshop。

3. 新建数据表。

i. 创建外部表ots_user_trace_log。

> **注意** 创建外部表时，暂时不支持在向导模式下使用DDL方式创建。您可以在向导模式下手动添加外部表字段进行表配置，或创建一个MaxCompute的ODPS SQL节点，在ODPS SQL节点中使用DDL语句创建外部表。以下以使用ODPS SQL节点的DDL语句创建外部表作为操作示例。

a. 单击新建的业务流程Workshop，右键单击**MaxCompute**，选择**新建 > ODPS SQL**，输入节点名称**workshop**单击**提交**。

b. 双击创建的ODPS节点，进入数据开发界面后，编译如下DDL语句，用于创建名称为ots_user_trace_log的外部表。

外部表ots_user_trace_log的建表语句如下。

```
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/评论',
    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/';
```

- STORED BY：必选参数，值为 `com.aliyun.odps.TableStoreStorageHandler` ，是MaxCompute内置处理Tablestore数据的StorageHandler，定义了MaxCompute和Tablestore的交互。
- SERDEPROPERITES：必选参数，是提供参数选项的接口，在使用TableStoreStorageHandler时，以下选项必须指定：
  - tablestore.columns.mapping：用于描述MaxCompute将访问的Tablestore表的列，包括主键和属性列。

    > **说明**
    > - 以冒号（:）开头的参数值为Tablestore主键，例如示例中的 `:md5` 、 `:uid` 和 `:ts` ，其它参数值均为属性列。
    > - 在指定映射时，您必须提供指定Tablestore表的所有主键，只需提供需要通过MaxCompute访问的属性列。提供的属性列必须是Tablestore表的列，否则即使外表可以创建成功，查询时也会报错。

  - tablestore.table.name：需要访问的Tablestore表名。如果指定的Tablestore表名错误（不存在），则会报错，MaxCompute不会主动创建Tablestore表。
- LOCATION：用来指定Tablestore的访问地址。请您根据环境准备，将自己的表格存储实例访问地址参数填写在此。

  > **说明** 如果您使用公网地址LOCATION 'tablestore://workshop-bj-001.cn-beijing.ots.aliyuncs.com/'报错，显示网络不同，可尝试更换为经典网地址LOCATION 'tablestore://workshop-bj-001.cn-beijing.ots-internal.aliyuncs.com/'。

c. 单击保存图标后，单击运行图标。

当运行日志显示运行成功后，外部表即已创建成功。

ii. 创建ods_user_trace_log表。

建表方法同上，建表语句如下。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/评论',
    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
);
```

iii. 创建dw_user_trace_log表。

建表方法同上，建表语句如下。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/评论',
    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
);
```

iv. 创建rpt_user_trace_log表。

建表方法同上，建表语句如下。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/评论',
    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
);
```

v. （可选）如果您使用的是标准模式，上述步骤创建的外部表仅提交到了开发环境，您需根据以下步骤，依次将所有创建的外部表提交到生产环境。

a. 单击MaxCompute下的表，右键选择导入表。

b. 在导入表弹窗中选择上述创建的开发环境外部表，单击**确认**。

> **说明** 鼠标悬浮在表名上即可在表名后看到该表是生产环境的表还是开发环境的表。如果误选了生产环境的表，会提示您表不存在。

c. 双击导入的外部表，单击**提交到生产环境**。

4. 验证建表结果。

i. 完成建表后，您可以在Workshop业务流程**MaxCompute > 表**下看到新建的4张表。

ii. 右键单击业务流程中**MaxCompute**下的**数据开发**，选择**新建 > ODPS SQL**。

iii. 在**新建节点**页面，输入**节点名称**，单击**提交**新建ODPS SQL节点。

iv. 在新建的ODPS SQL节点中输入如下SQL语句，单击图标。

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

返回表的结构信息如下：

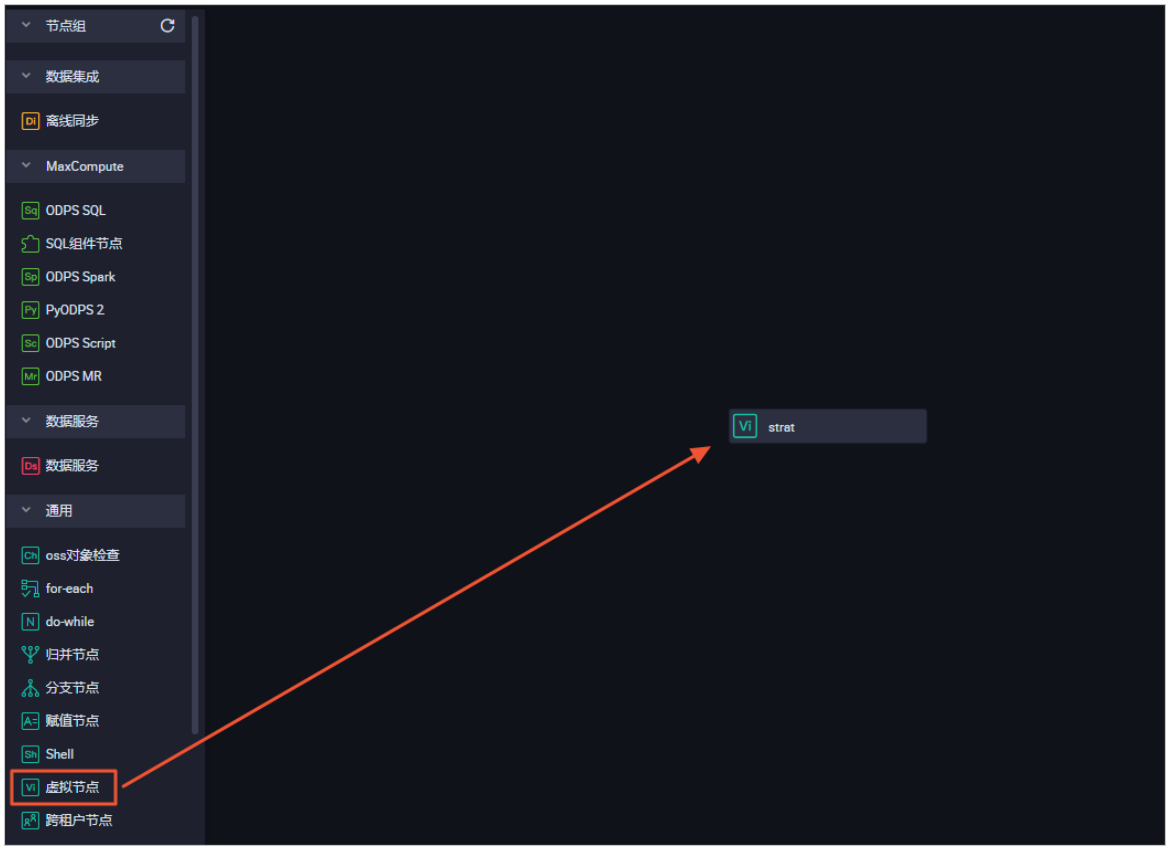
```
+------------------------------------------------------------------------------------+
| Owner: [redacted] | Project: [redacted]                                            |
| TableComment:                                                                      |
+------------------------------------------------------------------------------------+
| CreateTime:              2020-06-16 18:56:46                                       |
| LastDDLTime:             2020-06-16 18:56:46                                       |
| LastModifiedTime:        2020-06-16 18:56:46                                       |
+------------------------------------------------------------------------------------+
| InternalTable: YES      | Size: 0                                                  |
+------------------------------------------------------------------------------------+
| Native Columns:                                                                    |
+------------------------------------------------------------------------------------+
| Field           | Type       | Label | Comment                                     |
+------------------------------------------------------------------------------------+
| country         | string     |       | 国家                                        |
| province        | string     |       | 省份                                        |
| city            | string     |       | 城市                                        |
| device_brand    | string     |       | 设备品牌                                    |
| device          | string     |       | 终端型号                                    |
| system_type     | string     |       | 系统类型，Android、IOS、ipad、Windows_phone |
| 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 |
| customize_event_content | string     |       | 用户关注内容信息，在customize_event为浏览和评论时 包含该列 |
| pv              | bigint     |       | 浏览量                                      |
| uv              | bigint     |       | 独立访客                                    |
+------------------------------------------------------------------------------------+
| Partition Columns:                                                                 |
+------------------------------------------------------------------------------------+
| dt              | string     |                                                     |
+------------------------------------------------------------------------------------+
OK
2020-06-16 19:56:10 INFO ================================================================
2020-06-16 19:56:10 INFO Exit code of the Shell command 0
2020-06-16 19:56:10 INFO --- Invocation of Shell command completed ---
```

2.4.2. 设计工作流

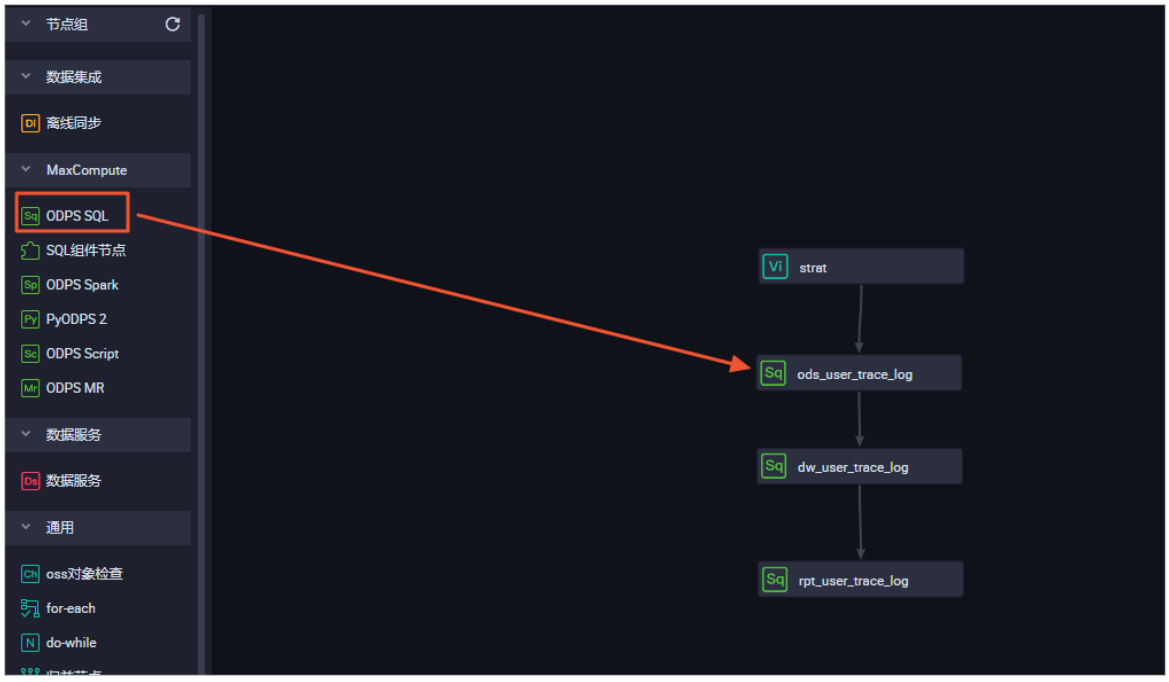
通过设计工作流，您可以明确在整体数据开发过程中各任务节点的排布。对于本教程中这种较为简单的单数据流场景，您可以选择每个数据表（数仓层次）对应一个工作流。

操作步骤

1. 进入DataWorks数据开发界面。
   i. 进入DataWorks工作空间列表，选择区域为**华东2（上海）**。
   ii. 单击已创建好的工作空间后的**进入数据开发**，进入工作空间的数据开发界面。
2. 在您创建的业务流程上双击，打开画布面板。
3. 向画布中拖入1个虚拟节点，命名为start。



4. 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语句。

前提条件

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，详细请参见注册自定义函数。

注册自定义函数

1. 添加资源
   i. 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。

      关于IP地址转换的自定义函数，详情请参见MaxCompute中实现IP地址归属地转换。

   ii. 右键单击WorkShop业务流程下的**MaxCompute**，选择**新建 > 资源**。需要分别新建**File**和**JAR**类型的资源。

- File类型上传地址库ip.dat。
  a. 输入**资源名称**，选中**大文件（内容超过500KB）**及**上传为ODPS资源**，然后单击**点击上传**。

  b. 单击**提交**。

- JAR类型对应Java包getaddr.jar。
  a. 您需要勾选**上传为ODPS资源**，然后单击**点击上传**。

  b. 上传完成后，单击**提交**。

> **说明** 提交时，请忽略血缘不一致信息。

2. 注册函数
   i. 在业务流程下右键单击**MaxCompute**，选择**新建 > 函数**，将函数命名为getregion。
   ii. 在**注册函数**页面，依次填写类名为odps.test.GetAddr，资源列表为getaddr.jar,ip.dat，命令格式为getregion(ip string)，保存后单击  提交函数注册。

配置节点

1. 配置虚拟节点start。
    i. 双击start节点，进入节点配置页面。

ii. 单击右侧的**调度配置**，在**调度依赖**区域下单击**使用工作空间根节点**完成配置。

iii. 在时间属性区域选择**重跑属性**为**运行成功或失败后皆可重跑**。

iv. 单击 按钮，完成节点提交。

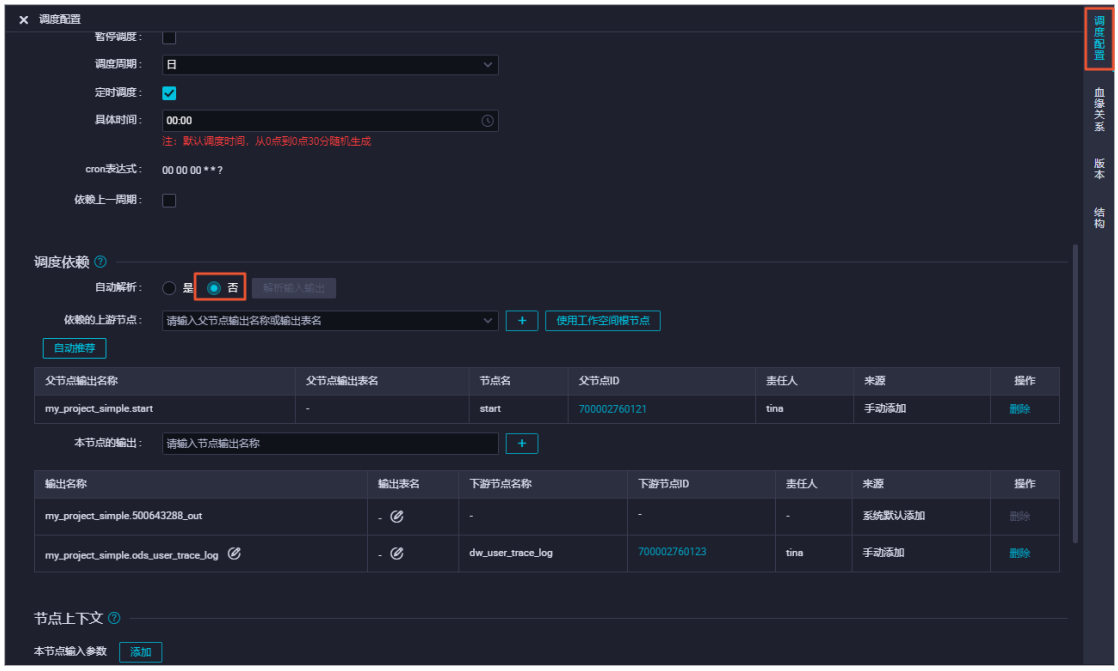
2. 配置ODPS SQL节点ods_user_trace_log

i. 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}释义请参见配置调度参数。

ii. 完成代码编写后，单击右侧的**调度配置**，选择**自动解析**为否。

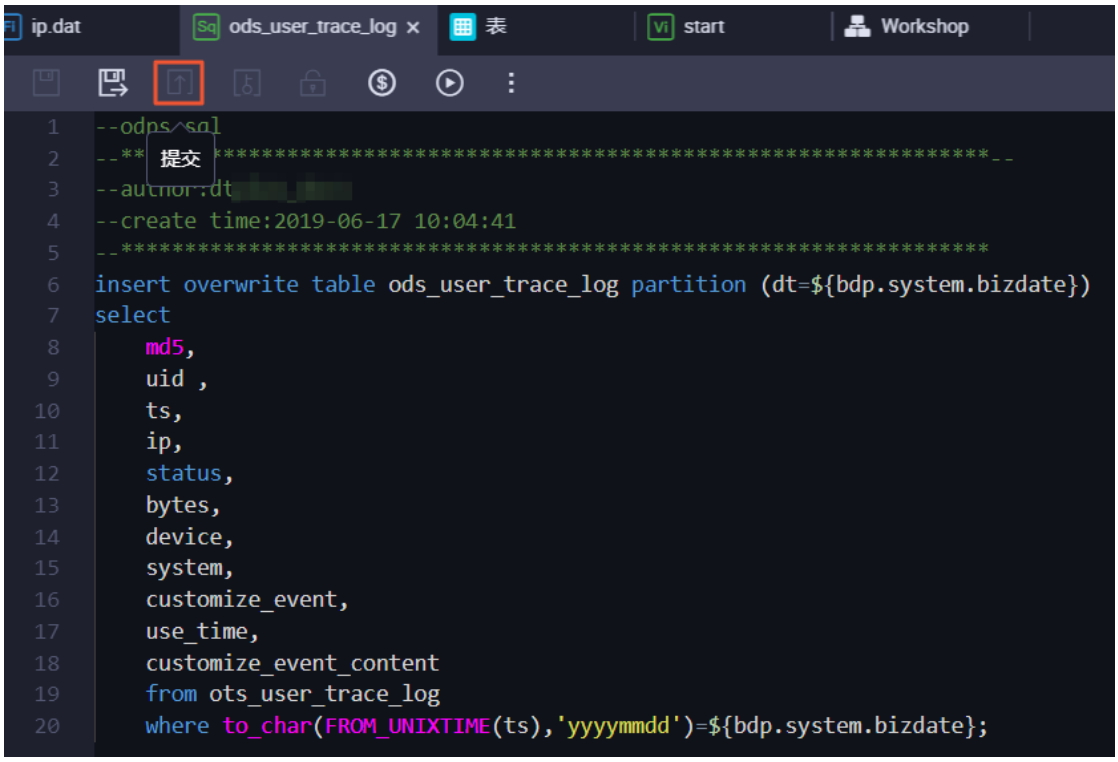


iii. 手动删除错误的依赖关系。

iv. 按照业务流程顺序搜索正确的上游节点，例如此处为start，并单击**添加**。

v. 在时间属性区域选择**重跑属性**为**运行成功或失败后皆可重跑**。

vi. 完成后，单击**提交**。



3. 配置ODPS SQL节点dw_user_trace_log

   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};
```

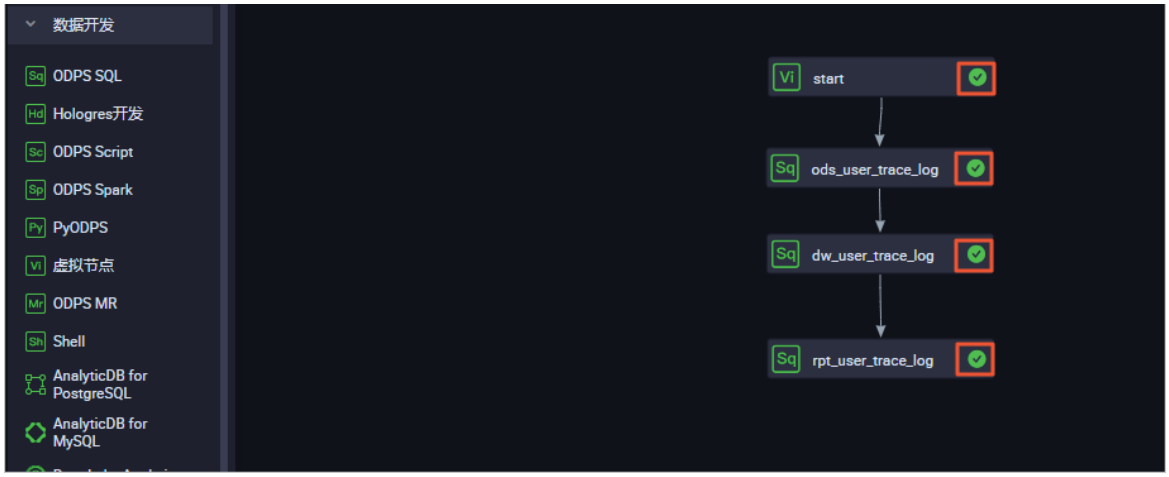
4. 配置ODPS SQL节点rpt_user_trace_log

   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;
```

5. 验证配置结果。

   双击业务流Workshop，打开画布面板。单击  按钮。运行成功如下图所示。



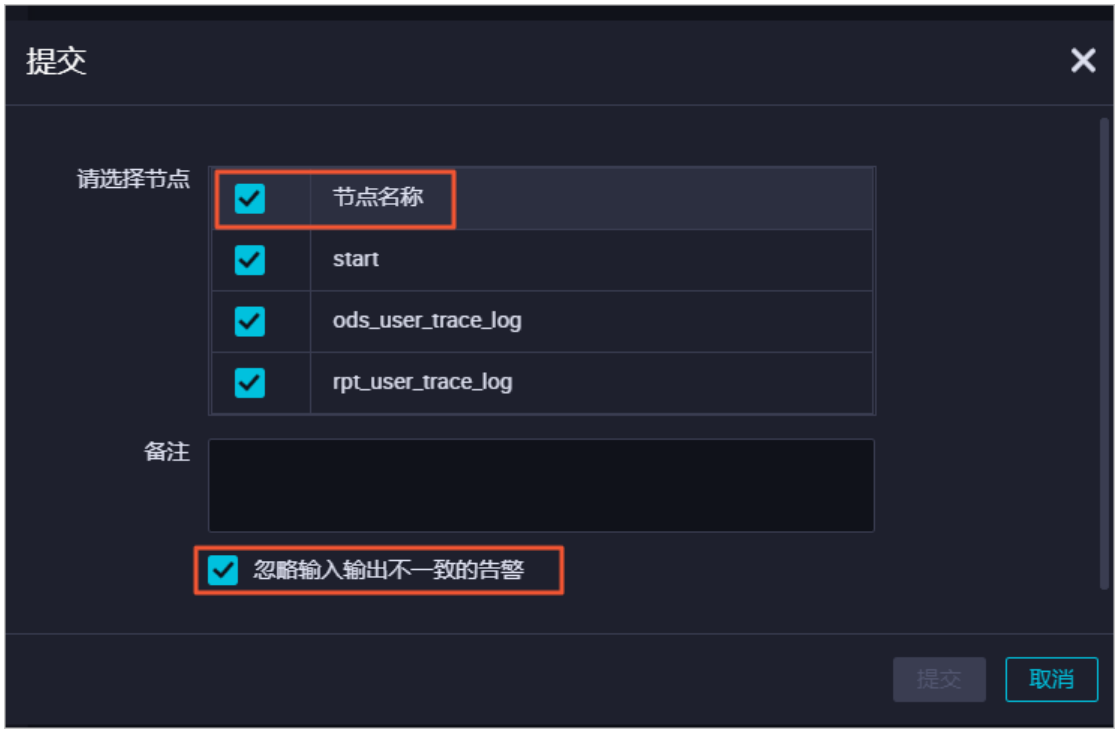
   如果运行状态异常，请右键单击出错节点，单击**查看运行日志**进行排查。

2.4.4. 任务提交与测试

您完成节点配置后，需要将任务提交到运维中心进行测试。

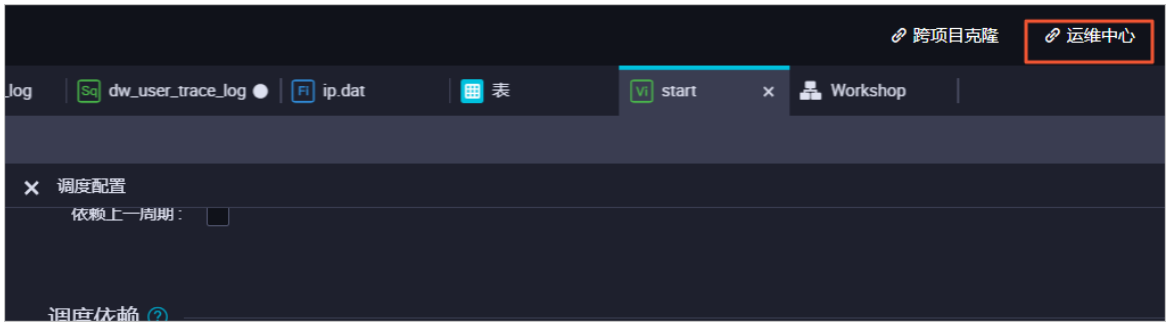
操作步骤

1. （可选）提交业务流程。如果您的节点在配置完成后已经提交完毕且无更新，请跳过本步骤。
   i. 双击业务流程名称Workshop，单击图标。
   ii. 勾选所有可提交节点及**忽略输入输出不一致的告警**，单击**提交**。



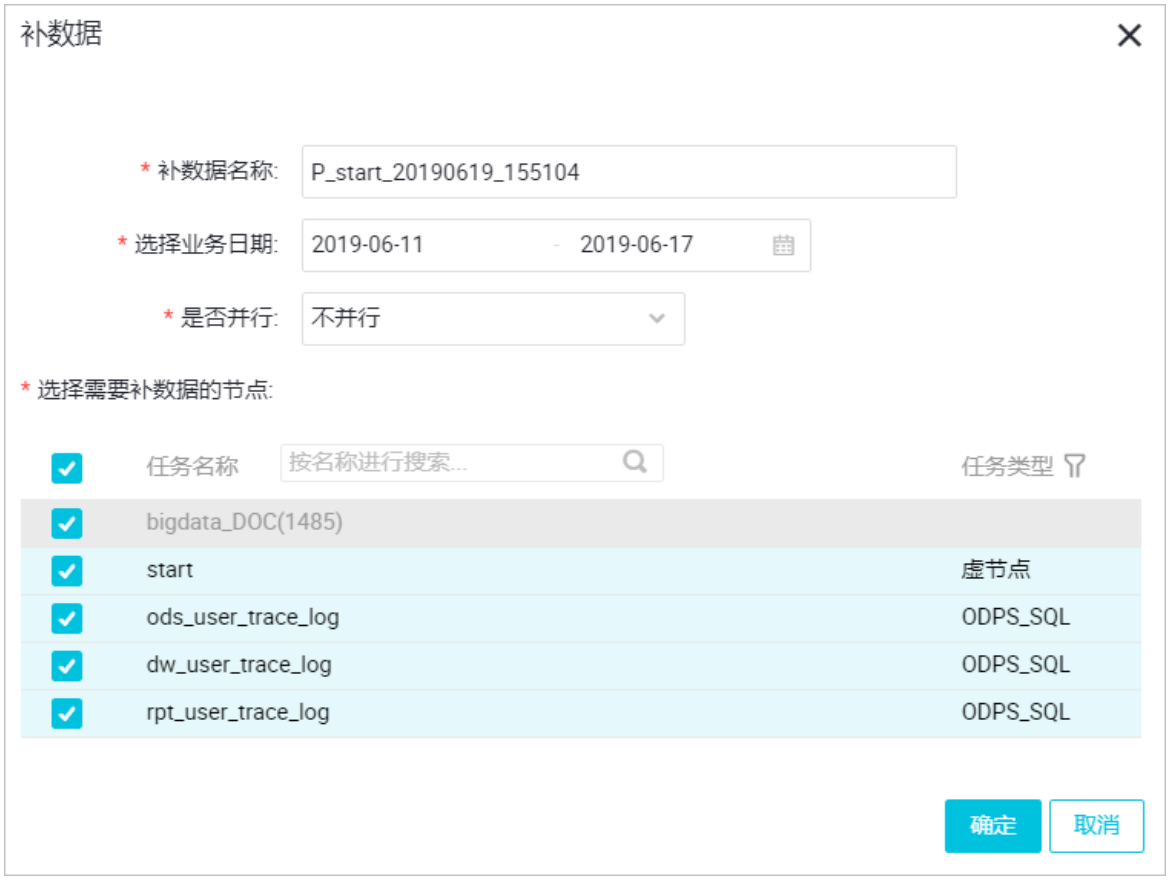
> **说明** 标准空间模式下，提交通过后，需要单击**发布**将任务发布至生产环境。

2. 单击右上角的**运维中心**。



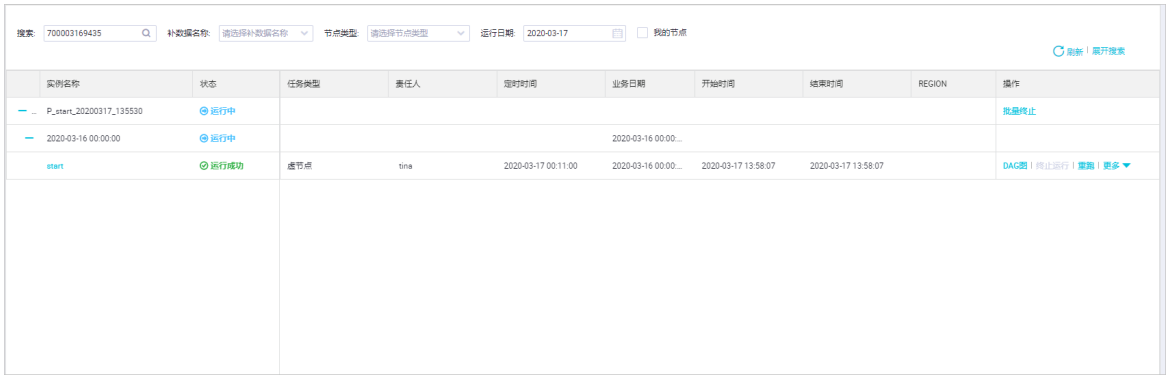
3. 在左侧导航栏，单击**周期任务运维 > 周期任务**，双击节点列表中的虚拟节点start。
4. 在右侧流程图上，右键单击虚拟节点start，选择**补数据 > 当前节点及下游节点**。

5. 在**补数据**页面，选中所有需要补数据的节点，选择业务日期为过去一周，单击**确定**。



> **说明** 关于补数据实例的详情请参见执行补数据并管理补数据实例。

6. 在左侧导航栏，单击**补数据实例**，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，右键单击出错节点，选择**查看运行日志**进行排查。

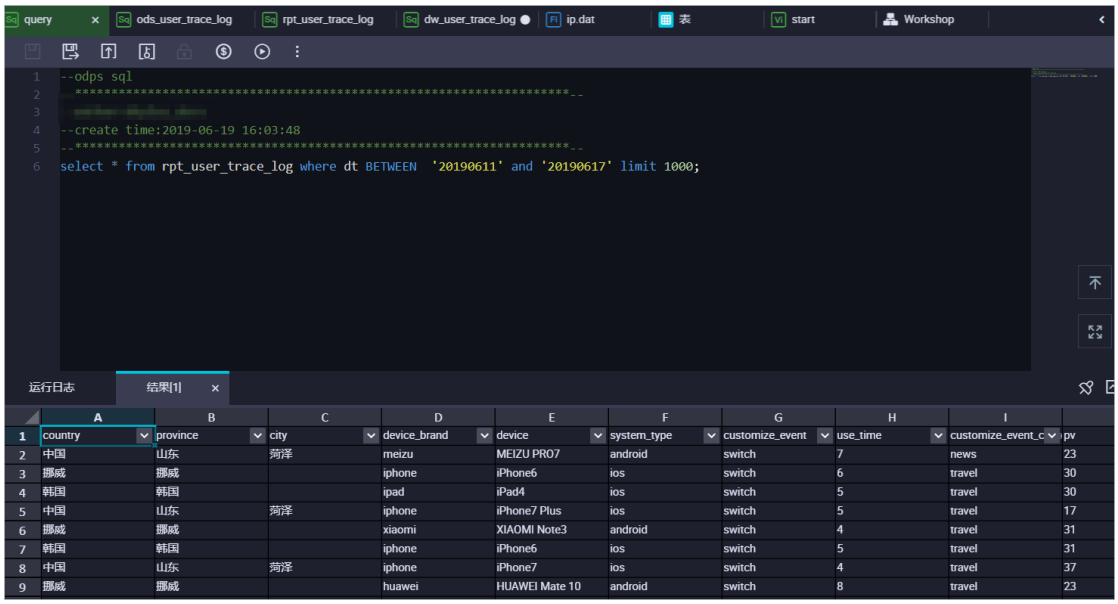
7. 补数据实例运行完成后，验证结果。
   i. 在左侧导航栏，单击业务流程**Workshop > MaxCompute**，右键单击**数据开发**，选择**新建 > ODPS SQL**，新建名为query的SQL节点。

ii. 输入如下SQL语句，查询2019年6月11日到2019年6月17日之间表rpt_user_trace_log中的数据，确认数据是否成功写入rpt_user_trace_log表。

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

iii. 单击图标。

查询结果如下。



	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. 数据可视化展示

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

在开始前，请确认您已经完成了环境准备、数据准备、数据建模与开发等全部步骤。

背景信息

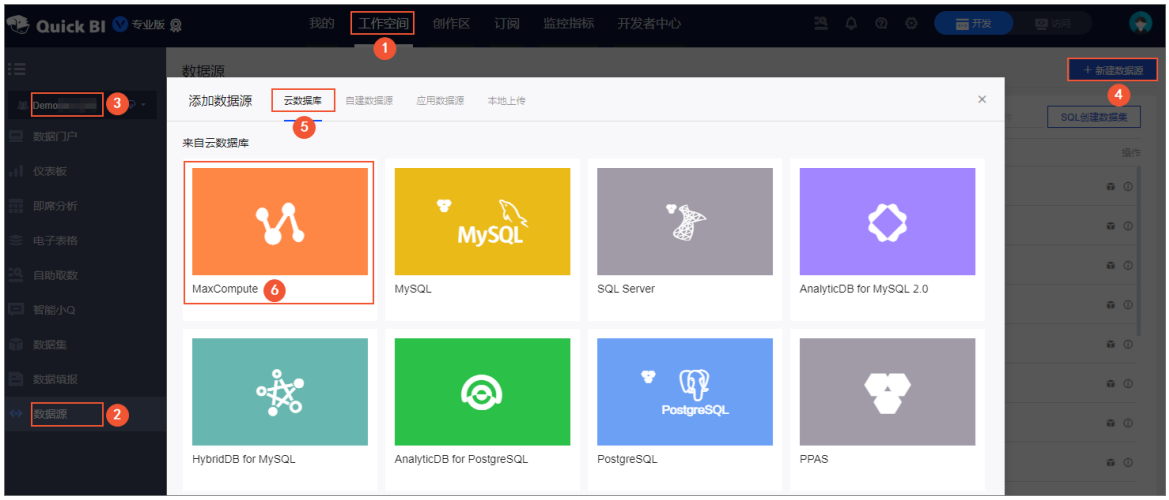
rpt_user_trace_log表包含了country、province、city、device_brand、use_time、pv等字段信息。您可以通过以下步骤制作仪表板，用以展示用户的核心指标、周期变化、用户地区分布和记录。

- 步骤一：连接数据源。
- 步骤二：创建数据集。
- 步骤三：可视化展示。

步骤一：连接数据源

1. 使用阿里云账号登录Quick BI控制台。
2. 在Quick BI首页，选择目标工作空间。

   本例中以默认空间为例介绍。
3. 在工作台页面，按照下图指引，连接数据源。



4. 在**添加MaxCompute数据源**对话框，配置数据源连接参数并测试连通性。

   添加成功后，您可以在数据源列表中，看到您创建的数据源。

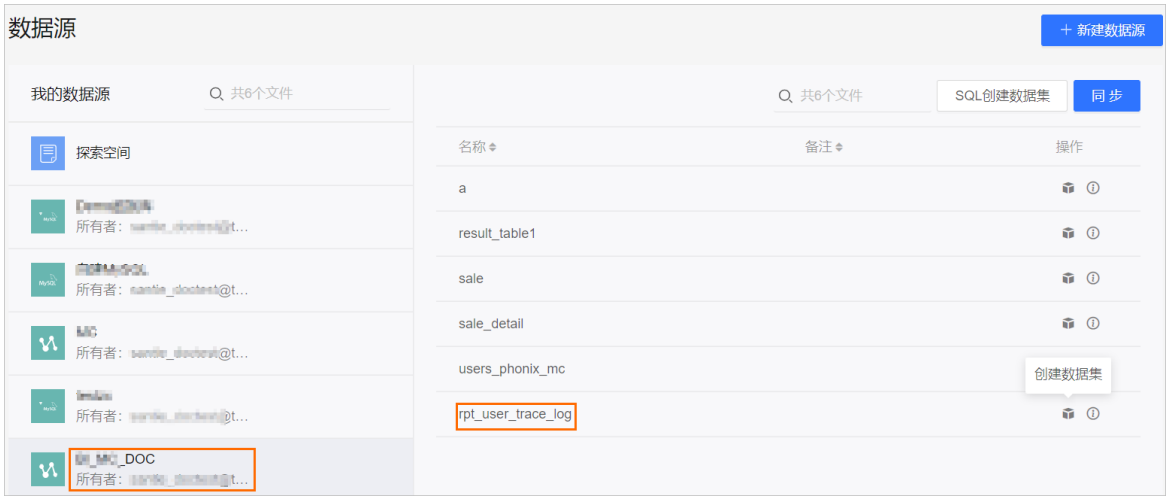
参数名	参数说明
显示名称	数据源配置列表的显示名称。名称只能由中英文、数字及下划线（_）、斜线（/）、反斜线（\）、竖线（|）、小括号（()）、中括号（[]）组成，不超过50个字符。
数据库地址	此处有默认地址，通常无需修改。 **说明** 数据库地址根据Region不同而变化，详细对应信息请参见Endpoint。
项目名称	MaxCompute项目名称。您可以登录MaxCompute控制台，在**项目管理**页签查看具体的项目名称。

参数名	参数说明
AccessKey ID	阿里云账号或RAM用户的AccessKey ID。您可以进入AccessKey管理页面获取AccessKey ID。
AccessKey Secret	AccessKey ID对应的AccessKey Secret。您可以进入AccessKey管理页面获取AccessKey Secret。

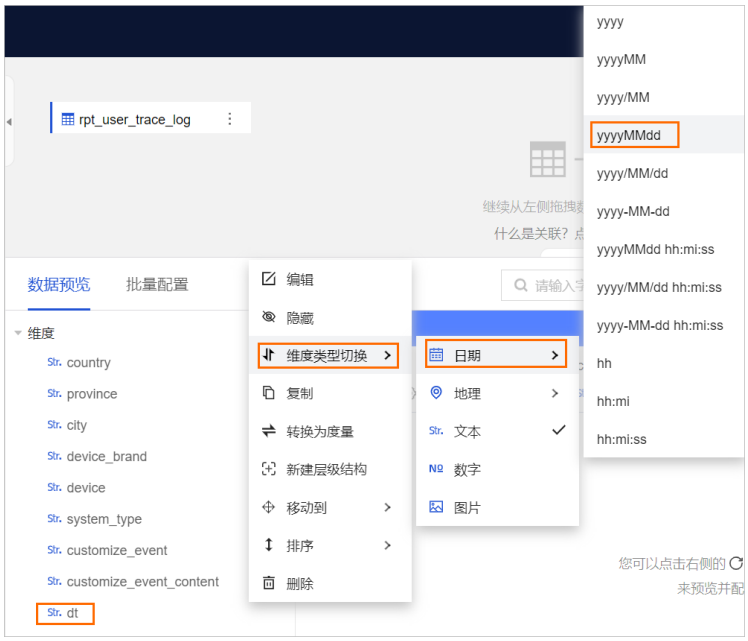
步骤二：创建数据集

Quick BI中数据集是可视化分析的基础，您可以将需要分析的数据表创建为数据集。更多操作请参见创建并管理数据集。

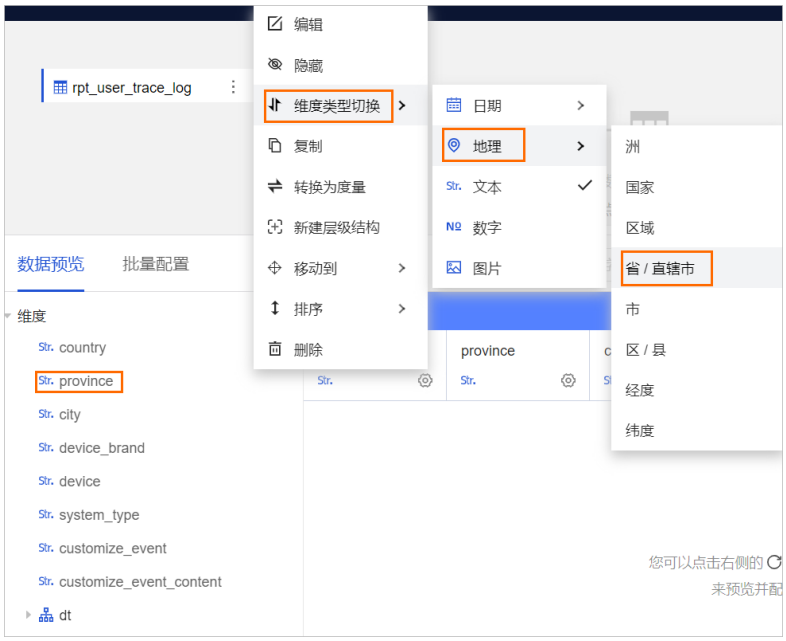
1. 在**数据源**页面选择连接成功的MaxCompute数据源，并在**rpt_user_trace_log**表后单击**创建数据集**图标。



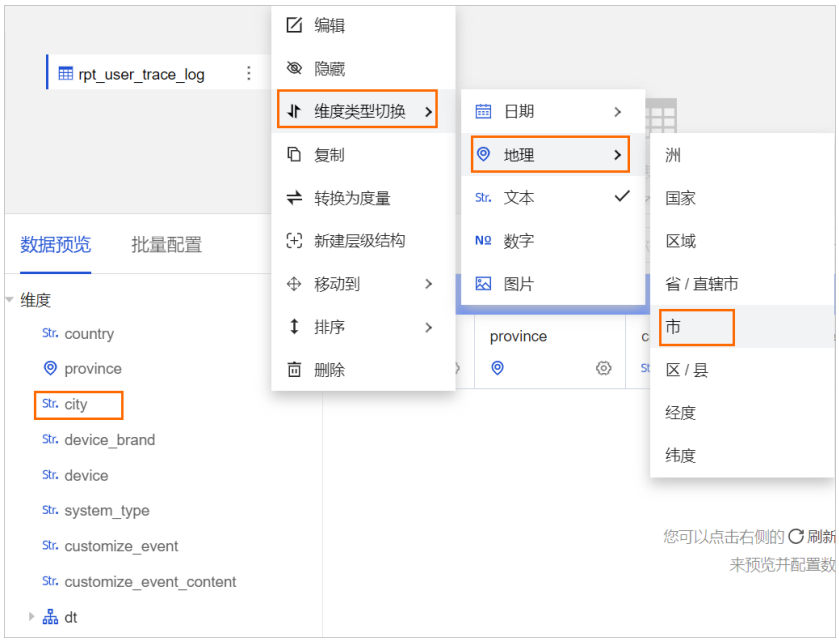
2. 在**数据集**页面，修改字段的展示格式。
   i. 按照下图指引，修改dt字段的日期展示格式。



ii. 按照下图指引，修改province字段类型的地理信息。



iii. 按照下图指引，修改city字段类型的地理信息。



iv. 按照下图指引，新建province层级结构。

v. 将city字段拖拽至province层级结构下，钻取效果如下。

步骤三：可视化展示

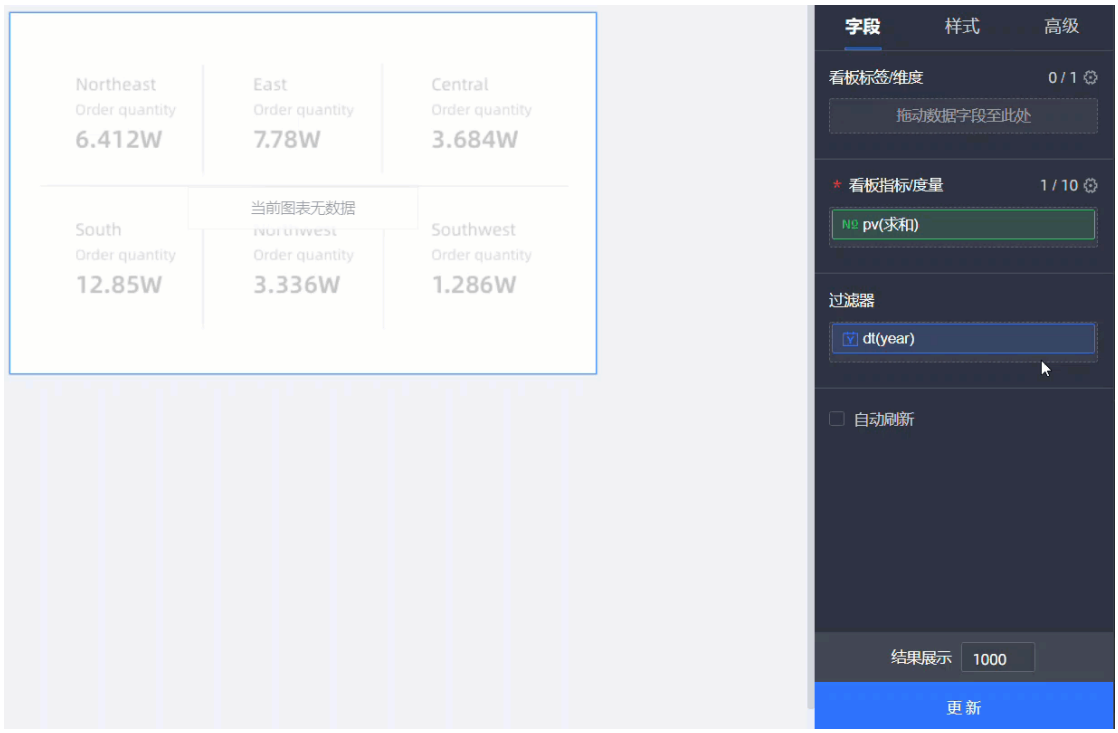
可视化图表可以帮助您直观、清晰地展示数据分析结果，如需了解更多，请参见可视化图表概述。

1. 在**数据集**页面，单击**开始分析**下拉列表的**创建仪表板**。

2. 制作**指标看板**。

i. 按照下图指引，制作pv指标看板。

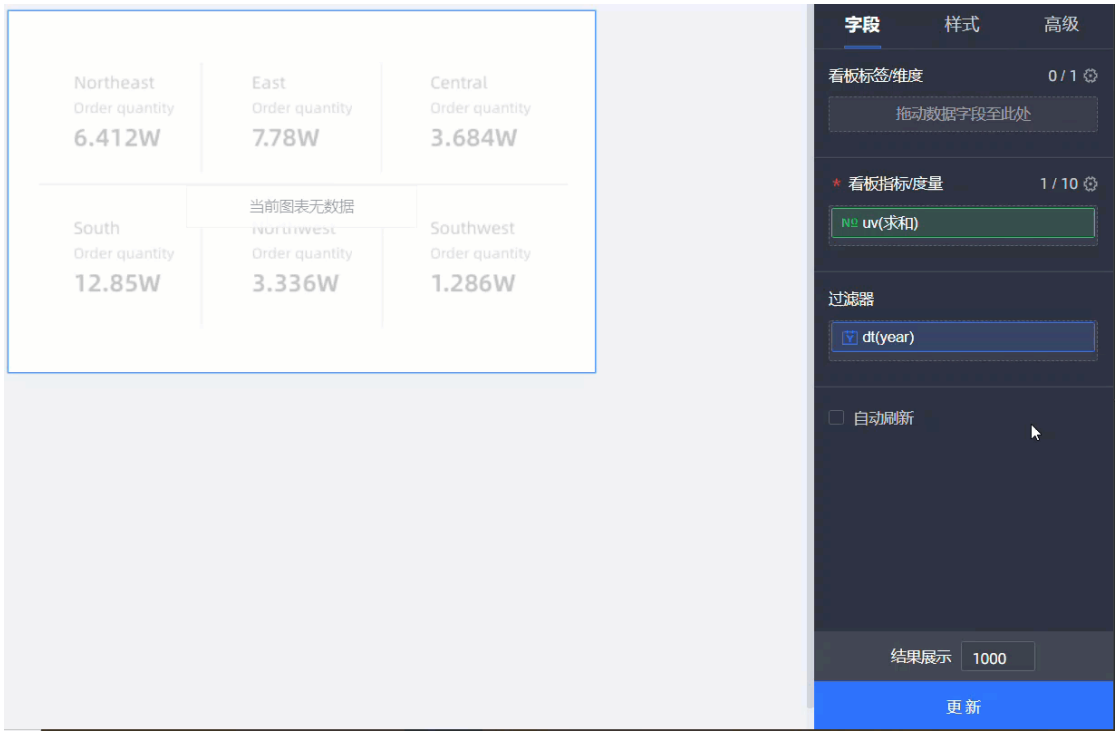
按照下图指引，设置pv指标看板过滤器。



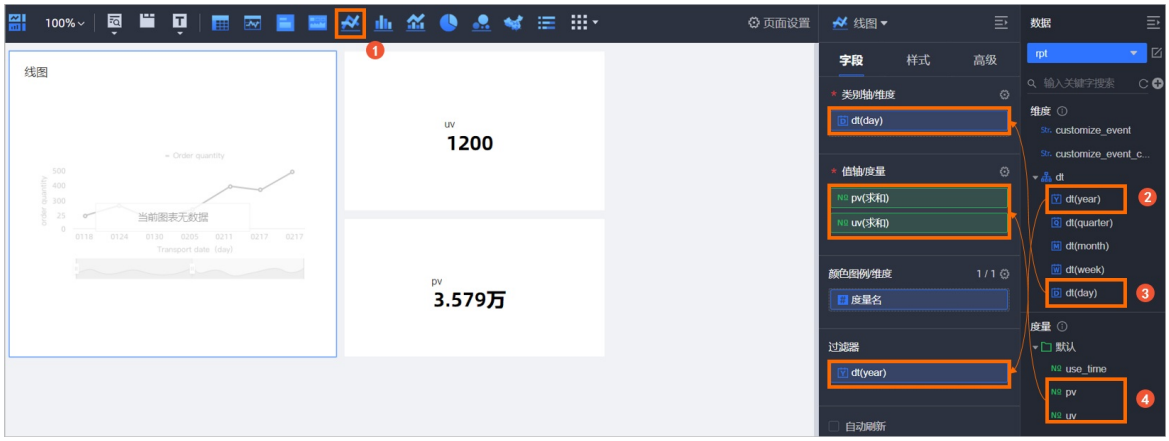
> **说明** 由于数据表rpt_user_trace_log为分区表，因此必须在**过滤器**中设置时间区间。上图过滤器的**时间区间**为2021年~2021年。

ii. 按照下图指引，制作uv指标看板。

按照下图指引，设置uv指标看板过滤器。

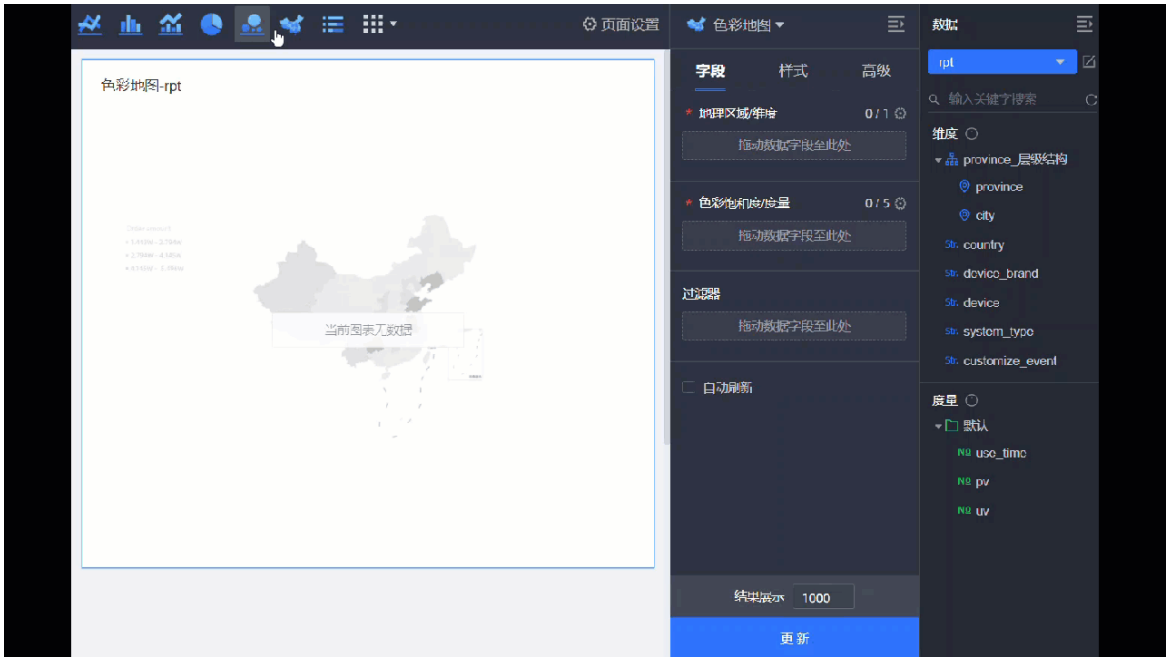


3. 按照下图指引，制作**线图**。

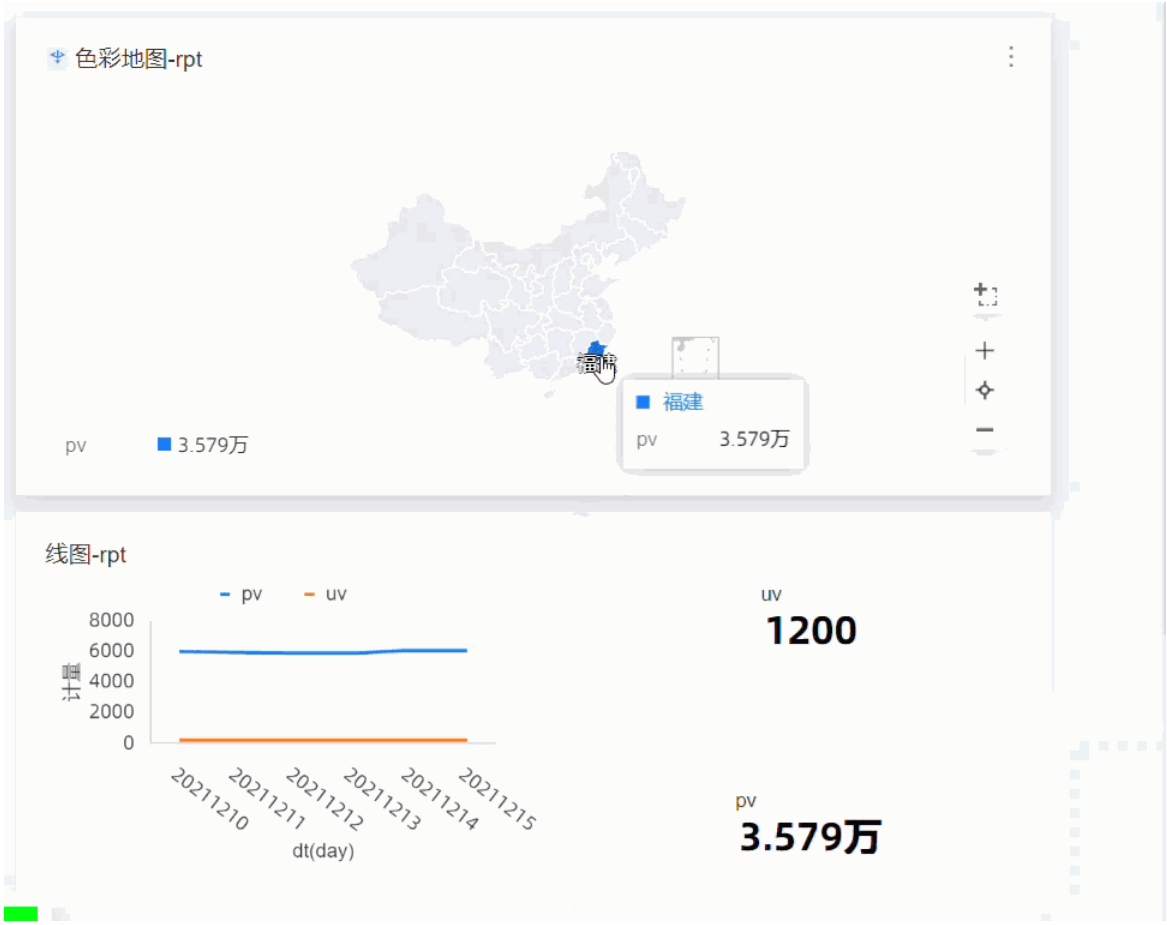


按照下图指引，设置线图过滤器。

4. 按照下图指引，制作**色彩地图**。



5. 设置完成后，在仪表板页面单击**保存**。
6. 在仪表板页面单击**预览**查看展示效果。



3.数据质量保障教程

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，请您首先完成搭建互联网在线运行分析平台教程，详情请参见业务场景与开发流程。

业务场景

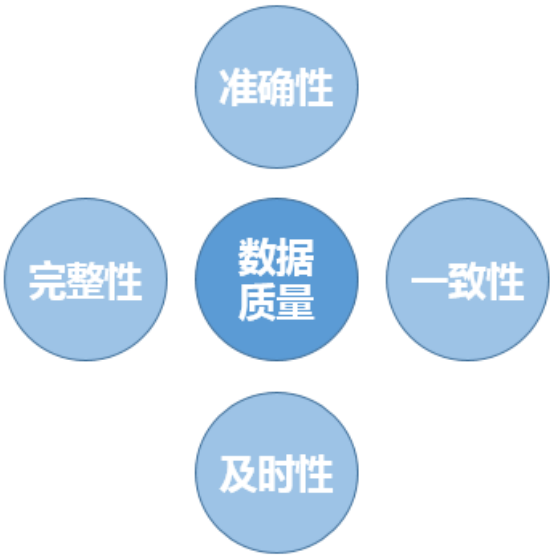
要保证业务数据质量，首先您需要明确数据的消费场景和加工链路。

本教程使用的数据来源于某网站上的HTTP访问日志。基于这份网站日志，您可以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。本教程基于搭建互联网在线运行分析平台教程，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。

数据质量的评估

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评估标准。



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、一致性；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（数据仓库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 `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。

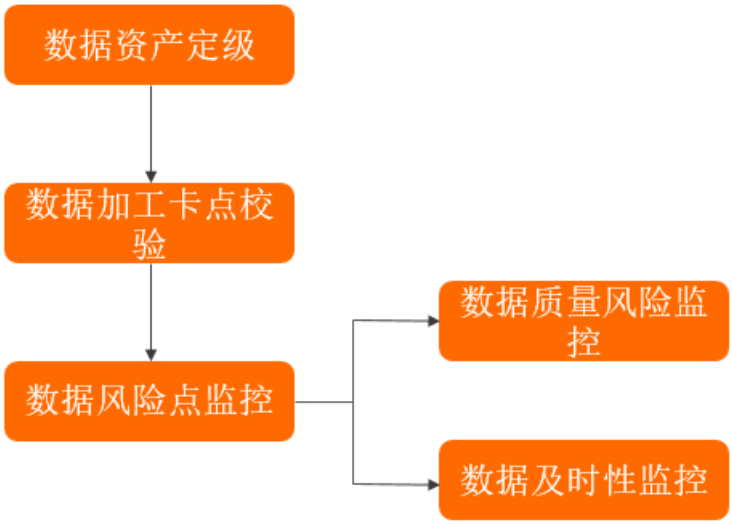
- 及时性

  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、风险点监控和及时性监控，您可以构建属于自己的数据质量保障体系。

数据质量管理的流程图如下。



数据质量管理的流程说明如下：

1. 分析业务场景，对数据流转链路上的整个依赖关系，进行资产定级。详情请参见数据资产定级。
2. 在业务系统的数据生成过程中进行卡点校验。详情请参见离线数据加工卡点校验。
3. 对数据风险点进行监控，包括数据的质量风险和及时性。详情请参见：
   - 数据质量风险监控
   - 数据及时性监控

3.3. 数据资产定级

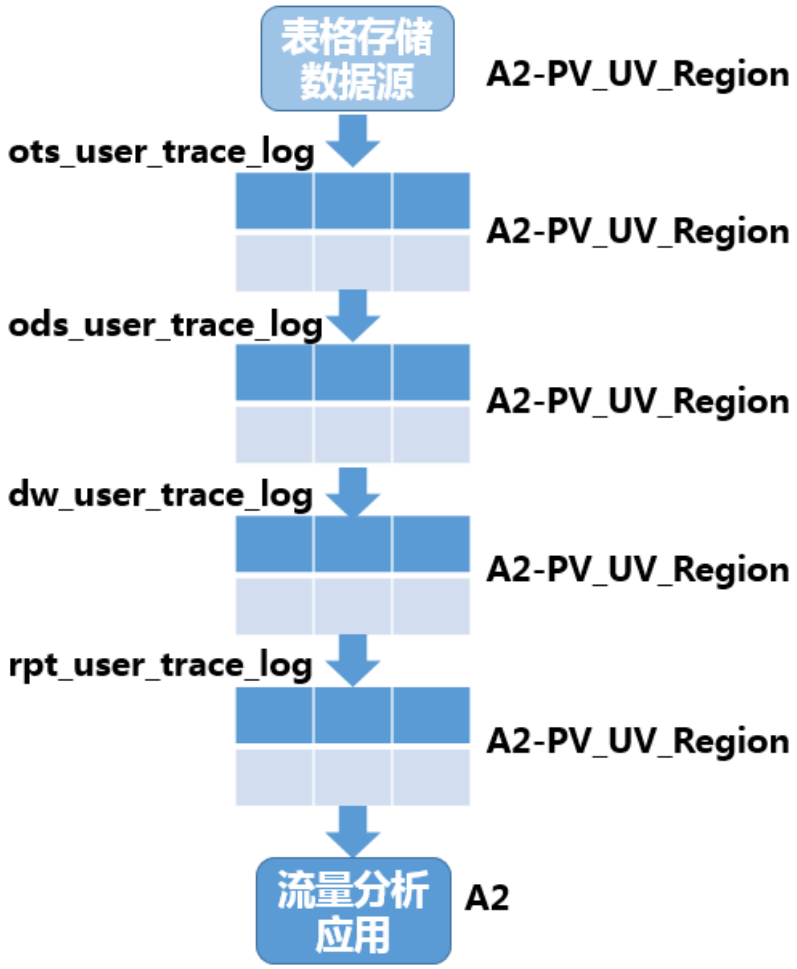
数据的资产等级，可以根据数据质量不满足完整性、准确性、一致性、及时性对业务的影响程度进行划分。

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。标记为A1。
- 全局性质：数据直接或间接用于企业级业务、效果评估和重要决策等。标记为A2。
- 局部性质：数据直接或间接用于某些业务线的运营、报告等，如果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。标记为A3。
- 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。标记为A4。
- 未知性质：无法明确数据的应用场景。标记为Ax。

资产等级标记包含毁灭性质为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，您需要整理消费各个表的应用业务。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，您可以使用第三方工具完成打标。

3.4. 离线数据加工卡点校验

本文为您介绍离线业务系统的数据在生成过程中进行的卡点校验。

代码提交卡点校验

代码提交卡点校验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。

  例如，表命名规范、生命周期设置及表注释等。

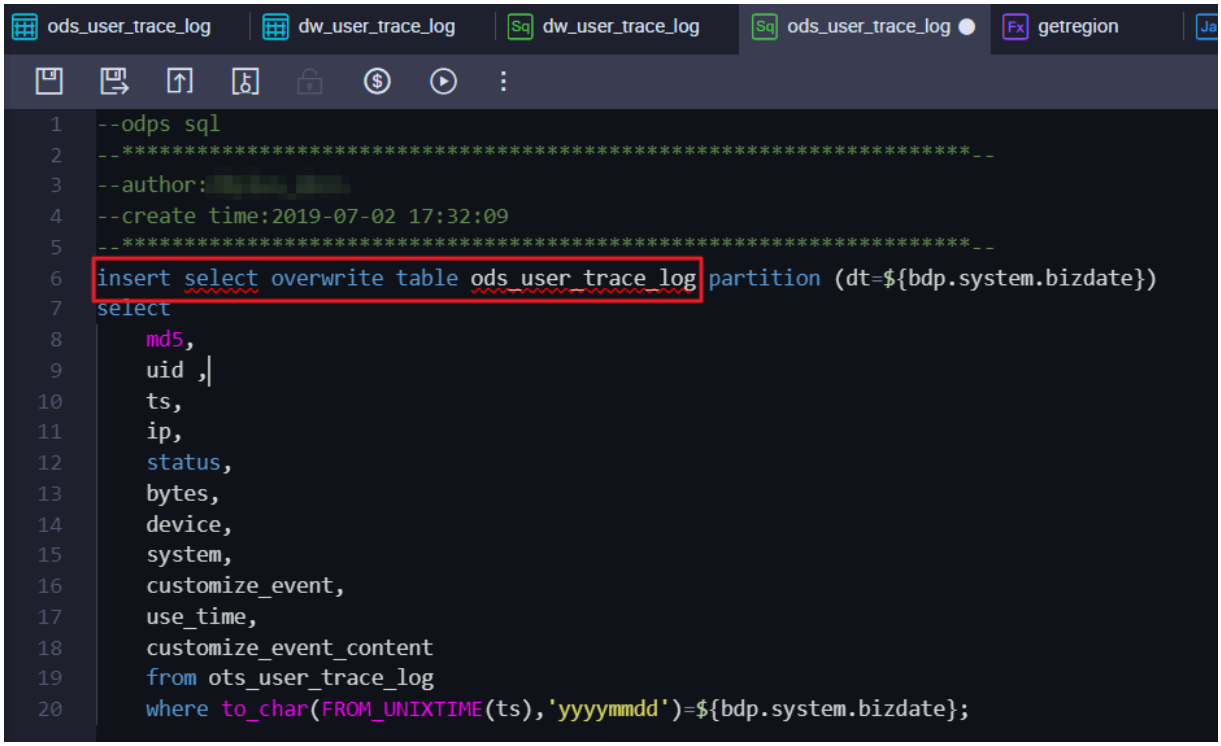
- 代码质量类规则。

  例如，分母为0提醒、NULL值参与计算影响结果提醒及插入字段顺序错误等。

- 代码性能类规则。

  例如，分区裁剪失效、扫描大表提醒及重复计算检测等。

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、生命周期及注释的其他规范，请参见表设计规范及MaxCompute数据开发规范。

任务发布上线卡点校验

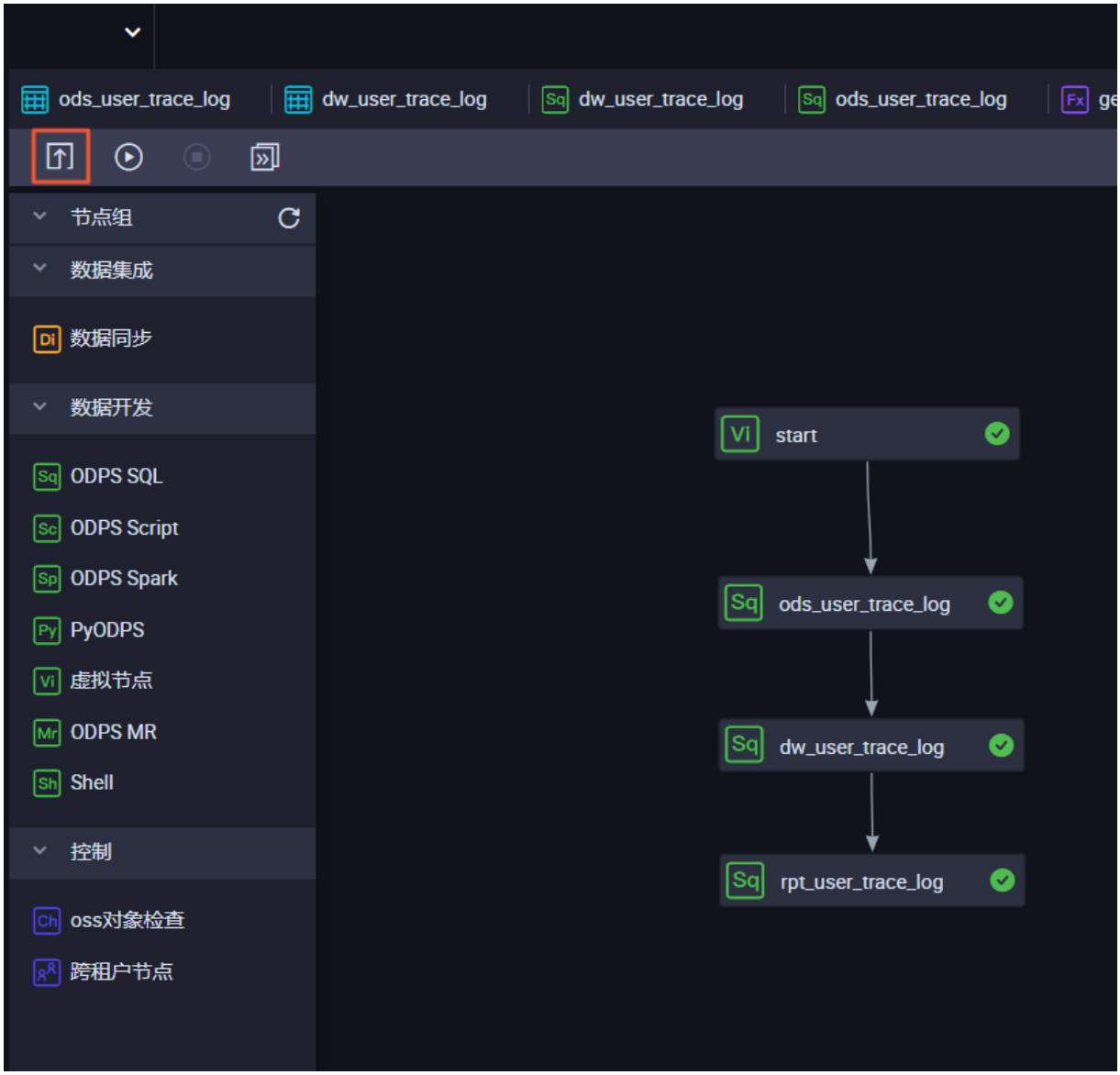
为保障线上数据的准确性，每次变更都需要经过测试再发布到线上生产环境，且生产环境测试通过后才算发布成功。

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。

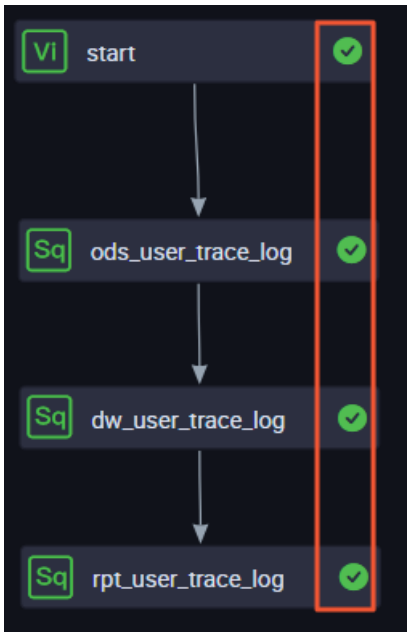
回归测试需要您能充分模拟真实环境进行测试：

- 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到开发环境，运行业务流程，观察是否存在报错。
- 对于简单模式项目，您可以直接运行业务流程，观察是否存在报错。

由于本教程使用简单模式，您直接提交任务运行业务流程即可。


ods_user_trace_log
dw_user_trace_log
dw_user_trace_log
ods_user_trace_log
节点组
数据集成
数据同步
数据开发
ODPS SQL
ODPS Script
ODPS Spark
PyODPS
虚拟节点
ODPS MR
Shell
控制
oss对象检查
跨租户节点
start
ods_user_trace_log
dw_user_trace_log
rpt_user_trace_log

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。



任务变更

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，这样可以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及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

首先您需要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，数据质量的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性和一致性。DWS和ADS层数据量较小、逻辑复杂，偏重数据的准确性。

> **说明** 如需了解各分层的详细说明，请参见数仓分层。

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。

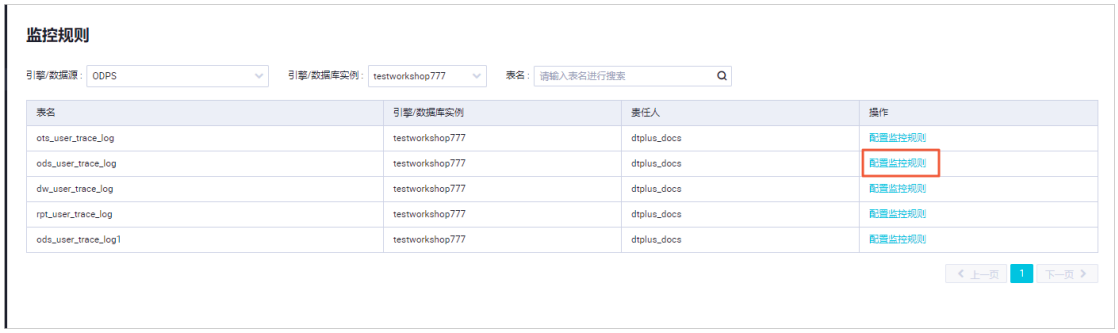
   ODS层表中的数据来源于OSS上的日志文件，作为源头表，您需要尽早判断此表分区中是否有数据。如果这张表中没有数据，则后续任务运行无意义，需要阻止后续任务运行。

   i. 进入**数据质量**页面。

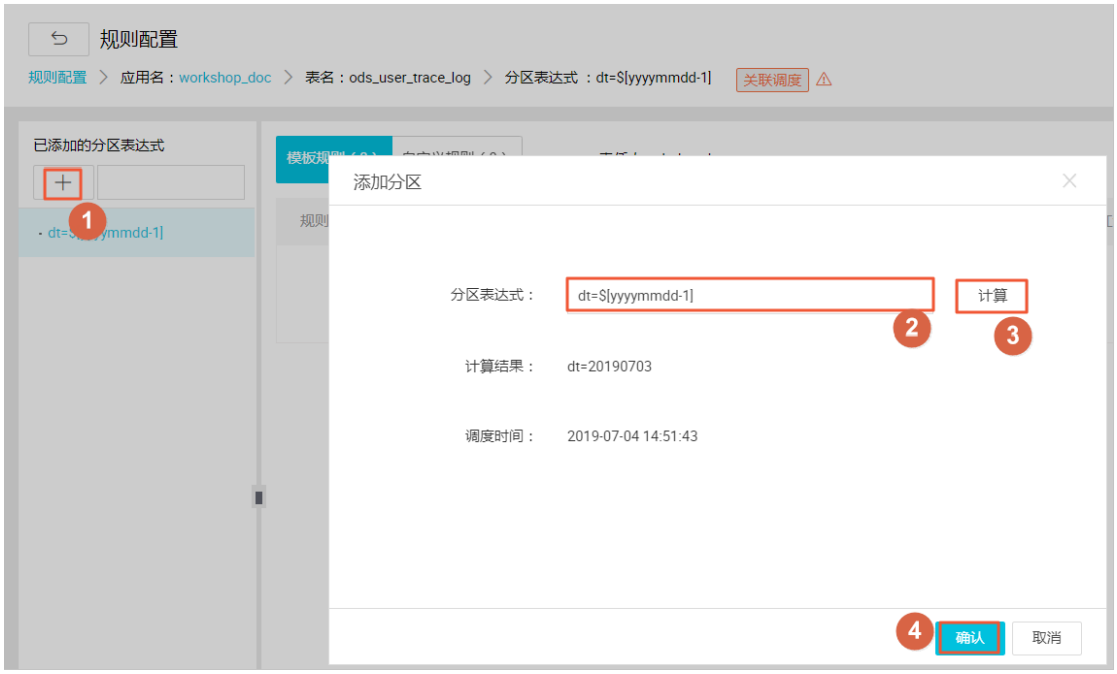
   在**数据开发**页面，单击左上角图标，选择**数据质量**。

   ii. 进入ods_user_trace_log监控规则页面。

   单击左侧导航栏上的**监控规则**，在**监控规则**页面找到代表外部数据源的ODS层表ods_user_trace_log，单击其后的**配置监控规则**。



iii. 添加分区。



a. 单击**+**，选择**分区表达式**为 dt=$[yyyymmdd-1]，对应表ods_user_trace_log的分区格式为${bdp.system.bizdate}（即获取到前一天的日期）。分区表达式的详细信息请参见配置调度参数。如果表中无分区列，可以配置无分区。

b. 单击**计算**，验证计算结果是否正确。

c. 单击**确认**，完成分区的添加。

iv. 创建规则确保ODS层表分区内存在数据。

a. 单击**创建规则**。

b. 单击**模板规则 > 添加监控规则**。

c. 输入配置参数。

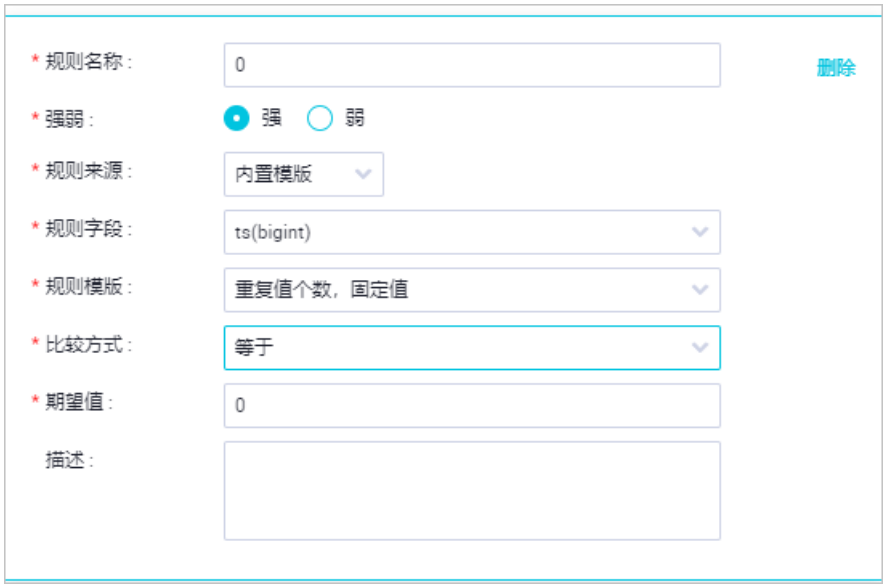
参数	描述
规则名称	请输入规则名称。您可以自定义。
强弱	设置为强规则。强弱规则说明如下： ■ 如果设置强规则，红色异常报警并阻塞下游任务节点，橙色异常报警不阻塞。 ■ 如果设置弱规则，红色异常报警不阻塞下游任务节点，橙色异常不报警不阻塞。
规则来源	选择**内置模板**。
规则字段	选择**表级规则**。
规则模板	选择**表行数，固定值**。

参数	描述
比较方式	选择**大于**。
期望值	设置为0。

v. 监控重复数据。

a. 单击**添加监控规则**。

b. 输入配置参数。



参数	描述
规则名称	请输入规则名称。您可以自定义。
强弱	设置为强规则。强弱规则说明如下： ■ 如果设置强规则，红色异常报警并阻塞下游任务节点，橙色异常报警不阻塞。 ■ 如果设置弱规则，红色异常报警不阻塞下游任务节点，橙色异常不报警不阻塞。
规则来源	选择**内置模板**。
规则字段	选择**ts(bigint)**。ts(bigint)值为用户时间戳，目的是避免ODS层出现重复的数据。
规则模板	选择**重复值个数**、**固定值**。
比较方式	选择**等于**。
期望值	设置为0。

vi. 监控空值数据。

a. 单击**添加监控规则**。

b. 输入配置参数。

参数	描述
规则名称	请输入规则名称。您可以自定义。
强弱	设置为强规则。强弱规则说明如下： ■ 如果设置强规则，红色异常报警并阻塞下游任务节点，橙色异常报警不阻塞。 ■ 如果设置弱规则，红色异常报警不阻塞下游任务节点，橙色异常不报警不阻塞。
规则来源	选择**内置模板**。
规则字段	选择**uid(string)**。uid(string)值为用户ID，目的是避免出现用户ID为空值的脏数据。
规则模板	选择**空值个数**、**固定值**。
比较方式	选择**等于**。
期望值	设置为0。

vii. 批量保存规则。完成上述操作后，单击**批量保存**。

viii. 规则试跑。

单击**试跑**，进行数据质量的校验规则。

ix. 查看试跑结果。

试跑完成后，单击**试跑成功！点击查看试跑结果**查看试跑结果。

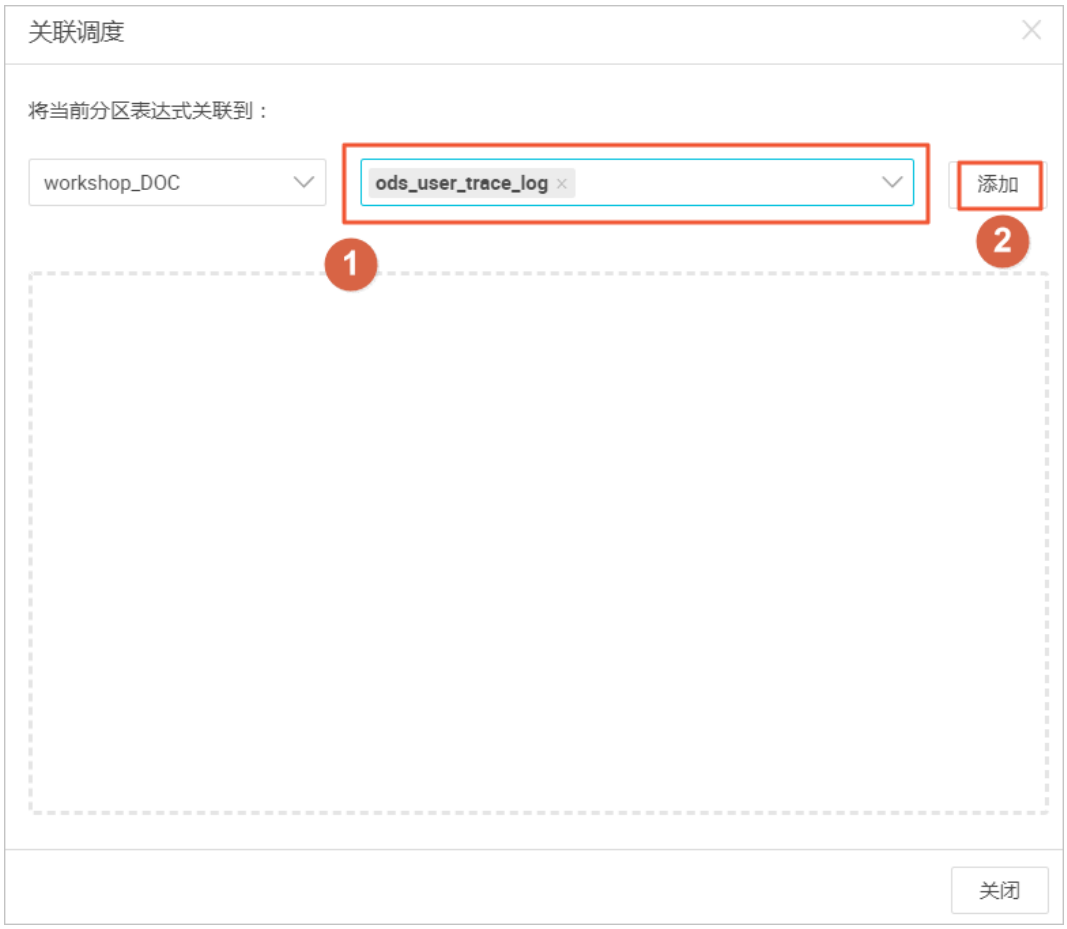
在弹出的页面中，您可以查看表数据是否已符合您的规则。根据试跑结果，可以确认此次任务产出的数据是否符合预期。建议每个表规则配置完毕后，都进行一次试跑操作，以验证表规则的适用性。

x. 关联调度。

在规则配置完毕，且试跑成功的情况下，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。

a. 在表监控规则页面，单击**关联调度**，配置规则与任务的绑定关系。

b. 在**关联调度**弹框中输入您需要关联的任务节点名称，单击**添加**。

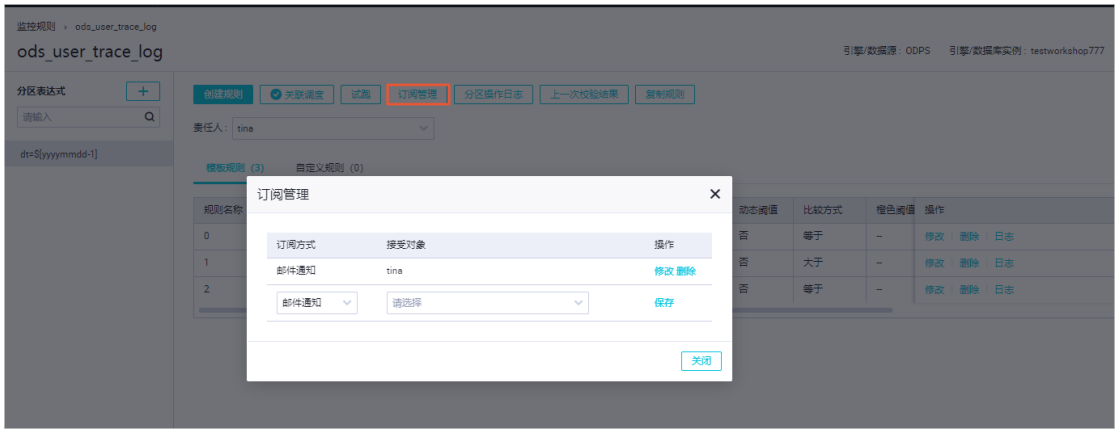


c. 单击**关闭**退出**关联调度**页面。如下图所示，关联调度配置成功。

xi. 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。

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，单击左侧导航栏上的**我的订阅**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，区别在于监控规则不同。

i. 添加分区表达式。

进入dw_user_trace_log表的规则配置页面，与ODS层一样配置分区为**dt=$[yyyymmdd-1]**，确保分区内存在表数据。

ii. 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。

iii. 监控表行数波动率。

参数	描述
规则名称	请输入规则名称。您可以自定义。
强弱	设置为强规则。强弱规则说明如下： ■ 如果设置强规则，红色异常报警并阻塞下游任务节点，橙色异常报警不阻塞。 ■ 如果设置弱规则，红色异常报警不阻塞下游任务节点，橙色异常不报警不阻塞。
规则来源	选择**内置模板**。
规则字段	选择**表级规则（table）**。
规则模板	选择**表行数、上周期波动率**。
比较方式	选择**绝对值**。
波动值比较	**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。

iv. 规则试跑并关联调度。方法和ODS层一致。

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不同。

i. 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]**。

ii. 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**。

iii. 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。

a. 单击**自定义规则 > 添加监控规则**。

b. 配置自定义规则参数。

参数	描述
规则名称	请输入规则名称。您可以自定义。
强弱	设置为弱规则。强弱规则说明如下： ■ 如果设置强规则，红色异常报警并阻塞下游任务节点，橙色异常报警不阻塞。 ■ 如果设置弱规则，红色异常报警不阻塞下游任务节点，橙色异常不报警不阻塞。
规则字段	选择规则字段为**pv(bigint)**。
采样方式	选择**sum**。
校验类型	选择**数值型**。
校验方式	选择**与固定值比较**。
比较方式	选择**大于**。
期望值	设置为100。当PV和异常锐减到100时，您可以及时收到告警。

c. 完成配置后，单击**批量保存**。

iv. 规则试跑并关联调度。方法与ODS层一致。

3.6. 数据及时性监控

基于MaxCompute的离线任务对数据产出有严格的时间要求，在确保数据准确性的前提下，还需要让数据能够及时提供服务。本文为您介绍如何使用DataWorks智能监控的规则管理功能监控数据的及时性。

前提条件

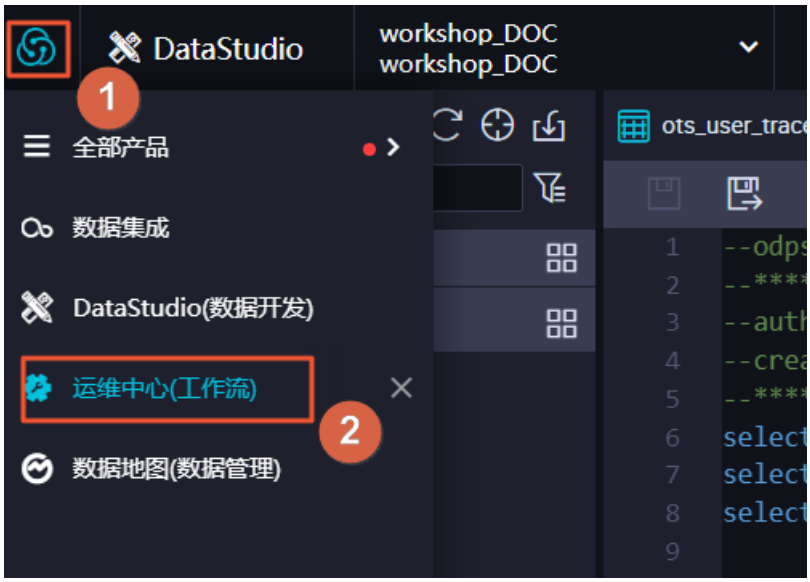
如果您想使用完整的智能监控功能，需要购买标准版及以上版本DataWorks，详情请参见DataWorks各版本详解。关于DataWorks智能监控功能详情请参见智能监控概述。

背景信息

您在监控数据产出的及时性前，首先需要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。
   i. 在DataStudio页面单击**运维中心（工作流）**。



   ii. 在**运维中心**页面，单击左侧导航栏上的**智能监控 > 规则管理**，关于**规则管理**的详情请参见自定义规则。
2. 新建自定义规则。

   单击右上角的**新建自定义规则**，输入参数后单击**确定**即可。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，则上报1次告警。连续上报3次告警，系统自动以邮件及短信的方式来上报。

分类	参数	描述
基本信息	**规则名称**	输入新建自定义规则的名称。
	对象类型	控制监控的粒度，包括**任务节点**、**基线**、**工作空间**、**业务流程**、**独享调度资源组**和**独享数据集成资源组**。
	规则对象	如果**对象类型**选择任务节点、基线、工作空间和业务流程，则需要填写**规则对象**。输入监控对象的名称或者ID后，在列表中选择需要添加的对象，单击图标。
	任务白名单	当**对象类型**为**基线**、**工作空间**、**业务流程**时，支持您输入**节点名称/ID**，**添加**至白名单列表中。白名单中的任务将不受监控。

分类	参数	描述
	资源组名称	如果**对象类型**选择独享调度资源组和独享数据集成资源组，则需要选择**资源组名称**。
触发方式	**触发条件**	如果**对象类型**选择任务节点、基线、工作空间和业务流程，此时触发条件取值如下： ○ **完成** 表示从实例任务运行的起始时间点开始监控，在任务运行成功时系统发送报警。 ○ **未完成** 表示从实例任务运行的起始时间点开始监控，到指定的目标时间点任务仍未结束运行，则系统发送报警。例如，实例任务的定时调度时间为1点，设置的未完成时间为2点，则1点时该任务开始运行，在2点时任务仍未结束运行，则发送报警。 ○ **出错** 表示从实例任务运行的起始时间点开始监控，如果任务运行出错，则系统发送报警。 实例任务运行出错即在**运维中心 > 周期任务运维 > 周期实例**的基本信息列，目标实例显示⊗状态。 ○ **周期未完成** 表示在指定的周期内，实例任务仍未结束运行，则系统发送报警。通常用于监控以小时为周期单位的实例任务。 例如，任务A每2小时调度一次，运行一次耗时25min。运行起始时间为每日0点0分，则该任务一天（24小时）共有12个任务周期，0点为第一个周期，2点为第二个周期，依次类推，22点为第12个周期。该任务正常运行时，会在每日0点25分、2点25分等时间节点执行完毕。如果在任意周期结束时间点该任务仍未结束运行，则发送报警。

分类	参数	描述
		说明 周期未完成可用于监控业务流程等任务。 当业务流程设置了**周期未完成**监控后，系统会根据您设置的周期N，对业务流程中的节点任务（例如，天任务、小时任务、分钟任务等），进行第N个周期任务的监控。如果任务实例数少于N时，则会忽略该任务的告警。 例如，设置的周期为3，业务流程中有如下两个节点任务，则告警监控情况如下： ■ 小时任务A：每2小时调度一次，运行一次耗时25min。运行起始时间为每日0点0分，则该任务一天（24小时）共有12个任务周期，0点为第一个周期，则第3个周期为4点。该任务正常运行时，第3个周期任务会在4点25分执行完毕。如果在该周期结束时间点该任务仍未结束运行，则发送报警。 ■ 分钟任务B：每10分钟调度一次，运行一次耗时2min。运行起始时间为每日0点0分，则该任务一小时共有6个任务周期，0点0分为第一个周期，则第3个周期为0点20分。该任务正常运行时，第3个周期任务会在0点22分执行完毕。如果在该周期结束时间点该任务仍未结束运行，则发送报警。
	触发条件	如果**对象类型**选择**独享调度资源组**和**独享数据集成资源组**，触发条件取值为： ○ **利用率**大于某个数值并持续多长时间。 例如：**利用率大于50%并持续15分钟**。 ○ **等资源实例数**大于某个数值并持续多长时间。 例如：**等资源实例数大于70并持续15分钟**。
报警行为	**报警方式**	包括**邮件**、**短信**、**电话钉钉群机器人**和**WebHook**。您可以添加钉钉群机器人接收报警，请参见下文的操作，将报警消息发送到**钉钉群**。如果您需要多个钉钉群接收报警信息，请添加多个Webhook地址。 ○ **报警方式为钉钉群机器人webhook时，您可单击操作列的发送测试消息**，测试该报警是否可以正常发送。如未收到告警信息，请参考智能监控进行排查。 ○ **自动重跑后仍出错** 报警方式为短信、邮件、电话时，您可以单击**校验联系方式**，校验手机号/邮箱是否填写正确。 表示从实例任务运行的起始时间点开始监控，如果任务运行出错且自动重跑后仍出错，则系统发送报警， 注意 ○ 您需要购买DataWorks专业版及以上版本，才可以使用电话告警功能。 ○ 如果您选择**报警方式**为**电话**，则需要选中**为了避免短时间内产生大量报警电话，DataWorks会对报警电话进行过滤，同一个用户在20分钟内最多接受到一通报警电话，其余报警电话将被降级为短信，请知悉。** ○ 仅支持钉钉Webhook地址。
	接收人	报警的对象，包括**任务责任人**、**值班表**和**其他**。
疲劳度控制	**最大报警次数**	报警的最大次数，超过设置的次数后，不再产生报警。
	最小报警间隔	两次报警之间的最小时间间隔。
	免打扰时间	在设置的时间段内不会发送报警。

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**。

3. 数据及时性优化。

   通常，影响数据按时产出的主要原因和优化方式如下表所示。

问题原因	问题优化
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。	扩容计算资源，或让核心计算任务独占资源。
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。	分级错峰，高峰时段让低优先级任务延迟启动。

问题原因	问题优化
缺少问题紧急预案，运维人员无法应对。	在任务正式运行前，进行充分的测试。