ALIBABA CLOUD

阿里云

智能数据构建与管理 Dataphin

使用教程

文档版本：20210712

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

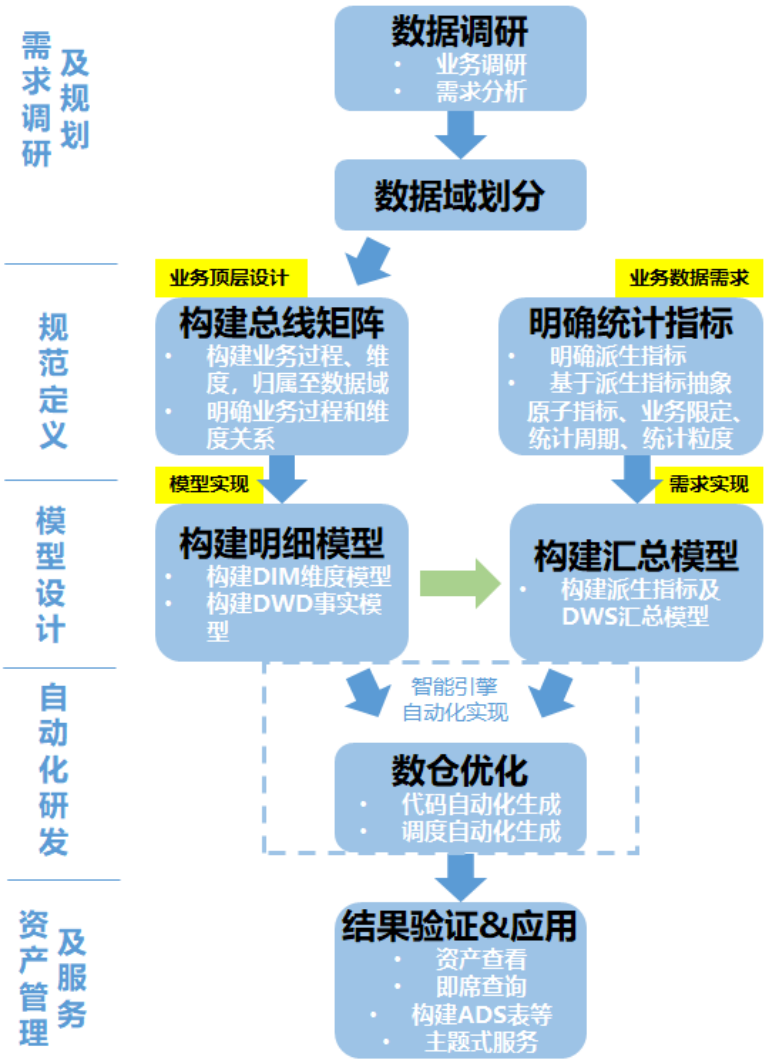
1.基于Dataphin构建数据仓库

1.1. 数仓构建流程

本文为您介绍如何使用Dataphin助力企业数据中台的建设与管理，快速构建标准且规范的数据仓库。

数仓构建流程

下图为使用Dataphin构建数据仓库的整体流程。



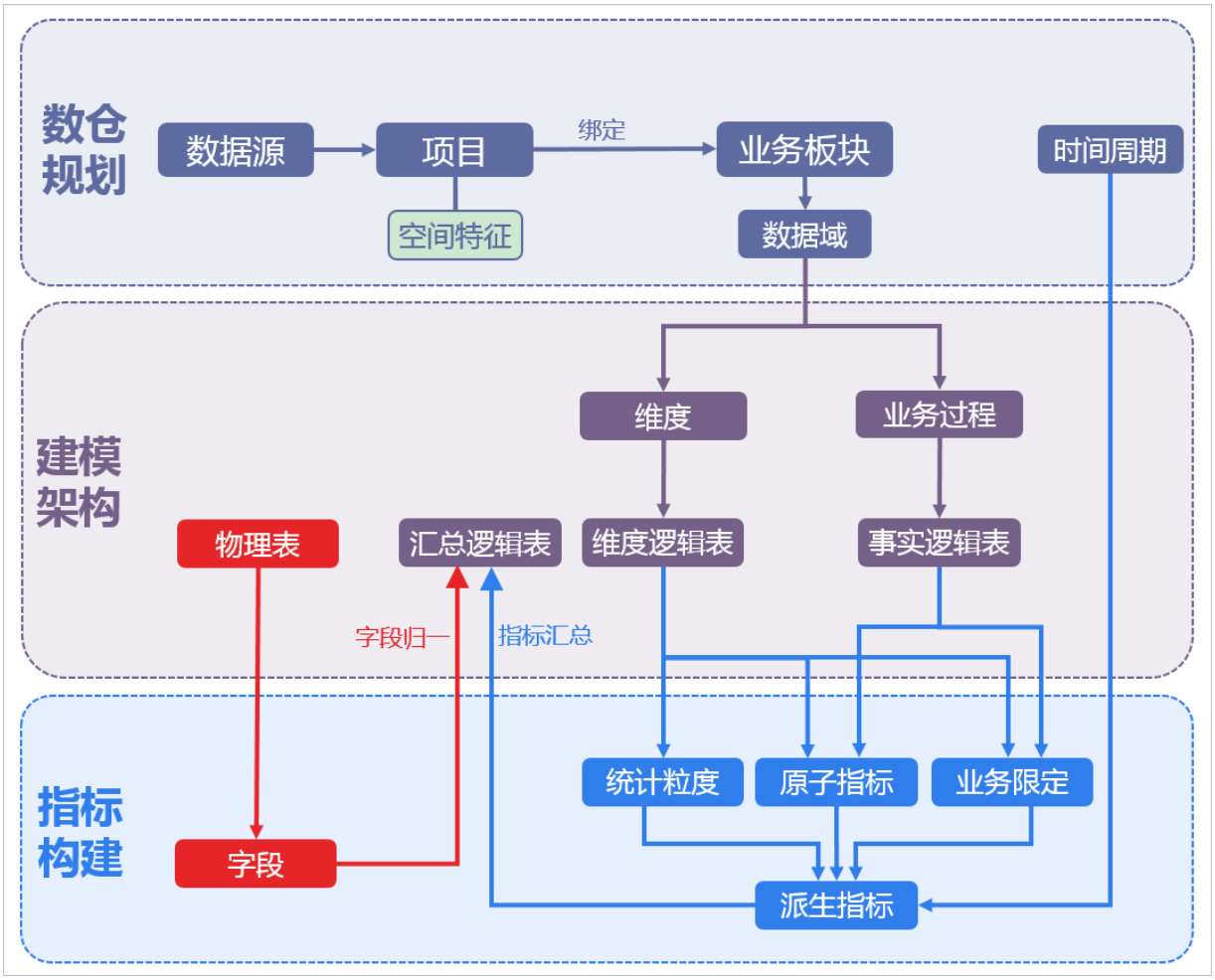
基本概念

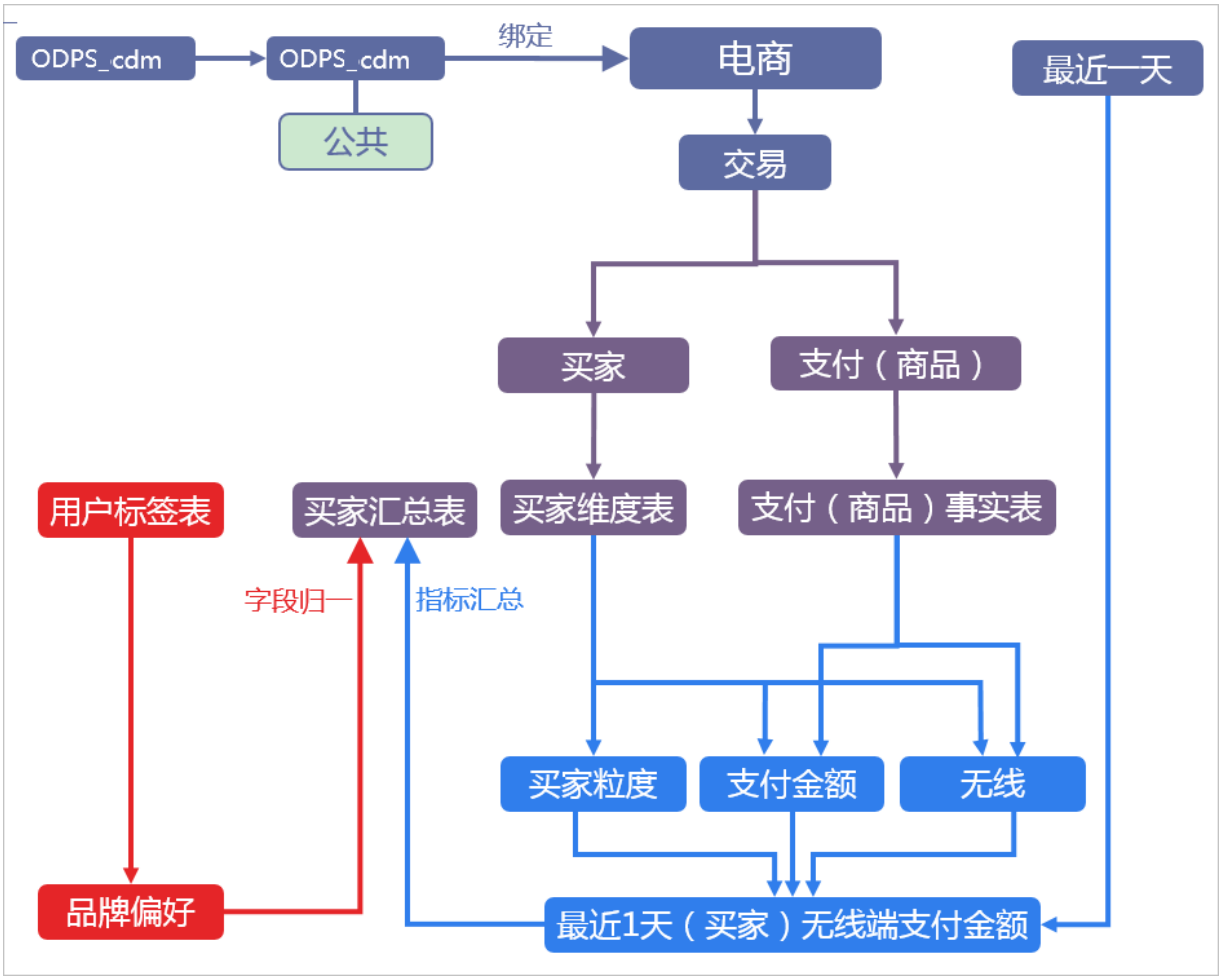
在正式学习本教程之前，您需要了解以下基本概念：

- 业务板块：业务板块定义了数据仓库的多种命名空间，是一种系统级的概念对象。当数据的业务含义存在较大差异时，您可以创建不同的业务板块，让各成员独立管理不同的业务，后续数据仓库的建设将按照业务板块进行划分。在Dataphin中，项目可以归属至业务板块以实现规范建模功能，同一个业务板块中可能包含多个不同的项目，所以业务板块与项目的关系为1:N。
- 数据域：数据域主要用于存放同一业务板块内不同概念的指标。例如，您可以划分出商品域、交易域、会员域等，用于存放不同意义的指标。
- 业务过程：业务过程即业务活动中所有的事件，通常为不可拆分的事件。创建业务过程，是为了从顶层视角，规范业务中的事务内容的类型及唯一性。

- 维度：维度即进行统计的对象。通常，维度是实际客观存在的实体。Dataphin遵循Ralph Kimball的维度建模理论，创建维度，即从顶层规范业务中实体（或称主数据）的存在性及唯一性。维度及维度组合，也是派生指标的统计粒度。
- 指标：指标分为原子指标和派生指标。派生指标是以原子指标为基准，组装统计粒度、统计周期及业务限定而生成的。
  - 原子指标是对指标统计口径、具体算法的一个抽象。根据计算逻辑复杂性，Dataphin将原子指标分为两种：
    - 原生的原子指标：例如支付金额。
    - 衍生原子指标：基于原子指标组合构建。例如，客单价通过支付金额除以买家数组合而来。
  - 派生指标是业务中常用的统计指标。为保证统计指标标准、规范、无二义性地生成，OneData方法论将派生指标抽象为四部分：`派生指标=原子指标+业务限定+统计周期+统计粒度`。
- 业务限定：统计的业务范围，用于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）。原子指标是计算逻辑的标准化定义，业务限定则是条件限制的标准化定义。
- 统计周期：统计的时间范围，也可以称为时间周期。例如最近1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，可以理解为聚合运算时的分组条件（类似于SQL中**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、省份这两个维度的组合。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。统计粒度也被称为粒度，是维度或维度组合，一般用于派生指标构建，是汇总表的唯一性识别方式。

基本概念之间的关系和举例如下图所示。





1.2. 数据调研

1.2.1. 确定需求

在基于Dataphin构建与管理企业数据中台之前，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，需要明确构建数仓服务的业务板块类型、每个板块具体满足什么业务需求。

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业务流程。业务流程通常与业务板块紧密耦合，对应一个或多个表及其所属数据源，可以作为构建数仓的原始数据来源。下表展现的是一个营销业务板块的业务流程模块。

业务流程	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表示包含该功能模块，N表示不包含。

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商品类目（例如厨具）在各省的销售总额、该类目销售额Top10的商品名称、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商品类目的精准营销，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，可能可用性较差。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思考和分析，并改进数据仓库。

需求分析的途径有两种：

- 通过与分析师、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、统计粒度，简称“粒度”，是维度或维度组合）汇总的，衡量标准是什么？例如，“省份”或者“类目”是维度，订单数是原子指标。
- 基于上个问题，进一步思考明细数据层的事实模型和公共可引用的维度模型、汇总数据层的汇总模型应该如何设计？是否有公共使用，命名及逻辑相似的统计指标，目前已经重复建设使用，需要通过上述设计规范化？

举例： 数据分析师需要了解A公司电商业务中最近1天厨具类目的成交金额。

1. 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（原子指标）、汇总的范围有多大（业务范围即业务限定，时间范围即统计周期）。例如，类目是统计粒度（基于维度），成交金额的总和是原子指标。该案例中，粒度应该是“类目”，“类目为厨具”是业务限定，最近1天是统计周期。

> **说明** 本例从类目为统计粒度的角度，分析需求处理。您可以在即席查询中定义汇总模型的筛选过滤条件，设定统计粒度的维度属性值为厨具，以免汇总模型数据稀疏。在真实业务场景下，可以根据业务需求、使用频度、复用性及汇总层数据计算存储进行考虑，拆解分析。例如，本例中还可以定义全表为粒度，只是该粒度中无需维度，然后定义业务限定是类目为厨具，其他保持不变，如无特殊数据情况，也可得到相同数据结果，只是计算存储过程消耗可能有不同。上述案例，不同路径，组合定义出来的派生指标，可能是相同结果，但是命名、计算逻辑实现可能略有不同。目前Dataphin上对于该类派生指标，认为是不同业务场景的指标，不进行强制去重。

2. 基于上述拆解，您还需要进一步思考并设计明细数据层的事实模型（原子指标中成交金额的数据来源）、公共可引用的维度模型（统计粒度的来源，且需要与成交金额所属事实模型有关联关系）和汇总数据层模型（原子指标、业务限定、统计周期的拆解和定义方式）。

需求调研的分析产出通常是记录业务需求的规范定义文档（派生指标、原子指标、业务限定、统计周期、统计粒度（即维度））。结合业务调研情况，您可以进一步产出设计明细逻辑模型设计文档（维度模型、事实模型）与概念模型设计文档（维度、业务过程及其关系）。

1.2.2. 分析业务过程

用户在业务系统中，通过埋点或日常积累的方式，获取了充足的业务数据。为梳理数据之间的逻辑关系和流向，需要理解用户的业务过程及数据系统。

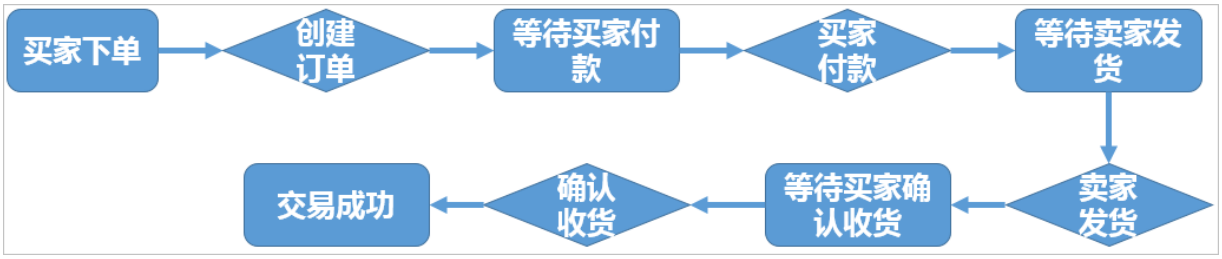
您可以采用过程分析法，列出整个业务过程涉及的每个环节，包括技术、数据、系统环境等。分析完企业的工作职责范围（部门）后，借助工具通过逆向工程抽取业务系统的真实模型。您可以参考业务规划设计文档和业务运行（开发、设计、变更等）相关文档，从以下几方面分析数据仓库涉及的源系统及业务管理系统：

- 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），也可以是某个事件的状态（例如当前的账户余额），还可以是一系列相关业务事件组成的业务过程。具体取决于您分析的是某些事件过去的发生情况、当前状态，或是事件流转效率。分析业务过程的流程如下：

1. 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。
2. 设计维表。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。
3. 确定衡量指标。

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



这是一个非常典型的电商交易业务过程图。在该业务过程中，有**创建订单**、**买家付款**、**卖家发货**、**确认收货**四个核心业务环节。确认收货即表示交易成功，所以我们重点分析**确认收货**环节。

在明确用户的业务过程之后，您可以根据需要分析决策的业务数据域，并在相应的数据域下创建具体的业务过程。

基于Dataphin完成业务过程的构建，详情请参见新建业务过程。

1.3. 架构设计与规范定义

1.3.1. 划分数据域

数据仓库是面向主题的应用，主要功能是将数据综合、归类并进行分析利用。数据仓库模型设计除横向的分层外，通常还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层次归类，目的是便于数据的管理和应用。

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。然后，进行跨源的主题域合并，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，但不轻易变动。划分数据域时，需满足以下两点：

- 能涵盖当前所有的业务需求。
- 能在新业务进入时，无影响地被包含进已有的数据域中和扩展新的数据域。

在业务调研之后，可以进行数据域的划分。划分数据域，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划分。

例如，A公司电商营销业务板块可以划分为如下表所示的数据域。数据域中的每一部分，都是根据实际业务过程进行归纳、抽象得出的。

数据域	业务过程举例
会员和店铺域	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（交易成功）
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）

基于Dataphin完成数据域的构建，详情请参见新建数据域。

1.3.2. 定义维度与构建总线矩阵

根据阿里巴巴OneData方法论，明确每个数据域中有哪些业务过程后，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。本教程中，以A电商公司的营销业务板块为例，在交易数据域中，我们重点分析确认收货（交易成功）的业务过程。

在确认收货的业务过程中，维度所依赖的业务角度主要有两个，即商品和收货地点（地域）。本教程中，假设收货和购买是同一个地点。

- 从商品角度分析，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：0全新；1闲置；2二手
  - 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用户删除；2下架；3未上架
  - 商品所在城市

- 商品所在省份

- 从地域角度分析，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，维度在企业级数据仓库中必须具有唯一性。维度在每个业务板块内必须具有唯一性，即每个维度在所属业务板块内有且只有一种定义。例如本教程内的省份维度，对于营销业务板块内的任何业务过程所传达的信息都是一致的。基于Dataphin完成维度的构建，详情请参见创建维度及维度表。

维度创建、发布成功后，系统会自动创建对应的维表（即维度逻辑表），并支持对维表进行添加属性、添加关联维度、添加子维度和物理化配置等操作。维表与维度一一对应，是通过丰富维度中的属性信息构建而成的，详情请参见添加关联维度、属性和子维度。

构建总线矩阵

明确每个数据域中有哪些业务过程后，即可构建总线矩阵，该总线矩阵将用于指导后续事实模型中关联维度的定义，构建数据仓库的雪花模型。

您需要定义每个数据域下的业务过程和维度，并明确每个业务过程与哪些维度相关。下表是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ID、类目名称、品牌ID、品牌名称、商品ID、商品名称、成交金额等维度，并明确了不同业务过程包含了哪些维度。

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

> **说明** Y表示包含该维度，N表示不包含。

1.3.3. 明确统计指标

统计指标包括派生指标、原子指标、业务限定、业务过程和统计粒度（即维度）。在设计模型前，建议先完成该部分工作，以便设计出易于使用的数据仓库。

指标定义注意事项

原子指标是明确统计口径和计算逻辑，事实模型或维度模型定义完成，即可创建原子指标。派生指标即常见的统计指标，`派生指标=统计周期+业务限定+原子指标+统计粒度`。

创建派生指标，注意事项如下：

- 已完成原子指标的创建，且需要确认原子指标的来源模型中有维度模型，以保证可以设置派生指标的统计粒度。
- 原子指标和业务限定来源于同一张维度表或事实表，且继承来源表的数据域。
- 统计粒度和时间周期必选，是否选择业务限定由具体的派生指标语义决定。例如，如果支付金额为原子指标，则最近7天买家支付金额（统计粒度为买家、时间周期为最近7天）和最近7天买家支付宝支付金额（统计粒度为买家、业务限定为支付宝支付、时间周期为最近7天）都可以作为派生指标。
- 派生指标唯一归属于一个原子指标，且继承原子指标的数据域。

确定指标

本教程中，用户是A公司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、该类目销售额前10的商品名称、各省用户购买力分布（人均消费额）等，用于营销分析。

基于规范定义，确认业务过程为确认收货（交易成功），对应事实模型中的度量（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和派生指标：

- 原子指标：商品成功交易的金额的总和。
- 派生指标：
  - 最近一天全省厨具类目各商品的销售总额。
  - 最近一天全省厨具类目的人均消费额（消费总额除以人数）。

最近一天全省厨具类目各商品的销售总额降序排序，取前10名的名称，即可得到该类目销售额前10的商品名称。

创建指标

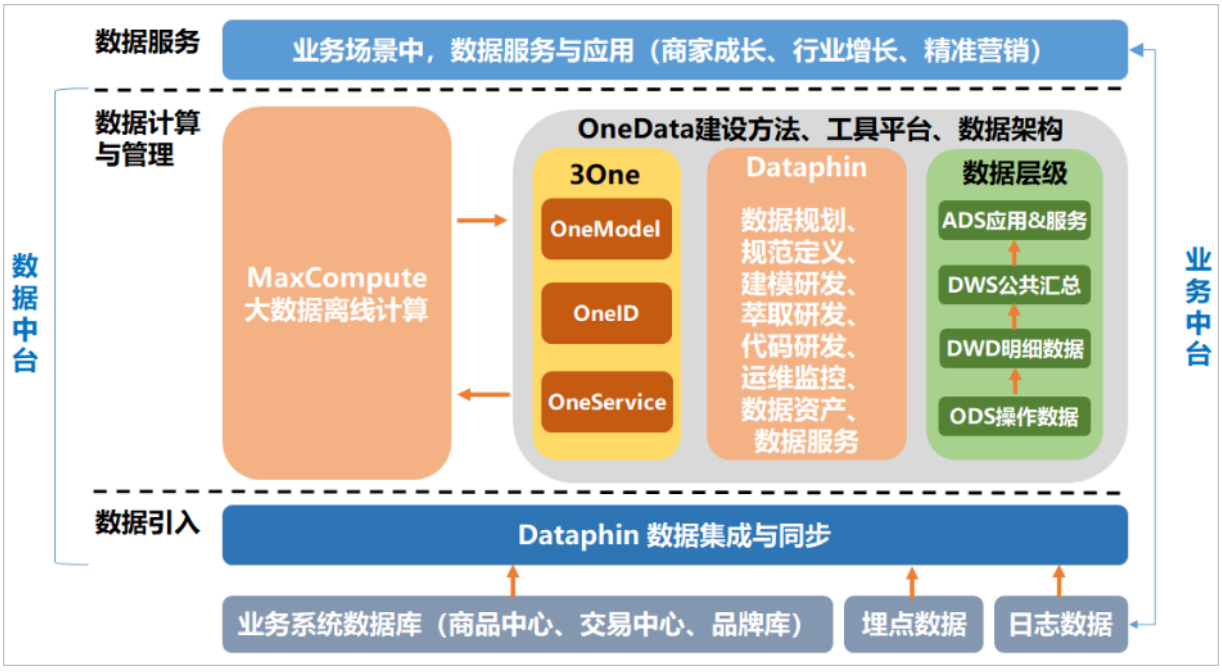
完成原子指标、派生指标、业务限定、维度和业务过程的创建，详情请参见规范建模。

1.4. 模型设计

1.4.1. 技术架构选型

根据阿里巴巴OneData方法论最佳实践，在设计数据模型前，您需要完成技术架构的选型。本教程中使用阿里云大数据产品Dataphin配合MaxCompute，完成整体的数据建模和研发流程。

完整的技术架构如下图所示。其中，Dataphin的数据集成及同步负责完成源业务系统数据引入。MaxCompute作为整个大数据开发过程中的离线计算引擎。Dataphin则基于OneData方法论——OneModel、OneID、OneService，囊括了数据建模研发、运维中心、监控报警、数据资产等在内的一系列功能。



1.4.2. 数仓分层

基于阿里巴巴OneData方法论最佳实践，在阿里巴巴的数据体系中，建议将数据仓库分为三层：数据引入层（ODS，Operational Data Store）、数据公共层（CDM，Common Dimensions Model）和数据应用层（ADS，Application Data Store）。

数据仓库自顶向下的分层和各层用途如下图所示。

- 数据引入层（ODS，Operational Data Store，又称数据基础层）：将原始数据几乎无处理地存放在数据仓库系统中，结构上与源系统基本保持一致，是数据仓库的数据准备区。这一层的主要职责是将基础数据同步、存储到MaxCompute。
- 数据公共层（CDM，Common Dimensions Model）：存放明细事实数据、维表数据及公共指标汇总数据。其中，明细事实数据、维表数据一般根据ODS层数据加工生成。公共指标汇总数据一般根据维表数据和明细事实数据加工生成。

  CDM层又细分为维度层（DIM）、明细数据层（DWD）和汇总数据层（DWS），采用维度模型方法作为理论基础，可以定义维度模型主键与事实模型中外键关系，减少数据冗余，也提高明细数据表的易用性。在汇总数据层同样可以关联复用统计粒度中的维度，采取更多的宽表化手段构建公共指标数据层，提升公共指标的复用性，减少重复加工。

  - 维度层（DIM，Dimension）：以维度作为建模驱动，基于每个维度的业务含义，通过添加维度属性、关联维度等定义计算逻辑，完成属性定义的过程并建立一致的数据分析维表。为了避免在维度模型中冗余关联维度的属性，基于雪花模型构建维度表。

    在Dataphin中，维度层的表通常也被称为维度逻辑表。
  - 明细数据层（DWD，Data Warehouse Detail）：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属性字段做适当冗余，也即宽表化处理。

    在Dataphin中，明细数据层的表通常也被称为事实逻辑表。
  - 汇总数据层（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。

    在Dataphin中，汇总数据层的表通常也被称为汇总逻辑表，用于存放派生指标数据。
- 数据应用层（ADS，Application Data Store）：存放数据产品个性化的统计指标数据，根据CDM层与ODS层加工生成。

1.4.3. 数据引入层（ODS）

基于阿里巴巴OneData方法论最佳实践，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长期累积的、非常细节的数据，且访问频率很高，是面向应用的数据。

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。在逻辑层面上，这些数据都是以二维表的形式存储。严格地说，虽然ODS层不属于数仓建模的范畴，但是合理地规划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。在Dataphin中，di后缀的事实模型为增量表（事务型），df后缀的事实模型为全量表（周期快照型）。
> - 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、数据质量规范，详情请参见ODS层设计规范。

建表示例

通过即席查询功能，您可以编写SQL语句创建所需的ODS表。为方便您使用，集中为您提供建表语句如下。

> **说明** 即席查询功能，详情请参见概述。

```
CREATE TABLE IF NOT EXISTS s_auction
(
  id                STRING COMMENT '商品ID',
  title             STRING COMMENT '商品名称',
  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 '商品名称',
  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;
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;
```

数据同步加载与处理

ODS的数据需要由各数据源系统同步、存储到MaxCompute，才能用于进一步的数据开发。本教程建议您使用Dataphin的数据引入功能完成数据同步，详情请参见概述。在使用数据引入功能的过程中，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。
- 数据引入支持全量数据同步、实时增量数据同步（分钟或小时调度实现）两种同步方式。
- 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，Dataphin中默认时间分区的名字为ds。
- 数据引入支持手动调整源表和目标表的同步字段。
  - 如果源表字段在目标表中不存在，用户需手动添加目标字段，或删除源表字段。
  - 如果源表字段与目标表字段不匹配，用户需先删除目标字段，然后重新添加与之匹配的字段。

1.4.4. 维度层（DIM）

本文为您介绍维度层的设计原则、维度表的规范、创建维度及查询维度逻辑表。

维度层简介

建立一致数据分析维表，可以降低数据计算口径和算法不统一风险。以维度作为建模驱动，基于每个维度的业务含义，通过定义维度及维度主键，添加维度属性、关联维度等定义计算逻辑和雪花模型，完成属性定义的过程并建立一致的数据分析维表。同时您可以定义维度主子关系，子维度的属性将合并至主维度使用，进一步保证维度的一致性和便捷使用性。

维度表设计原则

- 尽可能生成丰富的维度属性。

  例如电商公司的商品维度可能有近百个维度属性，为下游的数据统计、分析、探查提供了良好的基础。

- 尽可能多的给出包含一些富有意义的文字性描述。

  属性不应该是编码，而应该是真正的文字。在阿里巴巴维度建模中，通常是编码和文字同时存在，例如商品维度中的商品ID和商品标题、类目ID和类目名称等。ID通常用于不同表之间的关联，而名称通常用于报表标签。

- 区分数值型属性和事实。

  数值型字段是作为事实还是维度属性，可以根据字段的常用用途区分。例如，若用于查询约束条件或分组统计，则是作为维度属性；若用于参与度量的计算，则是作为事实。

- 尽量沉淀出通用的维度属性。
  - 通过逻辑处理得到维度属性。
  - 通过多表关联得到维度属性。
  - 通过单表的不同字段混合处理得到维度属性。
  - 通过对单表的某个字段进行解析得到维度属性。

维度表规范

提交普通维度或层级维度时，会自动生成对应的维度逻辑表，不支持用户自定义新建维度逻辑表。此外，Dataphin还支持定义枚举维度和虚拟维度。提交枚举维度和虚拟维度不会生成维度逻辑表。

> 说明
>
> - 枚举维度指的是维度表的值可枚举，以便规范统一枚举的维度值，维度作为派生指标统计粒度时，实现数据归一汇总计算。
> - 虚拟维度与某个字段关联后，以维度的形式作为统计粒度，定义派生指标。例如URL。

创建维度时，会自动生成维度逻辑表。基于Dataphin，维度表名称默认前缀为dim_，层级维度表默认增加后缀_lvl{n}。举例如下：

- dim_c1（普通维度逻辑表）
- dim_c1_lvl1（层级维度逻辑表）

创建维度及维度表

创建维度，详情请参见新建维度。

维度创建成功后，系统自动生成维度逻辑表。

查询维度表

逻辑表运维包含两个模块，一个是逻辑表任务，用来从逻辑表视角切入，为您展现逻辑表内部任务关系，详情请参见逻辑表任务。另一个是逻辑表实例，用于查看已运行的逻辑表任务包含的节点实例及其状态，详情请参见逻辑表实例。

1.4.5. 明细数据层（DWD）

基于阿里巴巴方法论最佳实践，事实表（事实模型，又称事实逻辑表）作为数据仓库维度建模的核心，紧紧围绕着业务过程进行设计。业务过程是通过事实表的度量、引用的维度与业务过程有关属性的方式获取。

Dataphin支持两种类型的事实表：

- 事务型事实表：用于描述业务过程，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，表名后缀一般为di。

- 周期快照型事实表：以具有规律性的、可预见的时间间隔（例如每天、每月、每年等）记录事实，一般表名后缀为df。

事实表设计原则

- 尽可能包含所有与业务过程相关的事实。

  设计事实表的目的是度量业务过程，所以分析哪些事实与业务过程有关，是事实表设计中至关重要的。在事实表中应该尽量包含所有与业务过程相关的事实，即使存在冗余，但是因为事实通常为数字型，带来的存储开销不会很大。

- 只选择与业务过程相关的事实。

  在选择事实时应该注意，只选择与业务过程有关的事实。例如，A公司的订单交易业务流程中，在设计下单这个业务过程的事实表时，不能包含支付金额这个表示支付业务过程的事实。

- 在选择维度和事实之前，必须先声明粒度。

  粒度（数据行数的最小单位，非统计粒度）的声明是事实表设计中不可忽视的重要一步。粒度用于确定事实表中一行所表示业务的细节层次，决定了维度模型的扩展性。在选择维度和事实之前，必须先声明粒度，且每个维度和事实必须与所定义的粒度保持一致。在事实表中，通常通过业务描述来表述粒度并定义事实表主键，但对于聚集性事实表的粒度描述（例如存在下单、支付等多个事务），可以基于多个字段拼接，形成新的字段作为事实表主键，也可以不定义主键，这样一行记录即最小粒度。

- 在同一个事实表中，不能包含多种不同粒度的事实。

  事实表中所有事实的粒度需要与表声明的粒度保持一致，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。

  在同一个事实表中，事实的单位应该保持一致。例如，原订单金额、 订单优惠金额、订单运费金额这三个事实，应该采用一致的计量单位，例如统一为元，以方便使用。

事实表设计方法

任何类型的事件都可以被理解为一种事务。例如，交易过程中的创建订单、买家付款，物流过程中的揽货、发货、签收，退款过程中的申请退款、申请客服介入等，都可以被理解为一种事务。事务型事实表，即针对这些过程构建的一类事实表，用以跟踪定义业务过程的个体行为，提供丰富的分析能力，作为数据仓库CDM层的明细数据。

下面以A公司的订单交易事务型事实表为例，阐述事务型事实表的一般设计过程。

1. 选择业务过程。

   按照之前的业务流程分析，A公司的交易订单流程包含四个重要过程：创建订单、买家付款、卖家发货、确认收货，即下单、支付、发货和收货四个业务过程。这四个业务过程不仅是交易过程中的重要时间节点，而且也是下游统计分析的重点，因此A公司的交易事务事实表设计着重从这四个业务过程进行展开。

   为了便于进行独立的分析研究，我们应该为每个业务过程建立一个事实表。本教程中，我们选择交易成功这个业务过程，建立事务型事实表。

2. 确定粒度。

   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（例如商品）。

   业务过程选定之后，就要针对业务过程确定一个粒度，即确定事务型事实表每一行所表达的细节层次。明确的粒度能确保对事实表中行的意思的理解不会产生混淆，保证所有的事实按照同样的细节层次记录。如果有字段可以表达这个粒度，可以定义为事实表的主键。

   应该尽量选择最细级别的粒度，以确保事实表的应用具有最大的灵活性。对于订单过程而言，每一种商品结算后都会产生一个订单，交易成功这个业务过程的粒度可以选择为订单。订单ID如果唯一，可以作为事实表主键以描述粒度。

3. 确定维度。

   选定好业务过程并且确定粒度之后，就可以确定维度信息了，应该选择能够描述清楚业务过程所处的环境的维度信息。例如，在A公司的交易订单事务事实表设计过程中，粒度为订单，确定的维度包含：买家、卖家、商品名称、商品类目、发货地区、收货地区、订单时间等维度。

4. 确定事实。

作为度量业务过程的核心，事实通常为整型或浮点型的十进制数值。事实表应该包含与业务过程描述有关的所有事实，且事实的粒度要与所确定的事实表的粒度一致。例如，在下单业务过程中，需要包含商品ID、商品价格、购买数量。在支付业务过程中，需要包含支付金额、红包金额、积分金额。在收货业务过程中，需要包含确认收货金额等。

5. 关联维度。

   在确定维度时，包含了买卖家维度、商品维度、类目维度、收发货维度等。维度建模理论建议在事实表中只保存这些维表的外键， 而A公司电商交易事务事实表在维度建模基础之上做了进一步的优化，将买卖家星级、标签、店铺名称、商品类型、商品特征、商品属性、 类目层级等维度都关联到事实表中，提高对事实表进行过滤查询、统计聚合的效率。

明细数据层（DWD）规范

Dataphin中默认的事实表命名规范为： `fct_{业务过程缩写}[_{自定义表命名标签缩写}] _{di/df，单分区增量/全量标识}` 。单分区增量全量标识通常为：i表示增量，f表示全量。例如，fct_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fct_asale_itm_df（A电商商品快照事实表，日刷新全量）。

创建事实逻辑表

完成事实逻辑表的创建，详情请参见新建事实逻辑表。

查询事实表

逻辑表运维包含两个模块，一个是逻辑表任务，用来从逻辑表视角切入，为用户展现逻辑表内部任务关系。另一个是逻辑表实例，用于查看已运行的逻辑表任务包含的节点实例及其状态，详情请参见逻辑表任务和逻辑表实例。

1.4.6. 汇总数据层（DWS）

汇总数据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表。汇总数据层的一个表通常会对应一个统计粒度（维度或维度组合）及该粒度下若干派生指标。

汇总表设计原则

聚集是指针对原始明细粒度的数据进行汇总。DWS汇总数据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、该类目销售额Top10的商品名称、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，所以原子指标只能基于一张事实表定义，但是支持原子指标组合为衍生原子指标。
- 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层设计时还需遵循数据公用性原则。数据公用性需要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。

汇总表规范

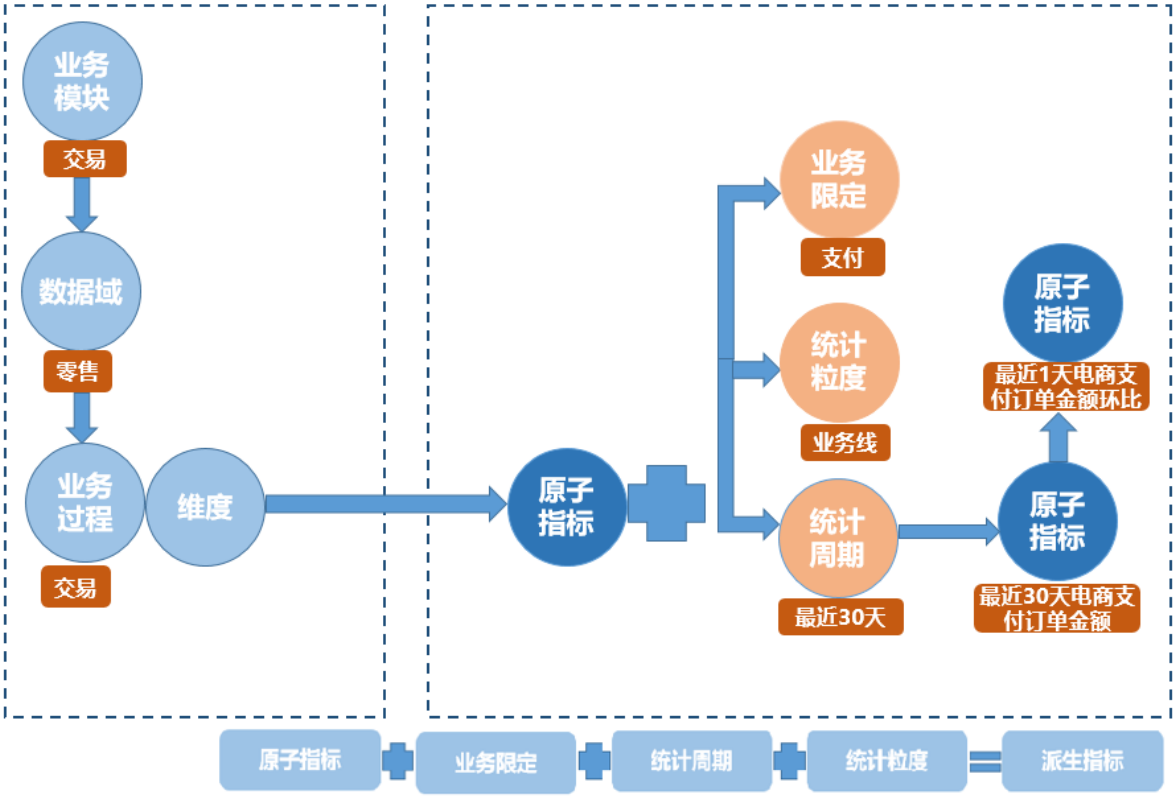
公共汇总表命名规范：dws_统计粒度。 举例如下：

- dws_report（report汇总表）
- dws_user（user汇总表）

创建汇总逻辑表

组成汇总表的统计指标有两种来源，具体如下：

- 系统按照相同统计粒度，自动汇聚。派生指标提交后，系统会自动生成新的汇总表。派生指标组成部分，如下图所示。



- 通过非派生指标的方式，创建汇总逻辑表，详情请参见新建汇总逻辑表。

查询汇总逻辑表

逻辑表运维包括**逻辑表任务**和**逻辑表实例**：

- **逻辑表任务**用于从逻辑表视角切入，为您展现逻辑表内部任务关系，详情请参见逻辑表任务。
- **逻辑表实例**用于查看已运行的逻辑表任务包含的节点实例及其状态，详情请参见逻辑表实例。

1.5. 主题式查询

主题式查询是基于Dataphin加工出的逻辑表，运用逻辑SQL对数据进行的AdHoc查询。本文为您介绍如何利用Dataphin进行主题式查询。

主题式查询简介

主题式查询是指面向业务主题的数据查询，屏蔽了物理模型中技术特性带来的影响，基于逻辑模型从业务视角出发对外提供查询服务。

主题式查询

完成主题式查询的构建，详情请参见查询和下载数据。

2.基于电商销售场景构建偏好标签

2.1. 概述

本文为您介绍本教程的背景信息、准备工作和操作流程。

背景信息

本教程带您体验Dataphin的数据萃取功能。基于Dataphin生产的数据资产，数据萃取模块能够识别并关联数据资产中的主数据（即贯穿各隔离业务的核心对象），提炼可直接应用的高价值标签数据，从而帮助企业构建自己的萃取数据中心。

本教程基于电商销售场景，为您介绍如何构建用户偏好标签，助力企业寻找业务发力点。

准备工作

> **说明** 完成数据萃取模块的初始化配置后，即可在萃取项目（Data_distill）下开发数据：
> - 如果您已有项目，则可以初始化配置数据萃取模块。
> - 如果您还没有项目，则无法初始化配置数据萃取模块。需要新建个测试项目（例如test）后，再开始初始化配置数据萃取模块。

- 新建计算源（basic）的MaxCompute（ODPS）项目，配置如下参数，其余参数保持默认，详情请参见创建工作空间。

参数	描述
工作空间名称	填写basic_odps。
选择计算引擎服务	选择MaxCompute。
实例显示名称	填写test。

- 新建计算源（Data_distill）的MaxCompute（ODPS）项目，配置如下参数，其余参数保持默认，详情请参见创建工作空间。

参数	描述
工作空间名称	填写Data_distill_odps。
选择计算引擎服务	选择MaxCompute。
实例显示名称	填写Mytest。

- 新建项目的计算源（basic），详情请参见新建MaxCompute计算源。

参数	描述
计算类型	默认为**MaxCompute**，不支持修改。
计算源名称	填写basic。
计算源描述	填写Baisc项目计算源。
Endpoint	默认为计算引擎的Endpoint，不支持修改。
Project Name	填写basic_odps。
Access ID	填写访问密钥中的AccessKey ID，您可以通过用户信息管理页面获取。

参数	描述
Access Key	访问密钥中的AccessKey Secret，您可以通过用户信息管理页面获取。

- 新建萃取项目的计算源（Data_distill），详情请参见新建MaxCompute计算源。

参数	描述
计算类型	默认为MaxCompute，不支持修改。
计算源名称	填写Data_distill。
计算源描述	填写Data_distill项目计算源。
Endpoint	默认为计算引擎的Endpoint，不支持修改。
Project Name	填写Data_distill_odps。
Access ID	填写访问密钥中的AccessKey ID，您可以通过用户信息管理页面获取。
Access Key	访问密钥中的AccessKey Secret，您可以通过用户信息管理页面获取。

- 新建Baisc模式的项目，配置如下参数，其余参数保持默认，详情请参见创建Basic项目空间。

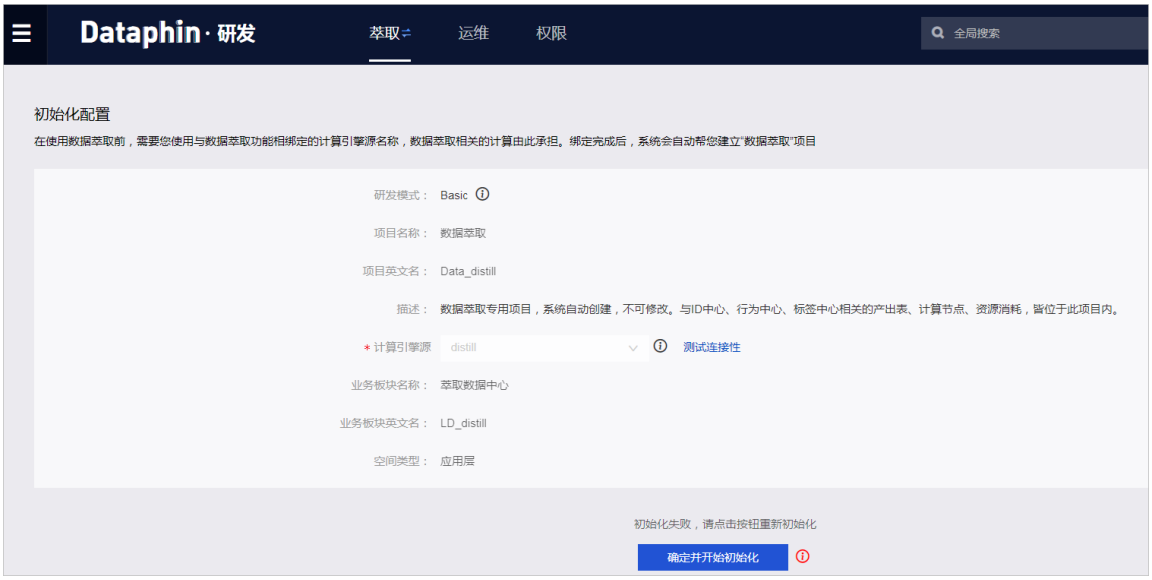
> **说明**
> - 如果您已有项目，则不需要新建项目。
> - 如果您还没有项目，则需要新建个测试项目（例如test）后，再开始初始化配置数据萃取模块。本教程以新建Baisc模式的项目为例。您也可以新建Dev-Prod模式的项目。

参数	描述
计算源	选择basic计算源。
英文名	输入test。
名称	输入测试。

- 初始化配置数据萃取模块。

> **说明** 完成数据萃取模块的初始化配置后，即可在萃取项目（Data_distill）下开发数据。

i. 登录Dataphin控制台。
ii. 在Dataphin控制台页面，选择工作区地域后，单击**进入Dataphin>>**。
   您也可以单击**快速开始相关工作**下的**数据研发**，快速进入数据开发模块。
iii. 在Dataphin页面，单击顶部菜单栏中的**研发**。
iv. 鼠标悬停至**开发**上后，单击**萃取**，进入**初始化配置**页面。



v. **计算引擎源**选择Data_distill后，单击**测试连接性**。

vi. 测试连接通过后，单击**确定并开始初始化**，初始化成功后即可在萃取模块开发数据。

操作流程

主流程	说明	操作指导
创建业务数据	通常，您的业务数据需通过创建同步任务或管道任务的方式，集成到Dataphin平台中的萃取项目中，以构建行为规则和标签。本教程为了让您快速熟悉标签构建的流程，采用代码任务的方式构建业务数据。	步骤一：创建数据表
创建行为元素	行为元素包括行为域、业务线、动作、对象和对象属性。本教程中的行为元素为： • 行为域：电商。 • 业务线：淘宝。 • 动作：购买、浏览和收藏。 • 对象：商品。 • 对象属性：商品价格和商品类目。	步骤二：创建行为元素
创建行为规则	行为规则根据行为和来源表唯一确定，将规范结构化的行为明细（行为域、业务线、动作、对象和对象属性）与实际数据进行匹配。本教程中行为规则包括： • 购买商品行为规则。 • 收藏商品行为规则。 • 浏览商品行为规则。	步骤三：创建行为规则
创建商品价格偏好标签	基于已创建的行为元素和行为规则，创建商品价格偏好的标签。	步骤五：创建商品价格偏好标签
创建商品类目价格偏好标签	基于已创建的行为元素和行为规则，创建商品类目偏好的标签。	步骤六：创建商品类目偏好标签
查询偏好数据	查询某个用户的偏好数据。	步骤七：查询偏好数据

2.2. 步骤一：创建数据表

本文为您介绍如何创建本教程中的业务数据表及ID倒排表。

背景信息

通常，您的业务数据需采用创建同步任务或管道任务的方式，导入Dataphin平台中的萃取项目，以构建行为规则和标签。本教程为了让您快速熟悉标签构建的流程，采用代码任务的方式构建业务数据。

本教程中的数据表包括：

- 用户浏览商品的表（ `s_item_view_d` ）。

column	type	comment
user_id	string	用户ID
item_id	string	商品ID
price	double	价格
cate_level1	string	商品类目
ds	string	分区

- 用户收藏商品的表（ `s_item_favor_d` ）。

column	type	comment
email	string	邮箱
item_id	string	商品ID
price	double	价格
cate_level1	string	商品类目
ds	string	分区

- 用户购买商品的表（ `s_item_buy_d` ）。

column	type	comment
phone_number	string	电话号码
item_id	string	商品ID
price	double	价格
cate_level1	string	商品类目
ds	string	分区

- ID倒排表（ `demo_id_mapping` ）。

column	type
key_type	string
key_id	string

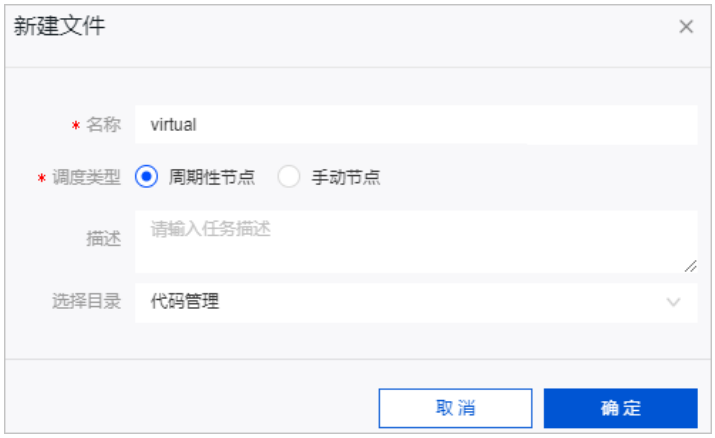
column	type
oneid	string
ds	string

步骤一：创建虚拟节点

虚拟节点是用于业务数据表配置调度参数时依赖的节点。

1. 登录Dataphin控制台。
2. 在Dataphin控制台页面，选择工作区地域后，单击**进入Dataphin>>**。
3. 进入**计算任务**页面。
   i. 在Dataphin首页，单击**研发**。
   ii. 单击项目后的图标，在项目列表中选择**Data_distill**项目。

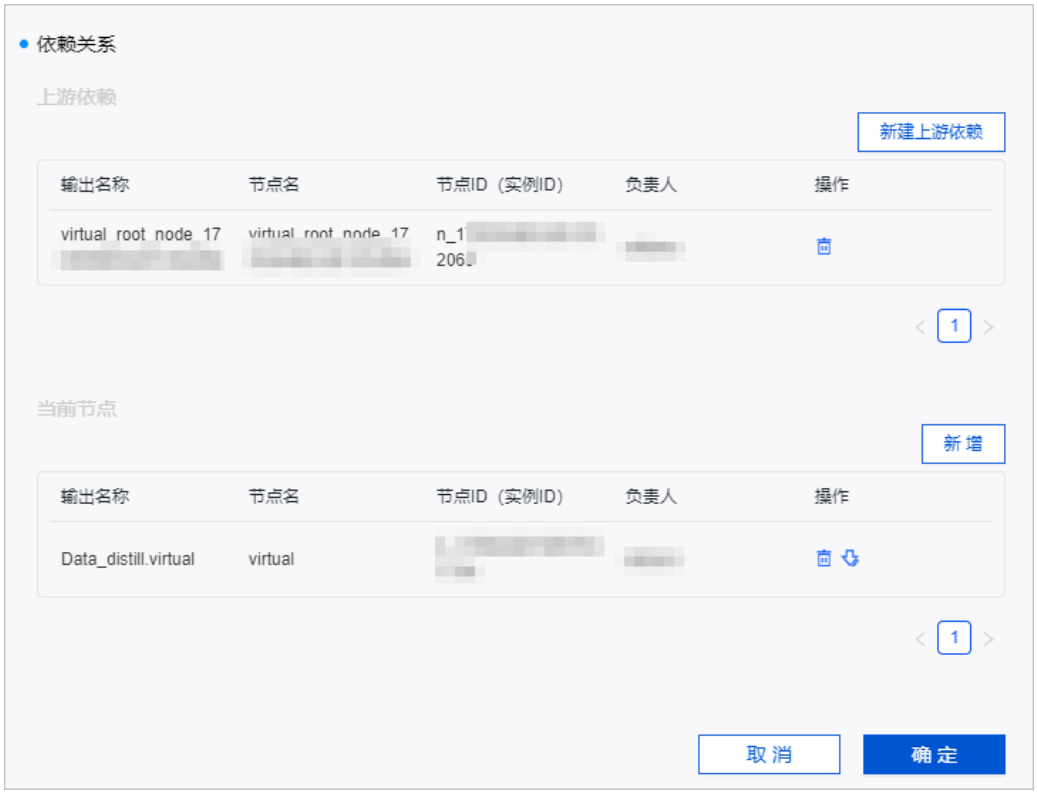
   如果系统中已有Dev项目和Prod项目，则单击图标后，需要单击**Basic**页签后，选择**Data_distill**项目。
   iii. 单击**数据处理**，系统默认进入**计算任务**页面。
4. 单击图标后，选择**VIRTUAL**。
5. 在**新建文件**对话框，配置参数。



参数	描述
名称	填写表的名称，本教程中填写virtual。
调度类型	选择**周期性节点**。
描述	填写对任务的简单描述。
选择目录	选择虚拟节点所属类目。

6. 完成参数配置后，单击**确定**。
7. 配置调度参数。
   i. 单击页面上方的**调度配置**。

ii. 配置**依赖关系**区域参数，其余参数均保持默认值。调度配置的更多内容，请参见调度配置。



参数	描述
上游依赖	添加该节点任务调度时依赖的上游节点： a. 单击**新建上游依赖**。 b. 在**新建上游依赖**对话框中，输入**virtual**搜索系统内置的虚拟节点。 c. 单击**确定新增**。
当前节点	设置当前节点的输出名称： a. 单击**新增**。 b. 在**新增当前节点输出**对话框中，填写输出名称为Data_distill.virtual，节点的输出名称是全局唯一的，且不区分大小写。 c. 单击**确定新增**。

iii. 单击**确定**，完成调度配置。

8. 单击页面右上的图标，完成虚拟节点的保存。
9. 提交虚拟节点（virtual）。
   i. 单击页面右上方的图标。
   ii. 在**提交备注**对话框，填写备注信息。
   iii. 单击**确定并提交**，提交成功的虚拟节点（virtual），即可进入生产环境。

步骤二：创建浏览商品的数据表（s_item_view_d）

1. 在**计算任务**页面，单击**计算任务**后的图标，选择**MAXC任务 > MAX_COMPUTE_SQL**。
2. 在**新建文件**对话框，配置参数。

参数	描述
名称	填写表的名称为s_item_view_d。
调度类型	选择**周期性节点**。
描述	填写对任务的简单描述。
选择目录	选择数据表的所属目录。

3. 完成参数配置后，单击**确定**。
4. 在**代码编写**页面，编写建表及向表中写入数据的SQL语句。

```
CREATE TABLE IF NOT EXISTS `s_item_view_d`
(
 user_id string COMMENT '用户ID',
 item_id string COMMENT '商品id',
 price double COMMENT '价格',
 cate_level1 string COMMENT '商品类目'
)
PARTITIONED BY (
 `ds` STRING
);
insert into table s_item_view_d partition (ds ='${bizdate}') values (1001,1,123,'女装'),(1001,1,123,'女装'),
(1001,1,123,'女装'),(1002,5,368,'零食'),(1002,4,429,'生鲜'),(1002,6,429,'零食'),(1002,3,399,'箱包'),
(1003,5,367,'家电'),(1003,6,728,'家具'),(1003,4,429,'家具'),
(1004,11,889,'保健品'),(1004,10,789,'动漫'),(1004,12,999,'保健品'),
(1005,5,2000,'箱包'),(1005,4,4999,'箱包'),(1005,5,2889,'箱包');
```

5. 单击页面右上方的**执行**，执行编写的建表语句。
6. 配置调度参数。
    i. 单击页面上方的**调度配置**。

ii. 配置**依赖关系**区域参数，其余参数均保持默认值。调度配置的更多内容，请参见调度配置。

参数	描述
上游依赖	添加该节点任务调度时依赖的上游节点： a. 单击**新建上游依赖**。 b. 在**新建上游依赖**对话框中，输入**virtual**搜索步骤一中创建的虚拟节点（Data_distill.virtual）。 c. 单击**确定新增**。
当前节点	设置当前节点的输出名称： a. 单击**新增**。 b. 在**新增当前节点输出**对话框中，填写输出名称为Data_distill.s_item_view_d，节点的输出名称是全局唯一的，且不区分大小写。 c. 单击**确定新增**。

iii. 单击**确定**，完成调度配置。

7. 单击页面右上的图标，完成数据表（s_item_view_d）的保存。
8. 提交数据表（s_item_view_d）。
   i. 单击页面右上方的图标。
   ii. 在**提交备注**对话框，填写备注信息。
   iii. 单击**确定并提交**，提交成功的数据表（s_item_view_d），即可进入生产环境。

步骤三：创建收藏商品的数据表（s_item_favor_d）

1. 在**计算任务**页面，单击**计算任务**后的图标，选择**MAXC任务 > MAX_COMPUTE_SQL**。
2. 在**新建文件**对话框，配置参数。

参数	描述
名称	填写表的名称为s_item_favor_d。
调度类型	选择**周期性节点**。
描述	填写对任务的简单描述。
选择目录	选择数据表的所属目录。

3. 完成参数配置后，单击**确定**。
4. 在**代码编写**页面，编写建表及向表中写入数据的SQL语句。

```
CREATE TABLE IF NOT EXISTS `s_item_favor_d`
(
 email string COMMENT '邮箱',
 item_id string COMMENT '商品id',
 price double COMMENT '价格',
 cate_level1 string COMMENT '商品类目'
)
PARTITIONED BY (
 `ds` string
);
insert into table s_item_favor_d partition (ds ='${bizdate}') values ('bochao3@hotmail.com','1',199,'女装'),('bochao3
@hotmail.com','2',199,'女装'),
('zjud02122@qq.com','4',429,'箱包'),('zjud02122@qq.com','3',399,'箱包'),
('never34@gmail.com','4',789,'家电'),('never34@gmail.com','5',799,'家电'),
('info664@163.com','10',789,'保健品'),('info664@163.com','11',889,'保健品'),
('772289335@outlook.com','5',2000,'箱包'),('772289335@outlook.com','4',4999,'箱包');
```

5. 单击页面右上方的**执行**，执行编写的建表语句。
6. 配置调度参数。
   i. 单击页面上方的**调度配置**。

ii. 配置**依赖关系**区域参数，其余参数均保持默认值。调度配置的更多内容，请参见调度配置。

参数	描述
上游依赖	添加该节点任务调度时依赖的上游节点： a. 单击**新建上游依赖**。 b. 在**新建上游依赖**对话框中，输入**virtual**搜索步骤一中创建的虚拟节点（Data_distill.virtual）。 **说明** 节点的输出名称是全局唯一的，且不区分大小写。 c. 单击**确定新增**。
当前节点	设置当前节点的输出名称： a. 单击**新增**。 b. 在**新增当前节点输出**对话框中，填写输出名称为Data_distill.s_item_favor_d。 c. 单击**确定新增**。

iii. 单击**确定**，完成调度配置。

7. 单击页面右上的图标，完成数据表（s_item_favor_d）的保存。
8. 提交数据表（s_item_favor_d）。
   i. 单击页面右上方的图标。
   ii. 在**提交备注**对话框，填写备注信息。
   iii. 单击**确定并提交**，提交成功的数据表（s_item_favor_d），即可进入生产环境。

步骤四：创建购买商品的数据表（s_item_buy_d）

1. 在**计算任务**页面，单击**计算任务**后的图标，选择**MAXC任务 > MAX_COMPUTE_SQL**。
2. 在**新建文件**对话框，配置参数。

参数	描述
名称	填写表的名称为s_item_buy_d。
调度类型	选择**周期性节点**。
描述	填写对任务的简单描述。
选择目录	选择数据表的所属目录。

3. 单击**确定**。
4. 在**代码编写**页面，编写建表及向表中写入数据的SQL语句。

```
CREATE TABLE IF NOT EXISTS `s_item_buy_d`
(
 phone_number string COMMENT '电话号码',
 item_id string COMMENT '商品id',
 price double COMMENT '价格',
 cate_level1 string COMMENT '商品类目'
)
PARTITIONED BY (
 `ds` STRING
);
insert into table s_item_buy_d partition (ds ='${bizdate}') values('15270159931','1',123,'女装'),
('18170159522','6',429,'零食'),
('13589374673','4',789,'家具'),
('17109872784','10',789,'动漫'),
('15300782675','5',2889,'箱包');
```

5. 单击页面右上方的**执行**，执行编写的建表语句。
6. 配置调度参数。
    i. 单击页面上方的**调度配置**。

ii. 配置**依赖关系**区域参数，其余参数均保持默认值。调度配置的更多内容，请参见调度配置。

参数	描述
上游依赖	添加该节点任务调度时依赖的上游节点： a. 单击**新建上游依赖**。 b. 在**新建上游依赖**对话框中，输入**virtual**搜索步骤一中创建的虚拟节点（Data_distill.virtual）。 **说明** 节点的输出名称是全局唯一的，且不区分大小写。 c. 单击**确定新增**。
当前节点	设置当前节点的输出名称： a. 单击**新增**。 b. 在**新增当前节点输出**对话框中，填写输出名称为Data_distill.s_item_buy_d。 c. 单击**确定新增**。

iii. 单击**确定**，完成调度配置。

7. 单击页面右上的图标，完成表的保存。
8. 提交数据表（s_item_buy_d）。
   i. 单击页面右上方的图标。
   ii. 在**提交备注**对话框，填写备注信息。
   iii. 单击**确定并提交**，提交成功的数据表（s_item_buy_d），即可进入生产环境。

步骤五：创建ID倒排表（demo_id_mapping）

ID倒排表基于OneID方法论，将同一用户的不同ID通过算法识别，用OneID连接起来，请参见新建并配置行为规则。

1. 在**计算任务**页面，单击**计算任务**后的图标，选择**MAXC任务 > MAX_COMPUTE_SQL**。
2. 在**新建文件**对话框，完成参数配置后，单击**确定**。

参数	描述
名称	填写表的名称为demo_id_mapping。
调度类型	选择**周期性节点**。
描述	填写对任务的简单描述。
选择目录	选择ID倒排表的所属目录。

3. 在**代码编写**页面，编写建表及向表中写入数据的SQL语句。

```
CREATE TABLE IF NOT EXISTS `demo_id_mapping`
(
 key_type string,
 key_id string,
 oneid string)
partitioned by (ds string);
insert into table demo_id_mapping PARTITION (ds ='${bizdate}')
select 'UserID', '1001', '001' union all
select 'UserID', '1002', '002' union all
select 'UserID', '1003', '003' union all
select 'UserID', '1004', '004' union all
select 'UserID', '1005', '005' union all
select 'Email', 'bochao1@hotmail.com', '001' union all
select 'Email', 'zjud0212@qq.com', '002' union all
select 'Email', 'never3@gmail.com', '003' union all
select 'Email', 'info64@163.com', '004' union all
select 'Email', '77228935@outlook.com', '005' union all
select 'Mobile', '15270159931', '001' union all
select 'Mobile', '18170159522', '002' union all
select 'Mobile', '13589374673', '003' union all
select 'Mobile', '17109872784', '004' union all
select 'Mobile', '15300782675', '005';
```

4. 单击页面右上方的**执行**，执行编写的建表语句。
5. 配置调度参数。
   i. 单击页面上方的**调度配置**。

ii. 配置**依赖关系**区域参数，其余参数均保持默认值。调度配置的更多内容，请参见调度配置。

参数	描述
上游依赖	添加该节点任务调度时依赖的上游节点： a. 单击**新建上游依赖**。 b. 在**新建上游依赖**对话框中，输入**virtual**搜索步骤一中创建的虚拟节点（Data_distill.virtual）。 **说明** 节点的输出名称是全局唯一的，且不区分大小写。 c. 单击**确定新增**。
当前节点	设置当前节点的输出名称： a. 单击**新增**。 b. 在**新增当前节点输出**对话框中，填写输出名称为Data_distill.demo_id_mapping。 c. 单击**确定新增**。

iii. 单击**确定**，完成调度配置。

6. 单击页面右上的图标，完成ID倒排表（demo_id_mapping）的保存。
7. 单击页面右上方的图标，提交ID倒排表（demo_id_mapping）。
8. 在**提交备注**对话框，填写备注信息。
9. 单击**确定并提交**，提交成功的ID倒排表（demo_id_mapping），即可进入生产环境。

2.3. 步骤二：创建行为元素

行为元素用于标准化定义与归类业务数据，同时是创建行为规则的基础元素。本文为您介绍如何创建行为元素。

背景信息

行为元素包括行为域、业务线、动作、对象和对象属性：

- 行为域用于聚合业务含义一致的行为数据，例如电商域、文娱域。
- 业务线将行为域中的行为数据进行归类，例如淘宝业务线、天猫业务线。
- 动作是行为主体发出的操作，例如购买、浏览。
- 对象是行为主体操作的具体事物，例如商品、电影。
- 对象属性是对对象的描述性信息，例如名称、品牌、年份。

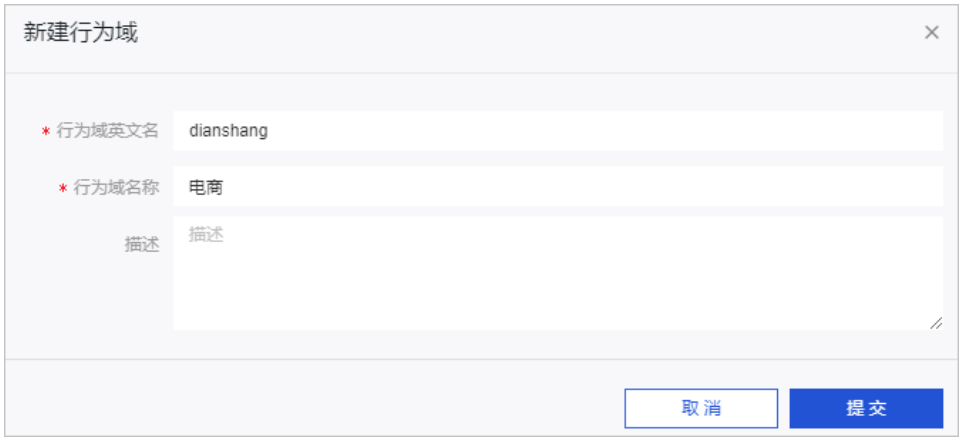
本教程中业务数据标准化定义包括：

- 行为域：电商。
- 业务线：淘宝。
- 动作：购买、浏览和收藏。
- 对象：商品。
- 对象属性：商品价格和商品类目。

步骤一：创建电商行为域

1. 登录Dataphin控制台。
2. 在Dataphin控制台页面，选择工作区地域后，单击**进入Dataphin>>**。
3. 进入**行为中心**。
   i. 在Dataphin产品首页，单击**研发**。
   ii. 在数据**开发**页面，鼠标悬停至**开发**上，单击**萃取**。
   iii. 在数据**萃取**页面，单击**行为中心**。
4. 单击左侧导航栏中的图标，进入**行为域&业务线**页面。
5. 单击**行为域&业务线**后的图标后，选择**行为域**。

6. 在**新建行为域**对话框，配置参数。

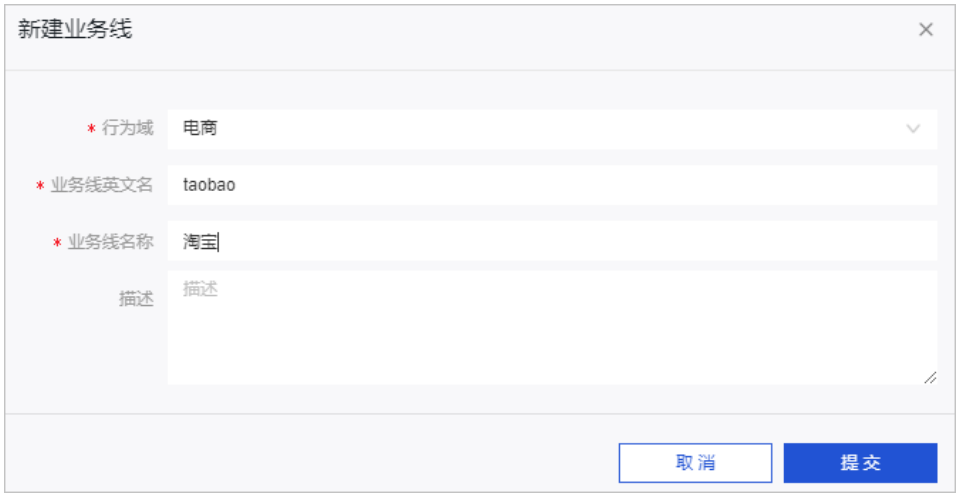


参数	描述
行为域英文名	填写 dianshang 。
行为域名称	填写**电商**。
描述	填写对电商的简单描述。

7. 单击**提交**。
8. 在**提交备注**对话框，填写备注信息。
9. 单击**确定并提交**，完成电商行为域的创建。

步骤二：创建淘宝业务线

1. 单击**行为域&业务线**后的图标，选择**业务线**。
2. 在**新建业务线**对话框，配置参数。

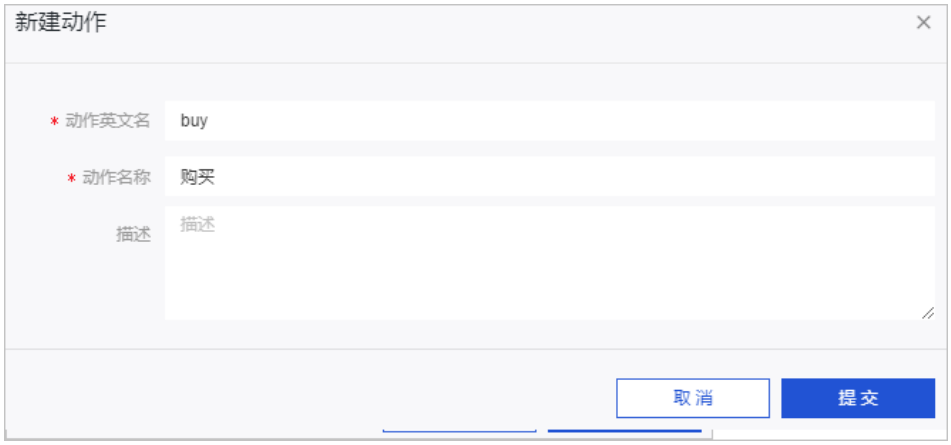


参数	描述
行为域	选择**电商**。
业务线英文名	填写 taobao 。
业务线名称	填写**淘宝**。
描述	填写对淘宝业务线的简单描述。

3. 单击**提交**。
4. 在**提交备注**对话框，填写备注信息。
5. 单击**确定并提交**，完成淘宝业务线的创建。

步骤三：创建购买动作

1. 单击左侧导航栏中的图标，进入**动作**页面。
2. 单击**动作**后的图标。
3. 在**新建动作**对话框，配置参数。

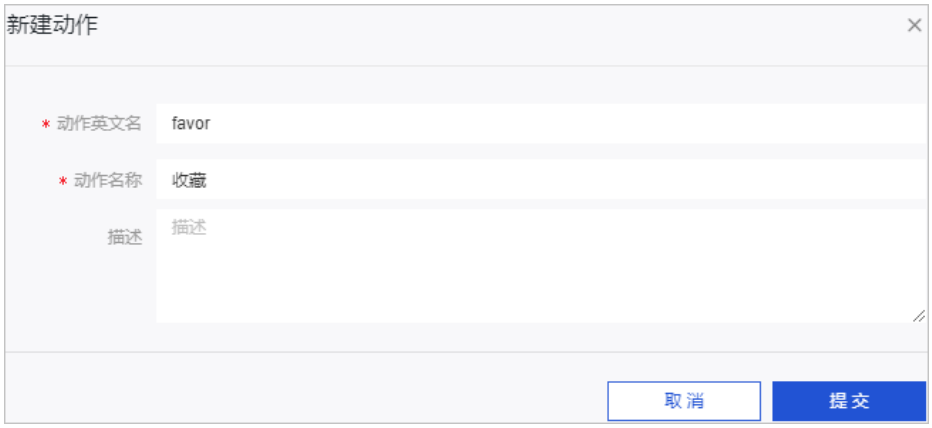


参数	描述
动作英文名	填写 `buy` 。
动作名称	填写**购买**。
描述	填写简单描述。

4. 单击**提交**。
5. 在**提交备注**对话框，填写备注信息。
6. 单击**确定并提交**，完成购买动作的创建。

步骤四：创建收藏动作

1. 单击左侧导航栏中的图标，进入**动作**页面。
2. 单击**动作**后的图标。
3. 在**新建动作**对话框，配置参数。

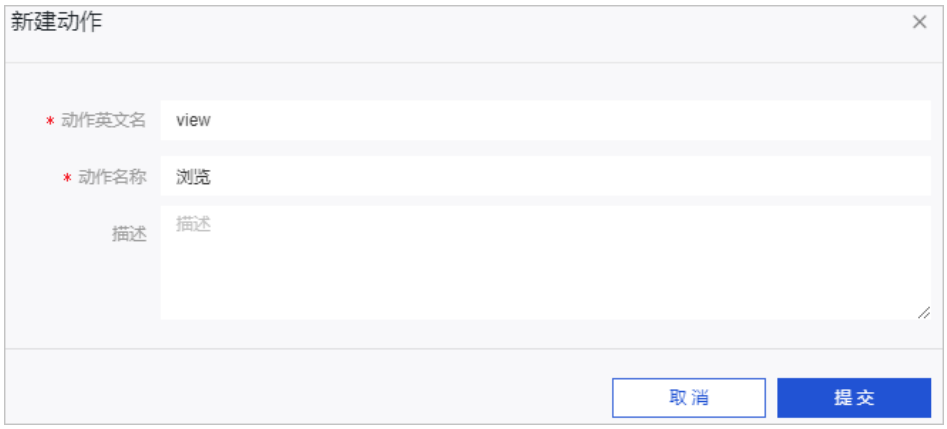


参数	描述
动作英文名	填写 favor 。
动作名称	填写**收藏**。
描述	填写简单描述。

4. 单击**提交**。
5. 在**提交备注**对话框，填写备注信息。
6. 单击**确定并提交**，完成收藏动作的创建。

步骤五：创建浏览动作

1. 单击左侧导航栏中的图标，进入**动作**页面。
2. 单击**动作**后的图标。
3. 在**新建动作**对话框，配置参数。



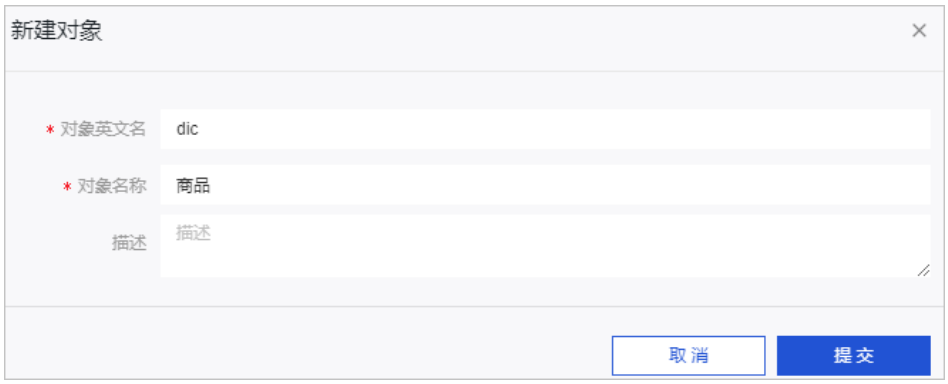
参数	描述
动作英文名	填写 view 。
动作名称	填写**浏览**。
描述	填写简单描述。

4. 单击**提交**。

5. 在**提交备注**对话框，填写备注信息。
6. 单击**确定并提交**，完成浏览动作的创建。

步骤六：创建商品对象

1. 单击左侧导航栏中的图标，进入**对象**页面。
2. 单击**对象**后的图标。
3. 在**新建对象**对话框，配置参数。

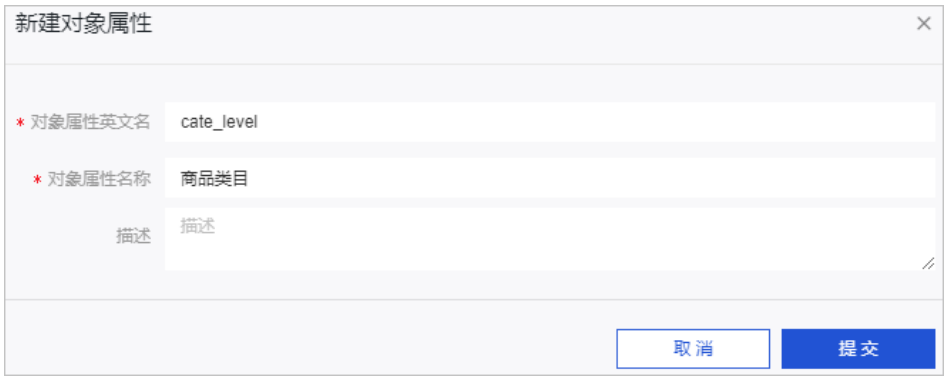


参数	描述
动作英文名	填写 dic 。
动作名称	填写为**商品**。
描述	填写简单描述。

4. 单击**提交**。
5. 在**提交备注**对话框，填写备注信息。
6. 单击**确定并提交**，完成商品对象的创建。

步骤七：创建商品类目对象属性

1. 单击左侧导航栏中的图标，进入**对象属性**页面。
2. 单击**对象属性**后的图标。
3. 在**新建对象属性**对话框，配置参数。

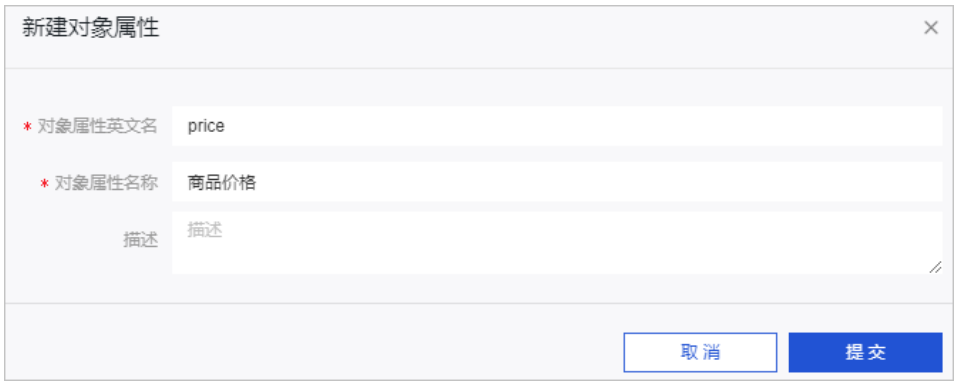


参数	描述
动作英文名	填写 cate_level 。
动作名称	填写为**商品类目**。
描述	填写简单描述。

4. 单击**提交**。
5. 在**提交备注**对话框，填写备注信息。
6. 单击**确定并提交**，完成商品类目对象属性的创建。

步骤八：创建商品价格对象属性

1. 单击左侧导航栏中的图标，进入**对象属性**页面。
2. 单击**对象属性**后的图标。
3. 在**新建对象属性**对话框，配置参数。



参数	描述
动作英文名	填写 price 。
动作名称	填写为**商品价格**。
描述	填写简单描述。

4. 单击**提交**。
5. 在**提交备注**对话框，填写备注信息。
6. 单击**确定并提交**，完成商品价格对象属性的创建。

2.4. 步骤三：创建行为规则

行为规则根据行为和来源表唯一确定，将规范结构化的行为明细（行为域、业务线、动作、对象）与实际数据进行匹配。本文为您介绍如何创建购买商品、浏览商品和收藏商品的行为规则。

前提条件

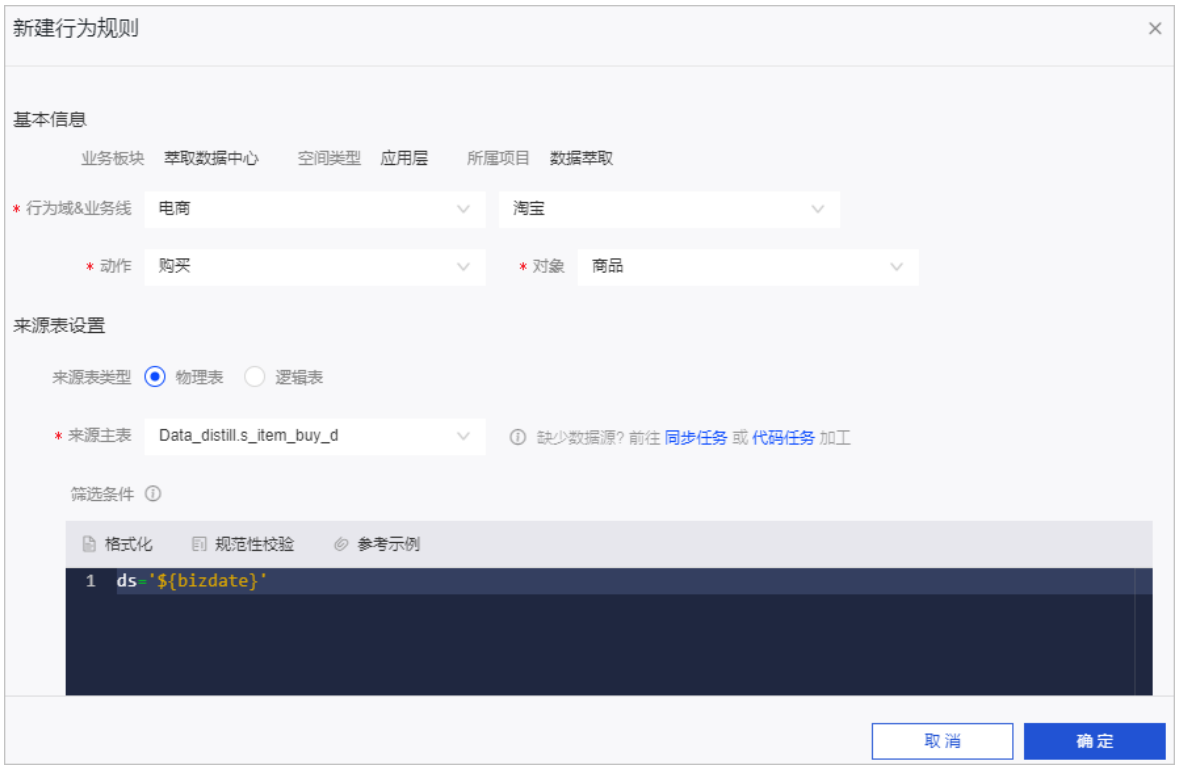
完成行为元素的创建，详情请参见步骤二：创建行为元素。

步骤一：创建购买商品行为规则

1. 登录Dataphin控制台。
2. 进入**行为中心**页面。

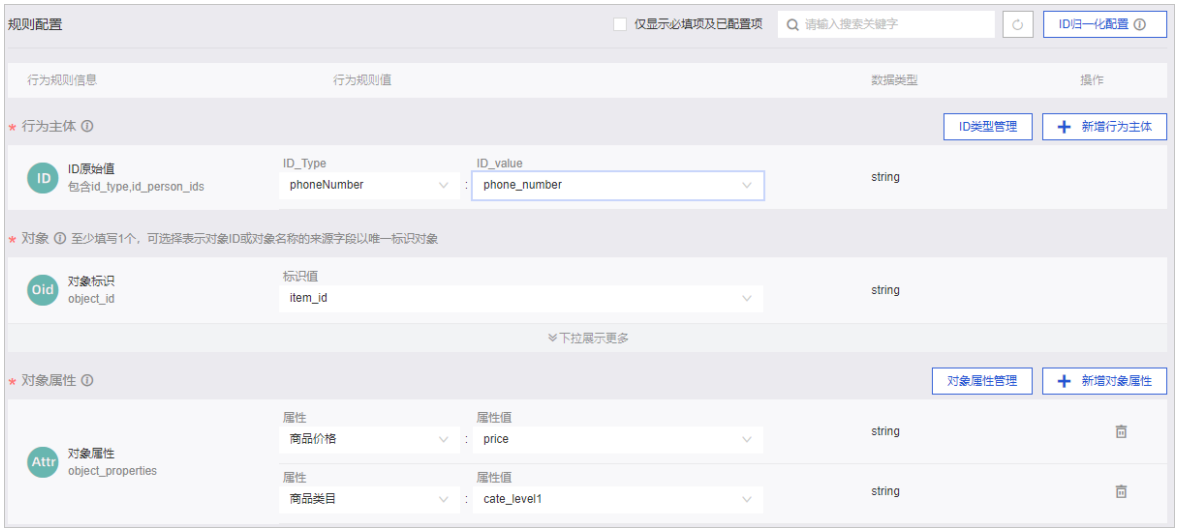
i. 在Dataphin控制台页面，选择工作区地域后，单击**进入Dataphin>>**。
ii. 在Dataphin产品首页，单击**研发**。
iii. 在数据**开发**页面，鼠标悬停至**开发**上，单击**萃取**。
iv. 在数据**萃取**页面，单击**行为中心**。
如果进入数据**萃取**页面，系统默认进入**行为中心**页面，则无需再次单击**行为中心**。

3. 单击左侧导航栏中的图标，进入**行为规则**页面。
如果进入**行为中心**页面，系统默认进入**行为规则**页面，则无需再次单击左侧导航栏中的图标。
4. 单击**行为规则**后的图标。
5. 在**新建行为规则**对话框，配置参数。



参数	描述
行为域	选择**电商**。
业务板块	选择**淘宝**。
动作	选择**购买**。
对象	选择**商品**。
来源表类型	选择**物理表**。
来源主表	选择Data_distill.s_item_buy_d。
筛选条件	输入ds='${bizdate}'。

6. 单击**确定**。
7. 在**新建行为规则：Data_distill.s_item_buy_d**页面，配置参数。



参数	描述
ID归一化配置	引入ID倒排表： i. 单击**ID归一化配置**。 ii. 在**ID归一化配置**对话框，选择ID倒排表为Data_distill.demo_id_mapping，同时打开**ID归一化**开关。 iii. 单击**确定并初始化**。 **说明** 仅支持选择当前项目内的ID倒排表。
行为主体	**ID_Type**选择Mobile，**ID_value**选择phone_number。
对象	**标识值**选择为item_id。
对象属性	添加**商品价格**和**商品类目**的对象属性： i. **属性**选择商品价格，**属性值**选择price。 ii. 单击**新增对象属性**。 iii. **属性**选择商品类目，**属性值**选择cate_level1。

8. 配置调度参数。
   i. 单击页面上方的**调度配置**。

ii. 配置**依赖关系**区域参数，其余参数均保持默认值。调度配置的更多内容，请参见调度配置。

参数	描述
上游依赖	添加该节点任务调度时依赖的上游节点： a. 单击**新建上游依赖**。 b. 在**新建上游依赖**对话框中，在搜索框中输入Data_distill.s_item_buy_d，搜索到Data_distill.s_item_buy_d节点。 c. 单击**确定新增**。

iii. 单击**确定**，完成调度配置。

9. 保存、预览并提交行为规则。

i. 单击页面右上方的图标，保存当前行为规则的配置。

ii. 单击页面右上方的图标，预览行为规则。

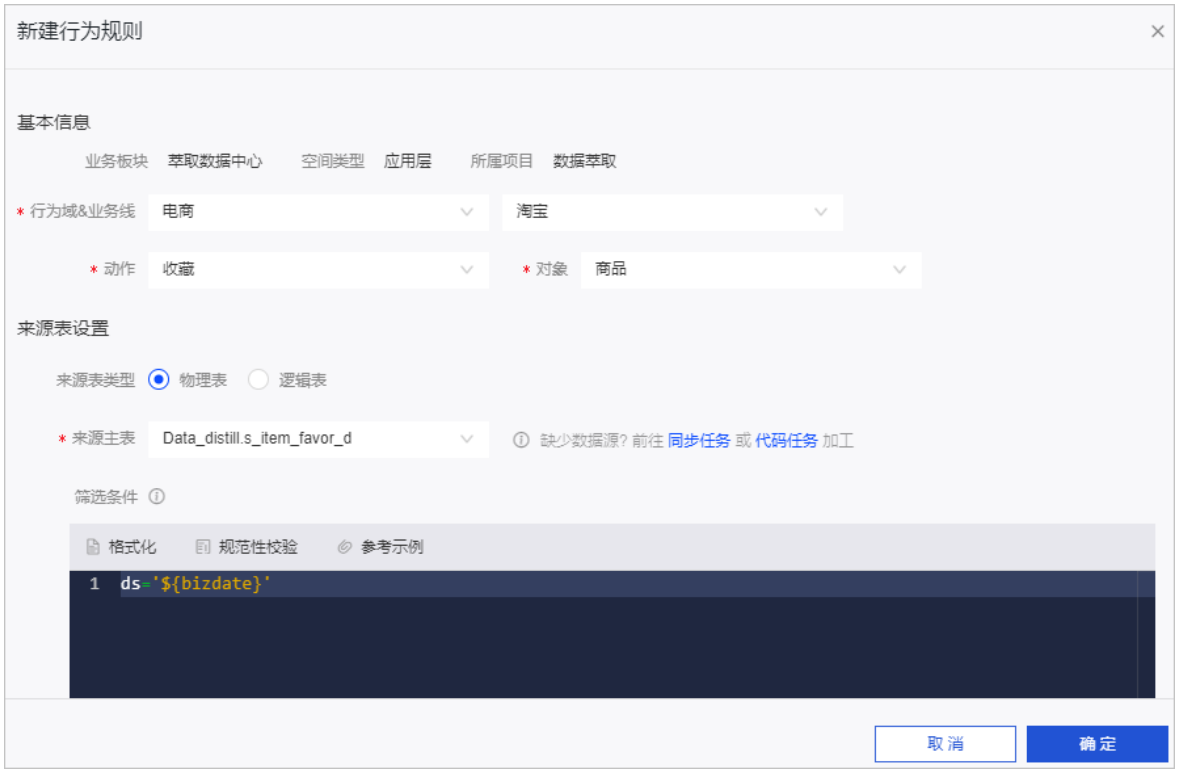
iii. 单击页面右上方的图标，完成行为规则的提交。

iv. 在**提交备注**对话框，填写备注信息。

v. 单击**确定并提交**，提交成功的行为规则，即可进入生产环境。

步骤二：创建收藏商品行为规则

1. 在**行为规则**页面，单击**行为规则**后的图标。
2. 在**新建行为规则**对话框，配置参数。



参数	描述
行为域	选择**电商**。
业务板块	选择**淘宝**。
动作	选择**收藏**。
对象	选择**商品**。
来源表类型	选择**物理表**。
来源主表	选择Data_distill.s_item_favor_d。
筛选条件	输入ds='${bizdate}'。

3. 单击**确定**。
4. 在**新建行为规则：Data_distill.s_item_favor_d**页面，配置参数。

参数	描述
ID归一化配置	确认ID倒排表： i. 单击**ID归一化配置**。 ii. 在**ID归一化配置**对话框，确认当前ID倒排表是否为Data_distill.demo_id_mapping。 如果不是，则单击ID倒排表显示区域，选择Data_distill.demo_id_mapping。 iii. 单击**确定**。
行为主体	**ID_Type**选择Email，**ID_value**选择email。
对象	**标识值**选择为item_id。
对象属性	添加**商品价格**和**商品类目**的对象属性： i. **属性**选择商品价格，**属性值**选择 price 。 ii. 单击**新增对象属性**。 iii. **属性**选择商品类目，**属性值**选择cate_level1。

5. 配置调度参数。
   i. 单击页面上方的**调度配置**。

ii. 配置**依赖关系**区域参数，其余参数均保持默认值。调度配置的更多内容，请参见调度配置。

参数	描述
上游依赖	添加该节点任务调度时依赖的上游节点： a. 单击**新建上游依赖**。 b. 在**新建上游依赖**对话框中，在搜索框中输入Data_distill.s_item_favor_d，搜索到Data_distill.s_item_favor_d节点。 c. 单击**确定新增**。

iii. 单击**确定**，完成调度配置。

6. 保存、预览并提交行为规则。

i. 单击页面右上方的图标，保存当前行为规则的配置。

ii. 单击页面右上方的图标，预览行为规则。

iii. 单击页面右上方的图标，完成行为规则的提交。

iv. 在**提交备注**对话框，填写备注信息。

v. 单击**确定并提交**，提交成功的行为规则，即可进入生产环境。

步骤三：创建浏览商品行为规则

1. 在**行为规则**页面，单击**行为规则**后的图标。
2. 在**新建行为规则**对话框，配置参数。

参数	描述
行为域	选择**电商**。
业务板块	选择**淘宝**。
动作	选择**浏览**。
对象	选择**商品**。
来源表类型	选择**物理表**。
来源主表	选择Data_distill.s_item_view_d。
筛选条件	输入ds='${bizdate}'。

3. 完成参数配置后，单击**确定**。
4. 在**新建行为规则：Data_distill.s_item_view_d**对话框，配置参数。

参数	描述
ID归一化配置	确认ID倒排表： i. 单击**ID归一化配置**。 ii. 在**ID归一化配置**对话框，确认当前ID倒排表是否为Data_distill.demo_id_mapping。 如果不是，则单击ID倒排表显示区域，选择Data_distill.demo_id_mapping。 iii. 单击**确定**。
行为主体	**ID_Type**选择UserID，**ID_value**选择user_id。
对象	**标识值**选择为item_id。
对象属性	添加**商品价格**和**商品类目**的对象属性： i. **属性**选择商品价格，**属性值**选择price。 ii. 单击**新增对象属性**。 iii. **属性**选择商品类目，**属性值**选择cate_level1。

5. 配置调度参数。
   i. 单击页面上方的**调度配置**。

ii. 配置**依赖关系**区域参数，其余参数均保持默认值。调度配置的更多内容，请参见调度配置。

参数	描述
上游依赖	添加该节点任务调度时依赖的上游节点： a. 单击**新建上游依赖**。 b. 在**新建上游依赖**对话框中，在搜索框中输入Data_distill.s_item_view_d，搜索到Data_distill.s_item_view_d节点。 c. 单击**确定新增**。

iii. 单击**确定**，完成调度配置。

6. 保存、预览并提交行为规则。

i. 单击页面右上方的图标，保存当前行为规则的配置。

ii. 单击页面右上方的图标，预览行为规则。

iii. 单击页面右上方的图标，完成行为规则的提交。

iv. 在**提交备注**对话框，填写备注信息。

v. 单击**确定并提交**，提交成功的行为规则，即可进入生产环境。

2.5. 步骤四：补数据

本文为您介绍如何为行为规则任务补数据。

前提条件

完成行为规则的创建，详情请参见步骤三：创建行为规则。

背景信息

通常，您构建的行为规则任务会参与生产环境的调度。本教程为了让您快速熟悉构建偏好标签的流程，采用补数据的方式，构建数据行为规则运行生成的数据。

补数据（电商-淘宝-收藏-商品）

1. 登录Dataphin控制台。
2. 进入**行为中心**页面。
   i. 在Dataphin控制台页面，选择工作区地域后，单击**进入Dataphin>>**。
   ii. 在Dataphin产品首页，单击**研发**。
   iii. 在数据**开发**页面，鼠标悬停至**开发**上，单击**萃取**。
   iv. 在数据**萃取**页面，单击**行为中心**。
   如果进入数据**萃取**页面，系统默认进入**行为中心**页面，则无需再次单击**行为中心**。
3. 单击左侧导航栏中的图标。

   如果进入**行为中心**页签，系统默认进入**行为规则**页面，则无需再次单击左侧导航栏中的图标。
4. 在**行为规则**页面，打开电商-淘宝-收藏-商品后，鼠标悬停至**Data_distill.s_item_favor_d**后，再将鼠标悬停至图标，单击**补数据**。
5. 在**行为规则补数据**对话框，**补数据生效时段**的**开始时间**和**结束时间**均配置为2020-03-24，**实例名称**配置为电商_淘宝_收藏_商品。
6. 单击**确定**。

补数据（电商-淘宝-浏览-商品）

1. 在**行为规则**页面，打开电商-淘宝-浏览-商品后，鼠标悬停至**Data_distill.s_item_view_d**后，再将鼠标悬停至图标，单击**补数据**。
2. 在**行为规则补数据**对话框，**补数据生效时段**的**开始时间**和**结束时间**均配置为2020-03-24，**实例名称**配置为电商_淘宝_浏览_商品。
3. 单击**确定**。

补数据（电商-淘宝-购买-商品）

1. 在**行为规则**页面，打开电商-淘宝-购买-商品后，鼠标悬停至**Data_distill.s_item_buy_d**后，再将鼠标悬停至图标，单击**补数据**。
2. 在**行为规则补数据**对话框，**补数据生效时段**的**开始时间**和**结束时间**均配置为2020-03-24，**实例名称**配置为电商_淘宝_购买_商品。
3. 单击**确定**。

2.6. 步骤五：创建商品价格偏好标签

本文为您介绍如何基于已创建的行为元素和行为规则，创建商品价格偏好的标签。

前提条件

完成购买商品、浏览商品和收藏商品的行为规则的补数据，详情请参见步骤四：补数据。

步骤一：创建基础指标

1. 登录Dataphin控制台。
2. 在Dataphin控制台页面，选择工作区地域后，单击**进入Dataphin>>**。
3. 在Dataphin产品首页，单击**研发**。
4. 在数据**开发**页面，鼠标悬停至**开发**上，单击**萃取**。
5. 在数据**萃取**页面，单击**标签中心**。

   在**标签中心**页签，系统默认进入**工厂标签**页面。
6. 单击**工厂标签**后的图标。
7. 在**新建工厂标签**对话框，配置参数。

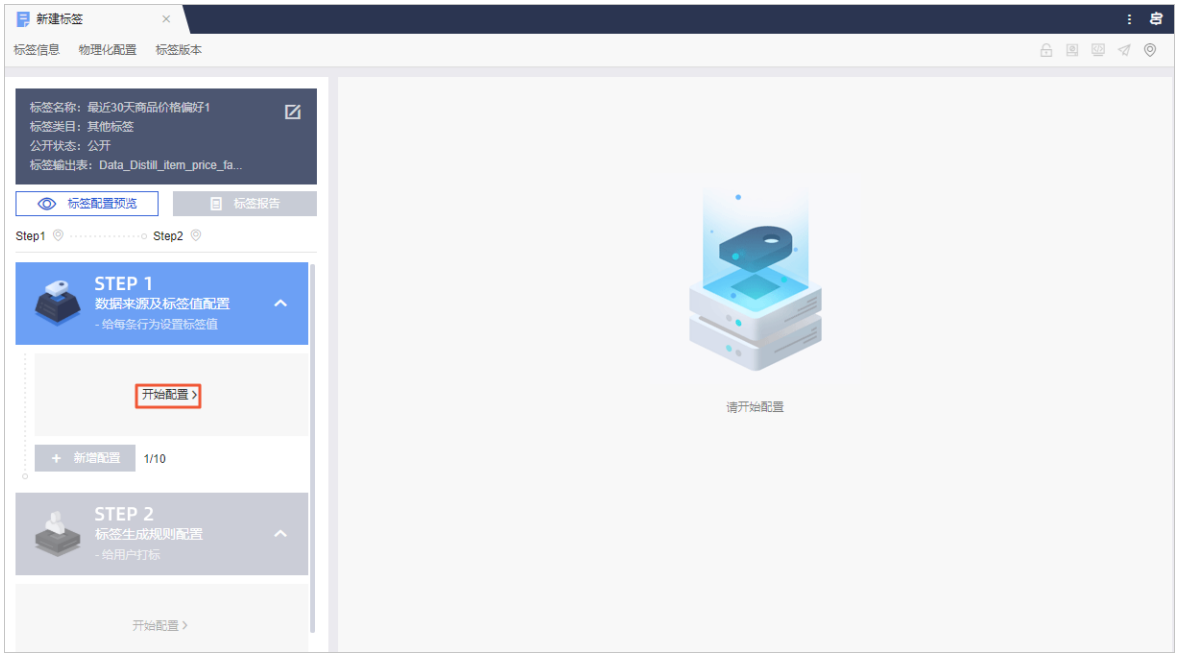
参数	描述
标签英文名	填写**标签英文名**为item_price_favor。
标签名称	填写**标签名称**为**最近30天商品价格偏好**。
描述	填写对标签的简单描述。
所属类目	选择**所属类目**为**其他标签**。
公开状态	选择**公开状态**为**公开**。
结果表名	填写输出表名为item_price_prefer。

8. 完成参数配置后，单击**确定**。

步骤二：配置数据来源、标签值和标签生成规则

本教程中需要配置购买商品、收藏商品和浏览商品的数据来源及标签值。

1. 在**新建标签**页面，单击**数据来源及标签配置**下方的**开始配置**，进入数据来源配置页面。



2. 配置购买商品的数据来源及标签值。

i. 在**数据来源配置**页面，配置参数。

参数	描述
按行为域筛选	选择**电商**。
按动作对象筛选	选择**购买-商品**。
按共有对象属性筛选	保持默认，无需筛选对象属性。
数据来源-时间跨度	保持默认，**30天**的**时间跨度**。

ii. 单击**下一步**。

iii. 在**标签值配置**页面，配置参数。

参数	描述
打标方式	选择**对象属性映射打标**。
共有对象属性	选择**商品价格**。

参数	描述
选择映射方式	选择**设计映射规则**，同时设计的映射规则如下： ■ 价格**小于200**的标签值为**低价格**。 ■ 价格**小于500**的标签值为**中价格**。 ■ **剩余所有**价格的标签值为**高价格**。 系统会按照设定的规则顺序逐一匹配生效，如果匹配成功，则不再执行匹配操作。根据上述设定的规则匹配后的效果如下： ■ price<200，标签值为低价格。 ■ 200≤price<500，标签值为中价格。 ■ 500≤price，标签值为高价格。

iv. 单击**确定**，完成购买商品行为规则的配置。

3. 单击**数据来源及标签值配置**下方的**新增配置**，新增浏览商品的数据来源及标签值。
4. 配置浏览商品的数据来源及标签值。

i. 在**数据来源配置**页面，配置参数。

参数	描述
按行为域筛选	选择**电商**。
按动作对象筛选	选择**浏览-商品**。
按共有对象属性筛选	保持默认，无需筛选对象属性。
数据来源-时间跨度	保持默认，**30天**的**时间跨度**。

ii. 单击**下一步**。

iii. 在**标签值配置**页面，配置参数。

参数	描述
打标方式	选择**对象属性映射打标**。
共有对象属性	选择**商品价格**。

参数	描述
选择映射方式	选择**设计映射规则**，同时设计的映射规则如下： ■ 价格**小于200**的标签值为**低价格**。 ■ 价格**小于500**的标签值为**中价格**。 ■ **剩余所有**价格的标签值为**高价格**。 系统会按照设定的规则顺序逐一匹配生效，如果匹配成功，则不再执行匹配操作。根据上述设定的规则匹配后的效果如下： ■ price<200，标签值为低价格。 ■ 200≤price<500，标签值为中价格。 ■ 500≤price，标签值为高价格。

iv. 单击**确定**，完成浏览商品行为规则来源数据及标签值的配置。

5. 单击**数据来源及标签值配置**下方的**新增配置**，新增收藏商品的数据来源及标签值。
6. 配置收藏商品数据来源及标签值。

i. 在**数据来源配置**页面，配置参数。

参数	描述
按行为域筛选	选择**电商**。
按动作对象筛选	选择**收藏-商品**。
按共有对象属性筛选	保持默认，无需筛选对象属性。
数据来源-时间跨度	保持默认，**30天**的**时间跨度**。

ii. 单击**下一步**。

iii. 在**标签值配置**页面，配置参数。

参数	描述
打标方式	选择**对象属性映射打标**。
共有对象属性	选择**商品价格**。

参数	描述
选择映射方式	选择**设计映射规则**，同时设计的映射规则如下： ■ 价格**小于200**的标签值为**低价格**。 ■ 价格**小于500**的标签值为**中价格**。 ■ **剩余所有**价格的标签值为**高价格**。 系统会按照设定的规则顺序逐一匹配生效，如果匹配成功，则不再执行匹配操作。根据上述设定的规则匹配后的效果如下： ■ price<200，标签值为低价格。 ■ 200≤price<500，标签值为中价格。 ■ 500≤price，标签值为高价格。

iv. 单击**标签生成规则配置**。

7. 在**标签生成规则配置**页面，配置参数。

参数	描述
返回标签值个数	填写为**5**。

参数	描述
选择衰减时间周期	选择为**7**。
请选择衰减时间曲线	通常认为，您的购买、收藏和浏览价格偏好随着时间变化不大，所以**请选择衰减时间曲线**选中**平滑衰减**。
标签生成规则-行为权重分配	行为包括购买商品、浏览商品和收藏商品的数据来源，所有行为的权重总和为1。设置三个行为的权重分别为： ○ **电商-淘宝-购买-商品**设置为**0.6**。 ○ **电商-淘宝-浏览-商品**设置为**0.3**。 ○ **电商-淘宝-收藏-商品**设置为**0.1**。
标签生成规则-覆盖用户范围	本教程中设置的覆盖用户范围为**0%~100%**。覆盖用户百分比和覆盖用户范围的详细说明如下： ○ 覆盖用户百分比：行为发生次数在某个值以下的用户数量占所有用户数量的百分比。例如，购买保湿类护肤品次数在3次以下的用户数量占所有购买保湿类护肤品用户数量的10%。 ○ 覆盖用户范围：限制用户打标的范围，防止行为发生次数过少或过多影响数据分析的准确度。 ■ 如果没有设置覆盖用户范围，则所有的行为记录都会参与计算。 ■ 如果设置了覆盖用户范围，则会按照设定的百分比过滤掉部分用户的行为记录。 例如，行为为购买保湿类护肤品，10%的用户购买次数在3次以下，90%的用户购买次数在100次以下。如果选择范围为10%-90%，则只对购买次数在3次以上100次以下的用户打标，防止购买次数过少（3次以下）或过多（100次以上）影响数据分析的准确度。

8. 完成配置后单击**确定**。
9. 保存、测试并提交标签。
   i. 单击页面右上方的图标，保存当前标签的配置。
   ii. 单击页面右上方的图标，测试运行标签。
   iii. 单击页面右上方的图标，提交最近30天商品价格偏好的标签。
   iv. 在**提交备注**对话框，填写备注信息。
   v. 单击**确定并提交**，提交成功的标签，即可进入生产环境。

2.7. 步骤六：创建商品类目偏好标签

本文为您介绍如何基于行为元素和行为规则，创建商品类目偏好的标签。

前提条件

完成购买商品、浏览商品和收藏商品的行为规则的创建，详情请参见步骤三：创建行为规则。

步骤一：创建基础指标

1. 登录Dataphin控制台。
2. 在Dataphin控制台页面，选择工作区地域后，单击**进入Dataphin>>**。
3. 在Dataphin产品首页，单击**研发**。
4. 在数据**开发**页面，鼠标悬停至**开发**上，单击**萃取**。
5. 在数据**萃取**页面，单击**标签中心**。
   在**标签中心**页签，系统默认进入**工厂标签**页面。

6. 在**工厂标签**页面，单击**工厂标签**后的图标。
7. 在**新建工厂标签**对话框，配置参数。

参数	描述
标签英文名	填写**标签英文名**为item_cate_prefer。
标签名称	填写**标签名称**为**最近30天商品类目偏好**。
描述	填写对标签的简单描述。
所属类目	选择**所属类目**为**其他标签**。
公开状态	选择**公开状态**为**公开**。
结果表名	填写输出表名为item_cate_prefer。

8. 单击**确定**。

步骤二：配置数据来源、标签值和标签生成规则

本教程中需要配置购买商品、收藏商品和浏览商品的数据来源及标签值。

1. 在**新建标签**页面，单击**数据来源及标签配置**下方的**开始配置**，进入数据来源配置页面。
2. 配置购买商品的数据来源及标签值。

i. 在**数据来源配置**页面，配置参数。

参数	描述
按行为域筛选	选择**电商**。
按动作对象筛选	选择**购买-商品**。
按共有对象属性筛选	保持默认，无需筛选对象属性。
数据来源-时间跨度	保持默认，**30天**的**时间跨度**。

ii. 单击**下一步**。

iii. 在**标签值配置**页面，配置参数。

参数	描述
打标方式	选择**对象属性映射打标**。
共有对象属性	选择**商品类目**。
选择映射方式	选择**属性值应用为标签**。

iv. 单击**确定**，完成购买商品的数据来源及标签值的配置。

3. 单击**数据来源及标签值配置**下方的**新增配置**，新增浏览商品的数据来源及标签值。
4. 配置浏览商品的数据来源及标签值。

i. 在**数据来源配置**页面，配置参数。

参数	描述
按行为域筛选	选择**电商**。
按动作对象筛选	选择**浏览-商品**。
按共有对象属性筛选	保持默认，无需筛选对象属性。
数据来源-时间跨度	保持默认，**30天**的**时间跨度**。

ii. 单击**下一步**。

iii. 在**标签值配置**页面，配置参数。

参数	描述
打标方式	选择**对象属性映射打标**。
共有对象属性	选择**商品类目**。
选择映射方式	选择**属性值应用为标签**。

iv. 单击**确定**，完成浏览商品的数据来源及标签值的配置。

5. 单击**数据来源及标签值配置**下方的**新增配置**，新增收藏商品的数据来源及标签值。
6. 配置商品的收藏数据来源及标签值。

i. 在**数据来源配置**页面，配置参数。

参数	描述
按行为域筛选	选择**电商**。
按动作对象筛选	选择**收藏-商品**。
按共有对象属性筛选	保持默认，无需筛选对象属性。
数据来源-时间跨度	保持默认，**30天**的**时间跨度**。

ii. 单击**下一步**。

iii. 在**标签值配置**页面，配置参数。

参数	描述
打标方式	选择**对象属性映射打标**。
共有对象属性	选择**商品类目**。
选择映射方式	选择**属性值应用为标签**。

iv. 单击**标签生成规则配置**。

7. 在**标签生成规则配置**页面，配置参数。

参数	描述
返回标签值个数	填写为**5**。
选择衰减时间周期	选择为**7**。
请选择衰减时间曲线	通常认为，您的购买、收藏和浏览类目偏好随着时间变化不大，所以**请选择衰减时间曲线**选中为**平滑衰减**。
标签生成规则-行为权重分配	行为包括购买商品、浏览商品和收藏商品的数据来源，所有行为的权重总和为1。三个行为的权重分别设置为： ○ **电商-淘宝-购买-商品**设置为**0.6**。 ○ **电商-淘宝-浏览-商品**设置为**0.3**。 ○ **电商-淘宝-收藏-商品**设置为**0.1**。

参数	描述
标签生成规则-覆盖用户范围	本教程中设置的覆盖用户范围为**0%~100%**。覆盖用户百分比和覆盖用户范围的详细说明如下： ○ 覆盖用户百分比：行为发生次数在某个值以下的用户数量占所有用户数量的百分比。例如，购买保湿类护肤品次数在3次以下的用户数量占所有购买保湿类护肤品用户数量的10%。 ○ 覆盖用户范围：限制用户打标的范围，防止行为发生次数过少或过多影响数据分析的准确度。 ■ 如果没有设置覆盖用户范围，则所有的行为记录都会参与计算。 ■ 如果设置了覆盖用户范围，则会按照设定的百分比过滤掉部分用户的行为记录。 例如，行为为购买保湿类护肤品，10%的用户购买次数在3次以下，90%的用户购买次数在100次以下。如果选择范围为10%-90%，则只对购买次数在3次以上100次以下的用户打标，防止购买次数过少（3次以下）或过多（100次以上）影响数据分析的准确度。

8. 完成配置后单击**确定**。
9. 保存、测试并提交标签。
   i. 单击页面右上方的图标，保存当前标签的配置。
   ii. 单击页面右上方的图标，测试运行标签。
   iii. 单击页面右上方的图标，提交最近30天商品类目偏好的标签。
   iv. 在**提交备注**对话框，填写备注信息。
   v. 单击**确定并提交**，提交成功的标签，即可进入生产环境。

2.8. 步骤七：查询偏好数据

本文为您介绍如何查询商品价格偏好和商品类目偏好的数据。

前提条件

- 完成商品价格偏好标签的创建，详情请参见步骤五：创建商品价格偏好标签。
- 完成商品类目偏好标签的创建，详情请参见步骤六：创建商品类目偏好标签。

操作步骤

1. 登录Dataphin控制台。
2. 在Dataphin控制台页面，选择工作区地域后，单击**进入Dataphin>>**。
3. 进入**即席查询**页面。
   i. 在Dataphin产品首页，单击**研发**。
   ii. 在数据**开发**页面，单击项目后的图标，在项目列表中选择Data_distill项目。
   如果系统中已有Dev项目和Prod项目，则单击图标后，需要单击Basic页签后，选择Data_distill项目。
   iii. 单击**即席查询**。
4. 新建即席查询文件夹。
   i. 单击即席查询后的图标。

ii. 在**新建文件夹**对话框，配置参数。

参数	描述
名称	填写test。
选择目录	选择文件夹所在的目录。

iii. 单击**确定**，完成文件夹的创建。

5. 单击**即席查询**后的图标。

6. 在**新建文件**对话框，配置参数。

参数	描述
名称	填写**即席查询**的名称。
描述	填写简单的描述。
选择目录	选择已创建的目录。

7. 单击**确定**。
8. 在**代码编写**页面，完成SQL查询语句的编写。
   - 商品类目偏好查询语句：`select * from data_distill_item_cate_prefer where ds='2020324';`。
   - 商品价格偏好查询语句：`select * from data_distill_item_price_prefer where ds='2020324';`。
9. 单击右上方的**执行**，查询数据。
   - 商品类目偏好查询的数据如下。

	id_type	id_value	prefer	behavior_count	confidence	ds
1	UserID	1002	[{"零食":0.5}]	4	1.0	20200324
2	Email	bo[redacted]@hotmail.com	[{"女装":1.0}]	2	0.2	20200324
3	Email	in[redacted]@163.com	[{"保健品":1.0}]	2	0.2	20200324
4	Email	ne[redacted]@gmail.com	[{"家电":1.0}]	2	0.2	20200324
5	Email	zju[redacted]@qq.com	[{"箱包":1.0}]	2	0.2	20200324
6	Mobile	1358937[redacted]	[{"家具":1.0}]	1	0.2	20200324

- 商品价格偏好查询的数据如下。

	id_type	id_value	prefer	behavior_count	confidence	ds
1	UserID	1002	[{"中价格":1.0}]	4	1.0	20200324
2	Email	bo[redacted]3@hotmail.com	[{"低价格":1.0}]	2	0.2	20200324
3	Email	inf[redacted]@163.com	[{"高价格":1.0}]	2	0.2	20200324
4	Email	nev[redacted]@gmail.com	[{"高价格":1.0}]	2	0.2	20200324
5	Email	zju[redacted]2@qq.com	[{"中价格":1.0}]	2	0.2	20200324
6	Mobile	135893[redacted]	[{"高价格":1.0}]	1	0.2	20200324

3.面向零售店铺的模型构建与管理

3.1. 概述

本文为您介绍本教程的背景信息、准备工作和操作流程。

背景信息

本教程带您体验Dataphin的规范建模和质量管理功能。Dataphin支持规范化、标准化的设计数据模型、拖拽式的定义数据模型，数据模型的代码系统自动生成。您只需要关注模型的设计和定义，无需关注代码的研发，大大提升了数据的研发效率。

本教程基于零售店铺销售场景，为您介绍如何构建销售模型，以获取目录销售额和实例销售额，同时对店铺ID的唯一性进行了管理，助您真正能够依赖数据进行经营决策。

准备工作

- 新建MaxCompute项目（dqe_demo_prod_odps），配置如下参数，其余参数保持默认，详情请参见创建工作空间。

参数	描述
工作空间名称	填写dqe_demo_prod_odps。
选择计算引擎服务	选择MaxCompute。
实例显示名称	填写test。

- 新建MaxCompute项目（dqe_demo_dev_odps），配置如下参数，其余参数保持默认，详情请参见创建工作空间。

参数	描述
工作空间名称	填写dqe_demo_dev_odps。
选择计算引擎服务	选择MaxCompute。
实例显示名称	填写Mytest。

操作流程

主流程	说明	操作指导
数仓规划	规划店铺销售模型的数仓，包括创建业务板块、计算源、数据源、项目及项目中的成员。	步骤一：规划数仓
创建数据表	通常，您的业务数据需采用创建同步任务或管道任务的方式，导入至Dataphin平台，以构建智能数据。本教程为了让您快速熟悉智能数据构建并管理的流程，采用代码任务的方式构建业务数据。	步骤二：创建数据表
规范定义	定义本教程中的维度、业务过程、原子指标、时间周期和派生指标。	步骤三：规范定义
规范建模	基于规范定义，构建数据模型。	步骤四：规范建模
发布任务	发布维度、维度逻辑表、事实逻辑表表、汇总逻辑表、原子指标和派生指标至生产环境。	步骤五：发布任务
补数据	通常，您构建的数据模型会参与生产环境的调度。本教程为了让您快速熟悉智能数据构建与管理的流程，采用补数据的方式，构建数据模型运行生成的数据。	步骤六：补数据

主流程	说明	操作指导
验证数据	验证数据模型运行后生成的数据是否符合您的预期。	步骤七：验证数据
创建质量规则	为数据表创建质量规则。系统会自动校验数据表质量并生成质量报告。	步骤八：创建质量规则
查看质量报告	质量报告是对数据表参与生产环境运维调度的结果进行周期性质量校验的结果。	步骤九：查看质量报告

3.2. 步骤一：规划数仓

本文为您介绍如何规划店铺销售的数仓，包括创建业务板块、计算源、数据源、项目及项目中的成员。

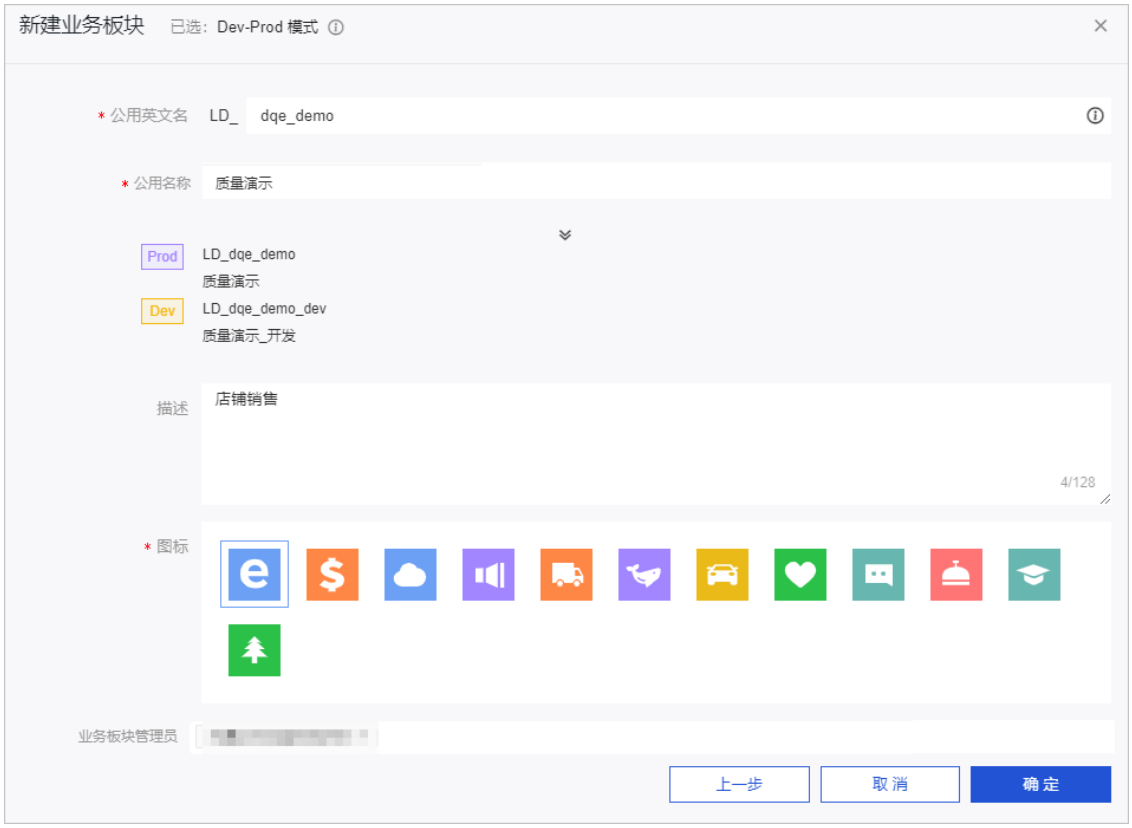
前提条件

完成MaxCompute项目的创建，详情请参见概述。

创建业务板块和数据域

1. 登录Dataphin控制台。
2. 在Dataphin控制台页面，选择工作区地域后，单击**进入Dataphin>>**。
3. 进入**业务板块**页面。
   i. 在Dataphin首页，单击顶部菜单栏的**规划**。
   ii. 在**规划**页面，单击左侧导航栏的**业务板块**。
4. 创建业务板块。
   i. 在**业务板块**页面，单击右上方的**新建业务板块**。
   ii. 在**新建业务板块**对话框，选择**Dev-Prod模式**后，单击**下一步**。

iii. 在**新建业务板块**对话框中，配置参数。



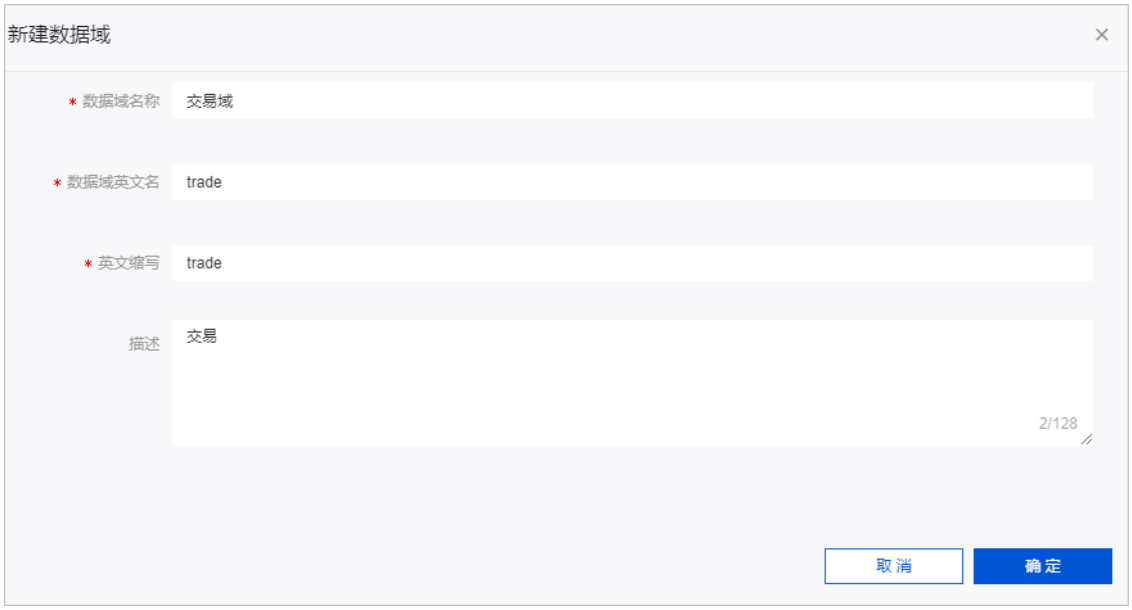
参数	描述
公用英文名	输入为dqe_demo。
公用名称	输入为质量演示。
描述	输入简单的描述。例如，店铺销售。
图标	选择图标。
业务板块管理员	您可以选择**业务板块管理员**，也可以不选择。

iv. 单击**确定**，完成业务板块（LD_dqe_demo和LD_dqe_demo_dev）的创建。

5. 创建交易域。

i. 在dqe_demo业务板块页面，单击页面右侧的**新建数据域**。

ii. 在**新建数据域**对话框中，配置参数。



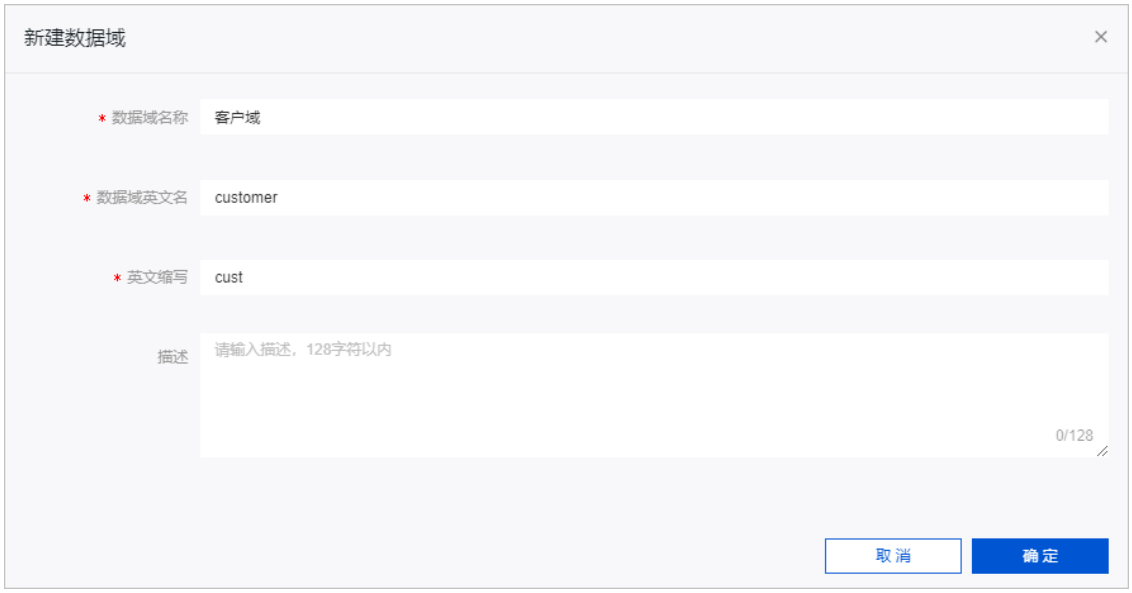
参数	描述
数据域名称	输入交易域。
数据域英文名	输入trade。
英文缩写	输入trade。
描述	输入简单的描述，例如交易。

iii. 单击**确定**。

6. 创建客户域。

i. 在dqe_demo业务板块页面，单击页面右侧的**新建数据域**。

ii. 在**新建数据域**对话框中，配置参数。



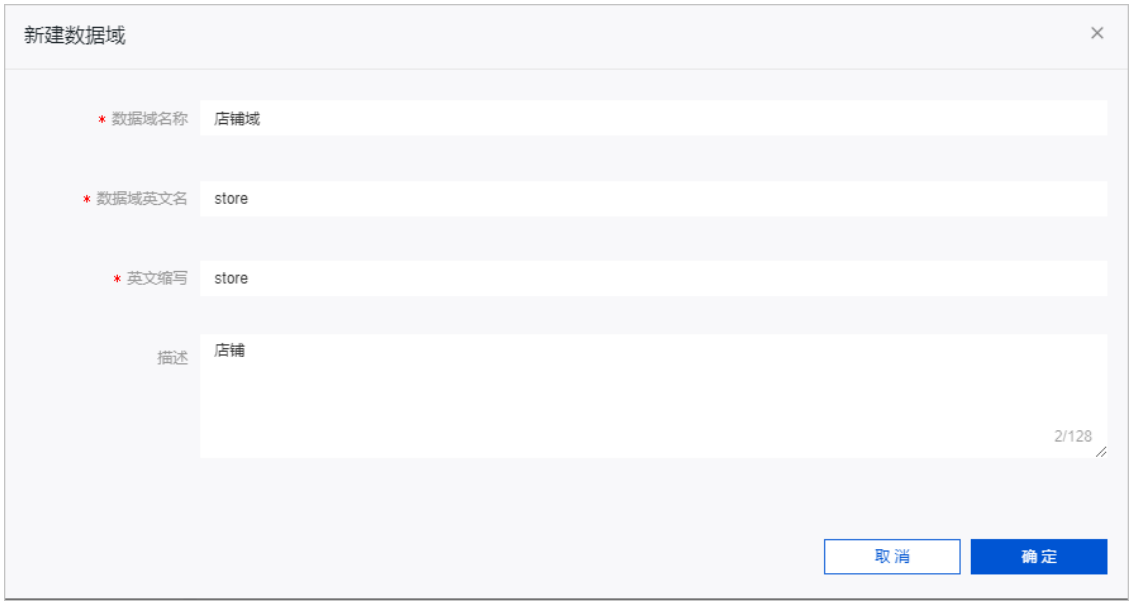
参数	描述
数据域名称	输入客户域。
数据域英文名	输入customer。
英文缩写	输入cust。
描述	输入简单的描述，例如客户。

iii. 单击**确定**。

7. 创建店铺域。

i. 在dqe_demo业务板块页面，单击页面右侧的**新建数据域**。

ii. 在**新建数据域**对话框中，配置参数。



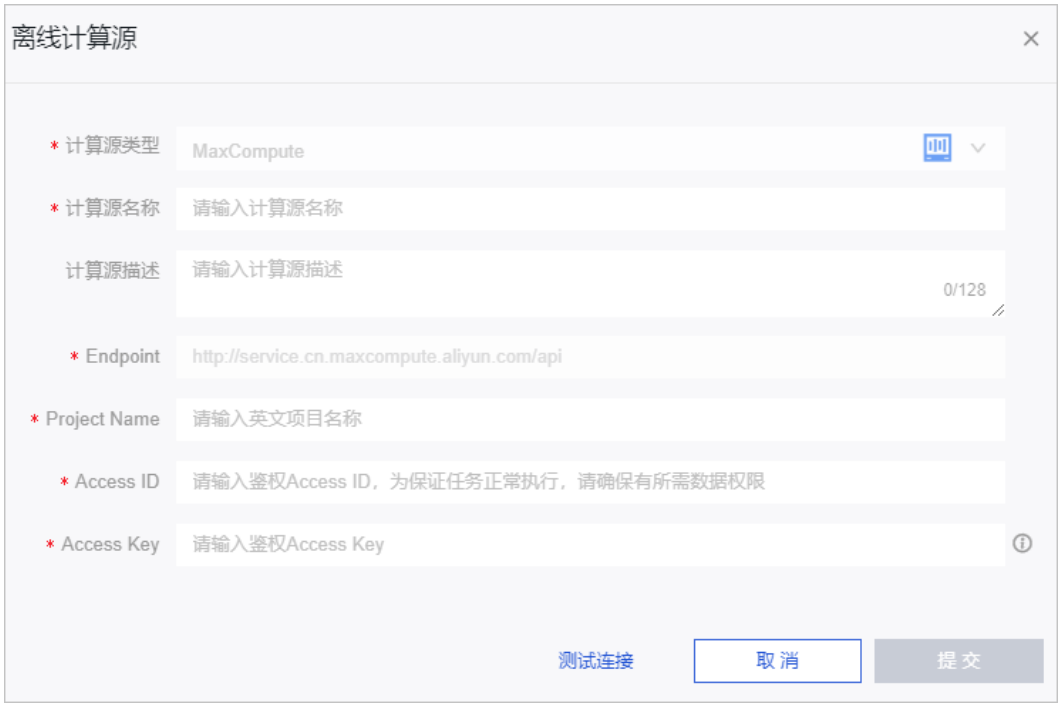
参数	描述
数据域名称	输入店铺域。
数据域英文名	输入store。
英文缩写	输入store。
描述	输入简单的描述，例如店铺。

iii. 单击**确定**。

创建计算源

1. 在**规划**页面，单击左侧导航栏中的**计算源**。
2. 创建Dev项目计算源。
   i. 在**计算源**页面中，鼠标悬停至页面右上方的**新增计算源**，单击**离线计算源**。

ii. 在**离线计算源**对话框中，配置参数。



参数	描述
计算类型	默认为**MaxCompute**，不支持修改。
计算源名称	输入dqe_demo_dev。 **说明** 建议计算源名称与MaxCompute计算源对应Project名称一致。
计算源描述	输入Dev项目的计算源。
Endpoint	默认为 http://service.cn.maxcompute.aliyun.com/api ，不支持修改。
Project Name	输入dqe_demo_dev_odps（MaxCompute项目名称）。
Access ID	访问密钥中的AccessKey ID，您可以通过用户信息管理页面获取。
Access Key	访问密钥中的AccessKey Secret，您可以通过用户信息管理页面获取。

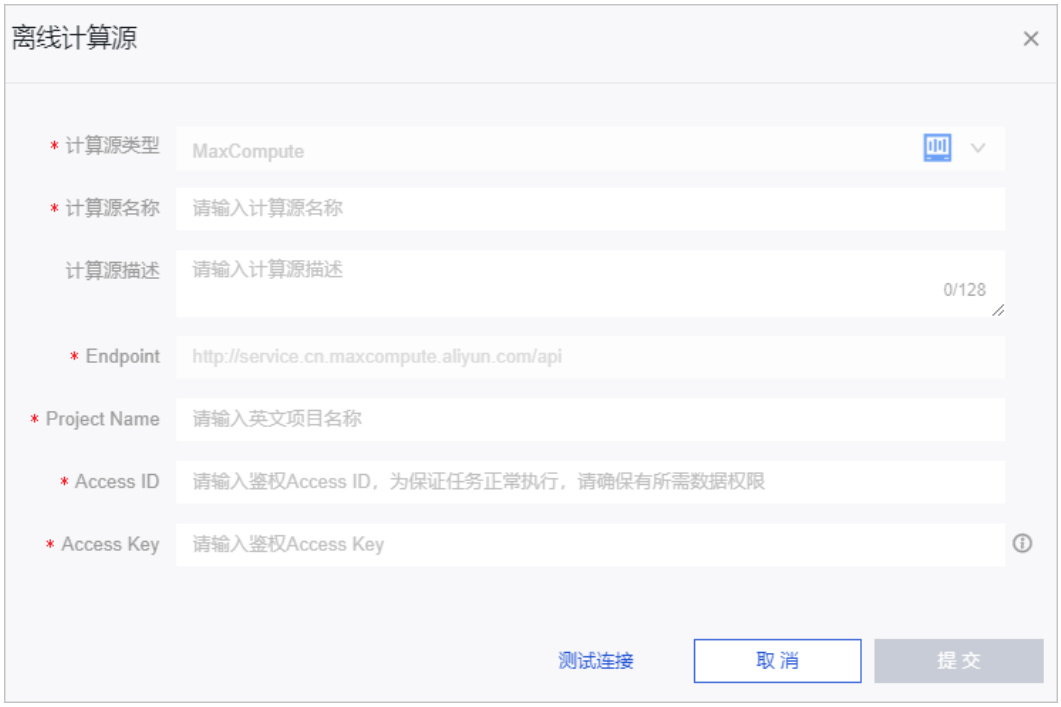
iii. 单击**测试连接**。

iv. 测试连接成功后，单击**提交**，完成**Dev**项目计算源的创建。

3. 创建**Prod**项目计算源。

i. 在**计算源**页面中，鼠标悬停至页面右上方的**新增计算源**，单击**离线计算源**。

ii. 在**离线计算源**对话框中，配置参数。



参数	描述
计算类型	默认为**MaxCompute**，不支持修改。
计算源名称	输入dqe_demo_prod。
计算源描述	输入Prod项目的计算源。
Endpoint	默认为 `http://service.cn.maxcompute.aliyun.com/api` ，不支持修改。
Project Name	输入dqe_demo_prod_odps（MaxCompute项目名称）。
Access ID	访问密钥中的AccessKey ID，您可以通过用户信息管理页面获取。
Access Key	访问密钥中的AccessKey Secret，您可以通过用户信息管理页面获取。

iii. 单击**测试连接**。

iv. 测试连接成功后，单击**提交**，完成**Prod**项目计算源的创建。

创建项目

1. 在**规划**页面，单击左侧导航栏中的**项目管理**。
2. 在**新建项目**对话框中，选择**Dev-Prod模式**后，单击**下一步**。
3. 在**新建项目**对话框中，配置参数。

参数	描述
公用名称	输入质量演示。
公用英文名	输入dqe_demo。
业务板块	选择LD_dqe_demo。
空间类型	选择应用层。
项目1：Dev	选择离线计算源为dqe_demo_dev。
项目2：Prod	选择离线计算源为dqe_demo_prod。
描述	输入简单的描述，例如店铺销售。

4. 单击**确定**，完成项目（dqe_demo和dqe_demo_dev）的创建。

3.3. 步骤二：创建数据表

文为您介绍如何创建本教程中的业务数据表。

背景信息

通常，您的业务数据需采用创建同步任务或管道任务的方式，导入至Dataphin平台，以构建智能数据。本教程为了让您快速熟悉智能数据构建并管理的流程，采用代码任务的方式构建业务数据。

本教程中的数据表包括开发环境的数据表（s_store、s_customer和s_store_sales）和生产环境的数据表（dqe_demo.s_store、dqe_demo.s_customer和dqe_demo.s_store_sales）。

其中，s_store和dqe_demo.s_store、s_customer和dqe_demo.s_customer、s_store_sales和dqe_demo.s_store_sales的结构相同：

- 店铺维度的数据表（s_store和dqe_demo.s_store）。

字段	类型
s_store_sk	bigint
s_store_id	string
s_rec_start_date	string
s_rec_end_date	string

- 客户维度的数据表（s_customer和dqe_demo.s_customer）。

字段	类型
c_customer_sk	bigint
c_customer_id	string
c_current_cdemo_sk	bigint
c_current_hdemo_sk	bigint

- 事实数据表（s_store_sales和dqe_demo.s_store_sales）。

字段	类型
ss_sold_date_sk	bigint
ss_sold_time_sk	bigint
ss_customer_sk	bigint
ss_addr_sk	bigint
ss_store_sk	bigint

步骤一：创建生产环境的数据表

1. 登录Dataphin控制台。
2. 在Dataphin控制台页面，选择工作区地域后，单击**进入Dataphin>>**。
3. 进入**离线计算任务**页面。
   i. 在Dataphin首页，单击顶部菜单栏的**研发**。
   ii. 在数据**开发**页面，单击项目名称后的图标，在**Dev**页签下选择**dqe_demo_dev**项目。

   如果您当前访问的是**dqe_demo_dev**项目，则无需再次选择项目。
   iii. 在数据**开发**页面，单击**数据处理**。

   如果进入数据**开发**页面后，系统默认进入**数据处理**页签，则无需再次单击**数据处理**页签。
   iv. 在**数据处理**页签，单击**即席查询**。
4. 在**即席查询**页签，单击图标。
5. 在**新建文件**对话框，配置参数。

参数	描述
名称	输入表的名称为建表。

参数	描述
描述	输入生产环境数据表。
选择目录	默认为临时代码。

6. 单击**确定**。
7. 在**代码编写**页面，编写建表及向表中写入数据的SQL语句。

```
--创建数据表dqe_demo.s_store。
CREATE TABLE IF NOT EXISTS dqe_demo.s_store
 (
  s_store_sk bigint ,
  s_store_id string ,
  s_rec_start_date string ,
  s_rec_end_date string
)
PARTITIONED BY (
  `ds` STRING
);
insert into table dqe_demo.s_store partition (ds ='${bizdate}') values(10001,'c0001','20200618','20200619');
insert into table dqe_demo.s_store partition (ds ='${bizdate}') values(10002,'c0002','20200519','20200520');
insert into table dqe_demo.s_store partition (ds ='${bizdate}') values(10003,'c0003','20200520','20200521');
insert into table dqe_demo.s_store partition (ds ='${bizdate}') values(10004,'c0004','20200519','20200520');
insert into table dqe_demo.s_store partition (ds ='${bizdate}') values(10005,'c0005','20200517','20200518');
insert into table dqe_demo.s_store partition (ds ='${bizdate}') values(10005,'c0005','20200515','20200520');
insert into table dqe_demo.s_store partition (ds ='${bizdate}') values(10007,'c0007','20200515','20200519');
insert into table dqe_demo.s_store partition (ds ='${bizdate}') values(10008,'c0008','20200514','20200518');
insert into table dqe_demo.s_store partition (ds ='${bizdate}') values(10009,'c0009','20200515','20200517');
insert into table dqe_demo.s_store partition (ds ='${bizdate}') values(100010,'c00010','20200513','20200516');
insert into table dqe_demo.s_store partition (ds ='${bizdate}') values(100011,'c00011','20200519','20200516');
--创建数据表dqe_demo.s_store_sales。
CREATE TABLE IF NOT EXISTS dqe_demo.s_store_sales
 (
  ss_sold_date_sk  bigint ,
  ss_sales_price bigint ,
  ss_customer_sk  bigint ,
  ss_list_price  bigint ,
  ss_store_sk  bigint
)
PARTITIONED BY (
  `ds` STRING
);
insert into table dqe_demo.s_store_sales partition (ds ='${bizdate}') values(11121,65,11121,65,10001);
insert into table dqe_demo.s_store_sales partition (ds ='${bizdate}') values(11121,78,11121,70,10001);
insert into table dqe_demo.s_store_sales partition (ds ='${bizdate}') values(11111,65,11161,62,10001);
insert into table dqe_demo.s_store_sales partition (ds ='${bizdate}') values(11121,78,11161,72,20004);
insert into table dqe_demo.s_store_sales partition (ds ='${bizdate}') values(11161,65,11161,58,20004);
insert into table dqe_demo.s_store_sales partition (ds ='${bizdate}') values(11161,78,11161,75,10003);
insert into table dqe_demo.s_store_sales partition (ds ='${bizdate}') values(11161,65,11121,55,10003);
insert into table dqe_demo.s_store_sales partition (ds ='${bizdate}') values(11181,78,11181,68,10003);
insert into table dqe_demo.s_store_sales partition (ds ='${bizdate}') values(11181,65,11181,60,20004);
insert into table dqe_demo.s_store_sales partition (ds ='${bizdate}') values(11181,78,11181,70,20004);
insert into table dqe_demo.s_store_sales partition (ds ='${bizdate}') values(11181,65,11181,60,20004);
--创建数据表dqe_demo.s_customer。
CREATE TABLE IF NOT EXISTS dqe_demo.s_customer
 (
  c_customer_sk  bigint ,
  c_customer_id string ,
  c_current_cdemo_sk  bigint ,
```

```
 c_current_hdemo_sk bigint
)
PARTITIONED BY (
 `ds` STRING
);
insert into table dqe_demo.s_customer partition (ds ='${bizdate}') values(11121,'c0001',10004,20004);
insert into table dqe_demo.s_customer partition (ds ='${bizdate}') values(11121,'c0002',10004,20004);
insert into table dqe_demo.s_customer partition (ds ='${bizdate}') values(11111,'c0003',10004,20004);
insert into table dqe_demo.s_customer partition (ds ='${bizdate}') values(11121,'c0004',10004,20004);
insert into table dqe_demo.s_customer partition (ds ='${bizdate}') values(11161,'c0005',10004,20004);
insert into table dqe_demo.s_customer partition (ds ='${bizdate}') values(11161,'c0006',10004,20004);
insert into table dqe_demo.s_customer partition (ds ='${bizdate}') values(11161,'c0007',10004,20004);
insert into table dqe_demo.s_customer partition (ds ='${bizdate}') values(11181,'c0008',10004,20004);
insert into table dqe_demo.s_customer partition (ds ='${bizdate}') values(11181,'c0009',10004,20004);
insert into table dqe_demo.s_customer partition (ds ='${bizdate}') values(11181,'c0008',10004,20004);
insert into table dqe_demo.s_customer partition (ds ='${bizdate}') values(11181,'c0008',10004,20004);
```

8. 单击页面右上方的**执行**，执行编写的建表语句。
9. 单击页面右上方的图标，保存即席查询。

步骤二：创建开发环境数据表（s_store）

1. 单击图标后，选择**MAX_COMPUTE_SQL**。
2. 在**新建文件**对话框，配置参数。

参数	描述
名称	输入表的名称为s_store。
调度类型	选择**手动节点**。
描述	输入店铺维度。
选择目录	选择代码管理。

3. 单击**确定**。
4. 在**代码编写**页面，编写建表及向表中写入数据的SQL语句。

```
CREATE TABLE IF NOT EXISTS `s_store`
(
 s_store_sk bigint ,
 s_store_id string ,
 s_rec_start_date string ,
 s_rec_end_date string
)
PARTITIONED BY (
 `ds` STRING
);
insert into table s_store partition (ds ='${bizdate}') values(10001,'c0001','20200618','20200619');
insert into table s_store partition (ds ='${bizdate}') values(10002,'c0002','20200519','20200520');
insert into table s_store partition (ds ='${bizdate}') values(10003,'c0003','20200520','20200521');
insert into table s_store partition (ds ='${bizdate}') values(10004,'c0004','20200519','20200520');
insert into table s_store partition (ds ='${bizdate}') values(10005,'c0005','20200517','20200518');
insert into table s_store partition (ds ='${bizdate}') values(10005,'c0005','20200515','20200520');
insert into table s_store partition (ds ='${bizdate}') values(10007,'c0007','20200515','20200519');
insert into table s_store partition (ds ='${bizdate}') values(10008,'c0008','20200514','20200518');
insert into table s_store partition (ds ='${bizdate}') values(10009,'c0009','20200515','20200517');
insert into table s_store partition (ds ='${bizdate}') values(100010,'c00010','20200513','20200516');
insert into table s_store partition (ds ='${bizdate}') values(100011,'c00011','20200519','20200516');
```

5. 单击页面右上方的**执行**，执行编写的建表语句。
6. 单击页面右上的图标，完成数据表（s_store）的保存。
7. 提交数据表（s_store）。
   i. 单击页面右上方的图标。
   ii. 在**提交备注**对话框，输入备注信息。
   iii. 单击**确定并提交**。

步骤三：创建开发环境数据表（s_customer）

1. 单击图标后，选择**MAX_COMPUTE_SQL**。
2. 在**新建文件**对话框，配置参数。

参数	描述
名称	输入表的名称为s_customer。
调度类型	选择**手动节点**。
描述	输入顾客维度。
选择目录	选择代码管理。

3. 单击**确定**。
4. 在**代码编写**页面，编写建表及向表中写入数据的SQL语句。

```
CREATE TABLE IF NOT EXISTS `s_customer`
(
 c_customer_sk bigint,
 c_customer_id string,
 c_current_cdemo_sk bigint,
 c_current_hdemo_sk bigint
)
PARTITIONED BY (
 `ds` STRING
);
insert into table s_customer partition (ds='${bizdate}') values(11121,'c0001',10004,20004);
insert into table s_customer partition (ds='${bizdate}') values(11121,'c0002',10004,20004);
insert into table s_customer partition (ds='${bizdate}') values(11111,'c0003',10004,20004);
insert into table s_customer partition (ds='${bizdate}') values(11121,'c0004',10004,20004);
insert into table s_customer partition (ds='${bizdate}') values(11161,'c0005',10004,20004);
insert into table s_customer partition (ds='${bizdate}') values(11161,'c0006',10004,20004);
insert into table s_customer partition (ds='${bizdate}') values(11161,'c0007',10004,20004);
insert into table s_customer partition (ds='${bizdate}') values(11181,'c0008',10004,20004);
insert into table s_customer partition (ds='${bizdate}') values(11181,'c0009',10004,20004);
insert into table s_customer partition (ds='${bizdate}') values(11181,'c0008',10004,20004);
insert into table s_customer partition (ds='${bizdate}') values(11181,'c0008',10004,20004);
```

5. 单击页面右上方的**执行**，执行编写的建表语句。
6. 单击页面右上的图标，完成数据表（s_customer）的保存。
7. 提交数据表（s_customer）。
   i. 单击页面右上方的图标。
   ii. 在**提交备注**对话框，输入备注信息。
   iii. 单击**确定并提交**。

步骤四：创建开发环境数据表（s_store_sales）

1. 单击图标后，选择**MAX_COMPUTE_SQL**。
2. 在**新建文件**对话框，配置参数。

参数	描述
名称	输入表的名称为s_store_sales。
调度类型	选择**手动节点**。
描述	输入销售数据。
选择目录	选择代码管理。

3. 单击**确定**。
4. 在**代码编写**页面，编写建表及向表中写入数据的SQL语句。

```
CREATE TABLE IF NOT EXISTS `s_store_sales`
(
 ss_sold_date_sk bigint ,
 ss_sales_price bigint ,
 ss_customer_sk bigint ,
 ss_list_price bigint ,
 ss_store_sk bigint
)
PARTITIONED BY (
 `ds` STRING
);
insert into table s_store_sales partition (ds ='${bizdate}') values(11121,65,11121,65,10001);
insert into table s_store_sales partition (ds ='${bizdate}') values(11121,78,11121,70,10001);
insert into table s_store_sales partition (ds ='${bizdate}') values(11111,65,11161,62,10001);
insert into table s_store_sales partition (ds ='${bizdate}') values(11121,78,11161,72,20004);
insert into table s_store_sales partition (ds ='${bizdate}') values(11161,65,11161,58,20004);
insert into table s_store_sales partition (ds ='${bizdate}') values(11161,78,11161,75,10003);
insert into table s_store_sales partition (ds ='${bizdate}') values(11161,65,11121,55,10003);
insert into table s_store_sales partition (ds ='${bizdate}') values(11181,78,11181,68,10003);
insert into table s_store_sales partition (ds ='${bizdate}') values(11181,65,11181,60,20004);
insert into table s_store_sales partition (ds ='${bizdate}') values(11181,78,11181,70,20004);
insert into table s_store_sales partition (ds ='${bizdate}') values(11181,65,11181,60,20004);
```

5. 单击页面右上方的**执行**，执行编写的建表语句。
6. 单击页面右上的图标，完成表的保存。
7. 提交数据表（s_store_sales）。
   i. 单击页面右上方的图标。
   ii. 在**提交备注**对话框，输入备注信息。
   iii. 单击**确定并提交**。

3.4. 步骤三：规范定义

规范定义是指以维度建模作为理论基础，构建总线矩阵，划分并定义数据域、业务过程、维度、原子指标、时间周期和派生指标。

本教程中规范定义包括如下内容：

- 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归类，用于数据的管理和应用。本教程中数据域包括交易域、客户域和店铺域。

- 维度

  维度是度量的基础，用来反映业务的一类属性，这类属性的集合构成一个维度，也可以称为一个实体对象。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。本教程中以store和customer为维度进行数据建模。

- 业务过程

  业务过程是指企业的业务活动事件，如下单、支付、退款都是业务过程。通常业务过程是企业活动中的事件，因此业务过程是一个不可拆分的行为事件。本教程中业务过程为店铺售卖商品，使用store_sales标识。

- 原子指标

  基于某一业务事件行为下的数据统计计量，是业务定义中不可再拆分的指标，具有明确业务含义的名词。例如，每笔订单的支付金额汇总为支付总额。本教程中以ss_sales_price和ss_list_price为具体待统计的数据并对其进行汇总创建原子指标，则sum（ss_sales_price）和sum（ss_list_price）是原子指标。

- 业务限定

业务限定为统计的业务范围的圈定。为保障所有统计指标统一、标准、规范地构建，业务限定在业务板块内唯一，并唯一归属于一个来源逻辑表，计算逻辑也以该来源逻辑表模型的字段为基础进行定义。本教程中没有设定业务限定。

- 派生指标

  `派生指标=原子指标+业务限定+统计周期+维度（维度的组合）（统计粒度）` 。派生指标即基于原子指标、时间周期和维度，圈定业务统计范围并分析获取业务统计指标的数值。本教程中派生指标统计店铺最近1天的目录销售总额和实际销售总额。顾客customer和store为维度，店铺售卖产品为业务过程，销售总额为原子指标
  即sum（ss_sales_price）和sum（ss_list_price），统计周期为最近1天。

3.5. 步骤四：规范建模

本文为您介绍如何完成本教程中零售店铺销售模型的构建。

前提条件

- 完成规划数仓，详情请参见步骤一：规划数仓。
- 完成数据表的创建，详情请参见步骤二：创建数据表。
- 完成规范定义，详情请参见步骤三：规范定义。

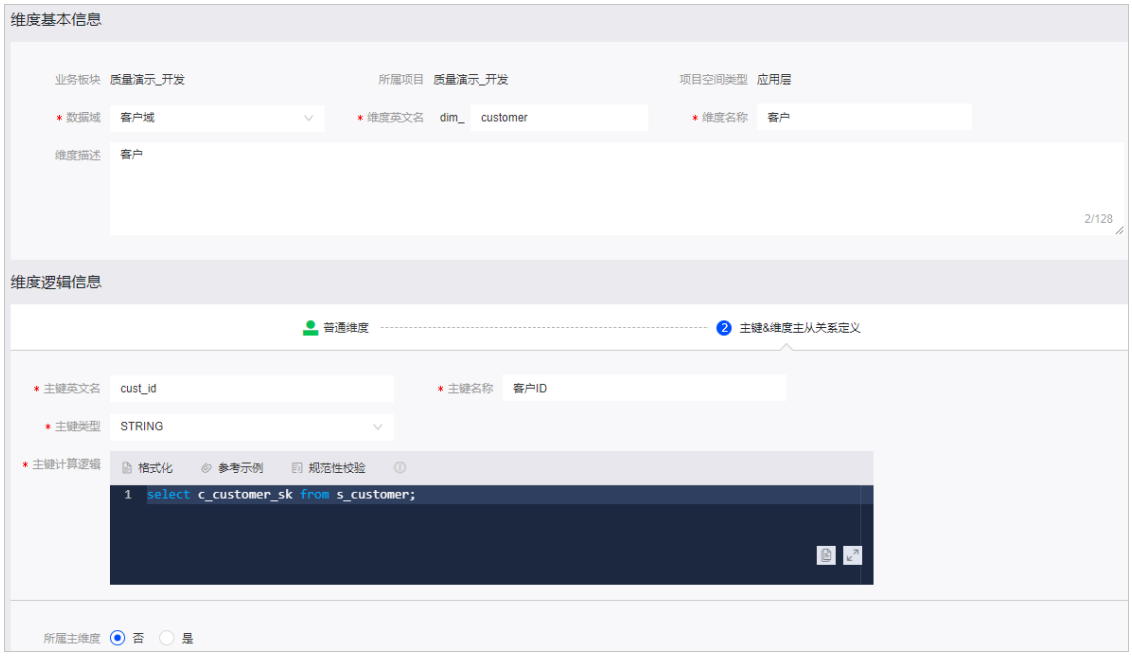
步骤一：创建维度

1. 登录Dataphin控制台。
2. 在Dataphin控制台页面，选择工作区地域后，单击**进入Dataphin>>**。
3. 进入**维度**页面。
   i. 在Dataphin首页，单击顶部菜单栏的**研发**。
   ii. 单击项目名称后的图标，单击**Dev**页签，选择**dqe_demo_dev**项目。

   如果您当前访问的是**dqe_demo_dev**项目，则无需再次选择项目。
   iii. 在数据**开发**页面，单击**规范建模**。

   如果进入数据**开发**页面后，系统默认进入**规范建模**页签，则无需再次单击**规范建模**页签。
   iv. 在**规范建模**页面，单击左侧导航栏的**维度**图标。
4. 创建customer维度。
   i. 在**维度**页面，单击图标。

ii. 在**新建维度**页面，配置参数。



区域	参数	描述
维度基本信息	**数据域**	选择客户域。
	维度英文名	输入customer。
	维度名称	输入客户。
	维度描述	输入维度的简单描述。
维度逻辑信息	**设置维度类型**	选择**普通维度**。
	主键英文名	输入cust_id。
	主键名称	输入客户ID。
	主键类型	选择STRING。
	主键计算逻辑	定义维度的**主键计算逻辑**： a. 在代码输入框中，输入内容如下。 `select c_customer_sk from s_customer;` b. 单击**主键计算逻辑**后面的**规范性校验**，校验您编写的代码是否符合语法规范。

iii. 单击页面上方图标，保存维度。

iv. 单击页面上方图标，提交维度。

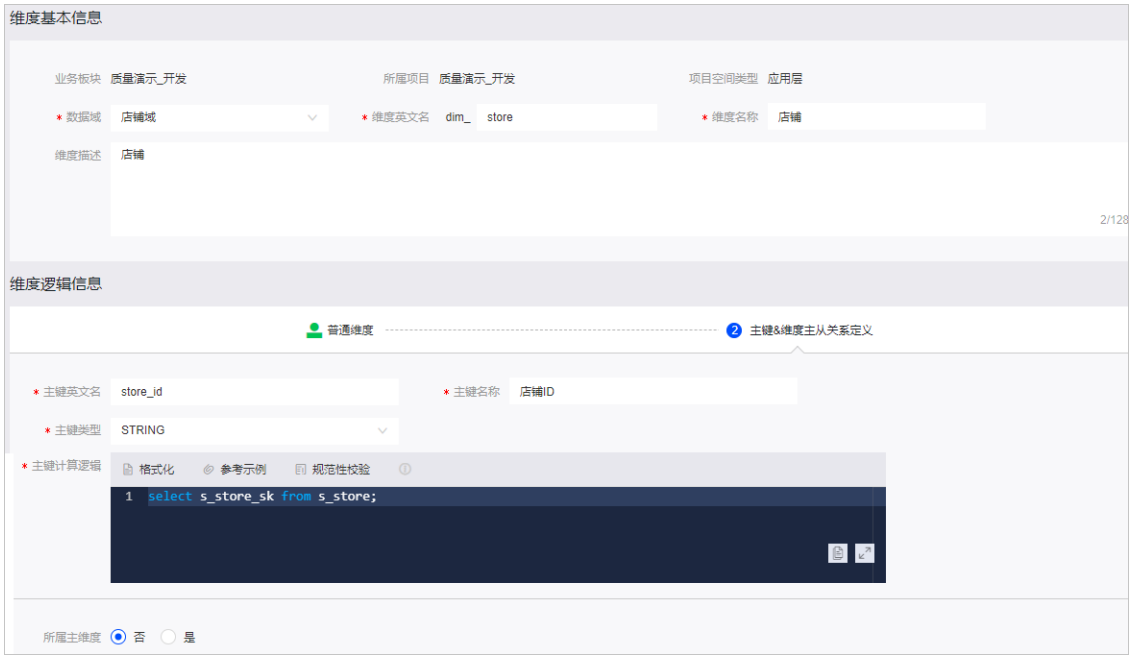
v. 在**提交备注**对话框，填写备注信息。

vi. 单击**确定并提交**。

5. 创建store维度。

i. 在**维度**页面，单击图标。

ii. 在**新建维度**页面，配置参数。



区域	参数	描述
维度基本信息	**数据域**	选择店铺域。
	维度英文名	输入store。
	维度名称	输入店铺。
	维度描述	输入维度的简单描述。
维度逻辑信息	**设置维度类型**	选择**普通维度**。
	主键英文名	输入store_id。
	主键名称	输入店铺ID。
	主键类型	选择STRING。
	主键计算逻辑	定义维度的**主键计算逻辑**： a. 在代码输入框中，输入内容如下。 `select s_store_sk from s_store;` b. 单击**主键计算逻辑**后面的**规范性校验**，校验您编写的代码是否符合语法规范。

iii. 单击页面上方图标，保存维度。

iv. 单击页面上方图标，提交维度。

v. 在**提交备注**对话框，填写备注信息。

vi. 单击**确定并提交**。

步骤二：创建业务过程和事实逻辑表

1. 进入**新建业务过程**对话框。

i. 在**规范建模**页面，单击左侧导航栏中的**业务过程**图标。

ii. 在**业务过程**页面，单击图标。

2. 在**新建业务过程**对话框，配置参数。

参数	描述
数据域	选择交易域。
英文名	输入store_sales。
名称	输入店铺销售。
描述	输入简单描述。

3. 单击**提交**。
4. 在**提交备注**对话框，输入备注信息。
5. 单击**确定并提交**，完成业务过程的创建。
6. 创建事实逻辑表。

i. 在**业务过程**页面，单击**store_sales**。

ii. 在**业务过程属性**框，鼠标悬停至图标后，选择**创建逻辑表**。

iii. 在**新建事实逻辑表**对话框，配置参数。

参数	描述
数据域	默认选择交易域（trade）。
业务过程	默认选择店铺销售（store_sales）。
明细类型	默认选择**事务型**。
英文名	无需配置。
名称	输入店铺销售记录。
描述	输入简单描述。
来源主表	选择dqe_demo_dev.s_store_sales。

iv. 单击**下一步**。

v. **是否设定主键**选择为**否**。

vi. 单击**提交**，完成事实逻辑表的创建。

7. 配置事实逻辑表。

i. 在**fct_store_sales_rc_di**逻辑事实表页面，单击**添加度量**。

ii. 在**新建度量**对话框，配置参数。

参数	描述
来源表	选择**引入字段**。
新建字段	新建字段： a. 单击左侧列表中ss_list_price和ss_sales_price字段后的图标。 b. 在**新建字段**区域，ss_list_price的**字段名称**输入目录价格，ss_sales_price的**字段名称**输入销售价格。

iii. 单击**保存并校验**。

iv. 在fct_store_sales_rc_di逻辑事实表页面，单击**添加事实属性**。

v. 在**新建事实属性**对话框，配置参数。

参数	描述
来源表	选择**引入字段**。
新建字段	新建字段： a. 单击左侧列表中ss_store_sk和ss_customer_sk字段后的图标。 b. 在**新建字段**区域，ss_store_sk的**字段名称**输入店铺ID，ss_customer_sk的**字段名称**输入客户ID。

vi. 单击**保存并校验**。

vii. 在fct_store_sales_rc_di逻辑事实表页面，单击**添加关联维度**。

viii. 在**新建关联维度**对话框，配置参数。

关联客户域和店铺域维度：

- 关联客户域维度：
  a. **关联维度信息**选择客户域数据域下的dim_customer客户维度逻辑表。**编辑关联逻辑**选择事实属性下的ss_customer_sk。其他参数均保持默认值。
  b. 单击**确定**。
- 关联店铺域维度：
  a. 在主表页面，单击**主表编辑**。
  b. 鼠标悬停至**新建字段**，单击**关联维度**。
  c. 在**新建关联维度**对话框，**关联维度信息**选择店铺域数据域下的dim_store店铺维度逻辑表。**编辑关联逻辑**选择事实属性下的ss_store_sk。其他参数均保持默认值。
  d. 单击**确定**。

8. 保存和提交事实逻辑表。

i. 单击图标，保存事实逻辑表。

ii. 单击图标，提交事实逻辑表。

iii. 在**提交备注**对话框，输入备注信息。

iv. 单击**确定并提交**，完成事实逻辑表的创建。

步骤三：创建原子指标

1. 在**规范建模**页面，单击左侧导航栏中的**原子指标**图标。

2. 在**原子指标**页面，单击图标后，选择**新建原子指标**。
3. 在**新建原子指标**页面，**数据源**选择为**交易域**，**来源表**选择为fct_store_sales_rc_di后，单击**新建原子指标**。
4. 创建sum_list_price原子指标。
   i. 在**新建原子指标**对话框，配置参数。

参数	描述
主要来源字段	选择fct_store_sales_rc_di.ss_list_price。
英文名	输入sum_list_price。
名称	输入目录销售额。
描述	输入简单描述。
数据类型	选择BIGINT。
计算逻辑	输入sum(fct_store_sales_rc_di.ss_list_price)。

   ii. 单击**规范性校验**，校验**计算逻辑**的语法。
   iii. 单击**提交**。
   iv. 在**提交备注**对话框，输入备注信息。
   v. 单击**确定并提交**，完成原子指标的创建。
5. 创建sum_sales_price原子指标。
   i. 在**新建原子指标**页面，单击**新建原子指标**。
   ii. 在**新建原子指标**对话框，配置参数。

参数	描述
主要来源字段	选择fct_store_sales_rc_di.ss_sales_price。
英文名	输入sum_sales_price。
名称	输入实际销售额。
描述	输入简单描述。
数据类型	选择BIGINT。
计算逻辑	输入sum(fct_store_sales_rc_di.ss_sales_price)。

   iii. 单击**规范性校验**，校验**计算逻辑**的SQL语句。
   iv. 单击**提交**。
   v. 在**提交备注**对话框，输入备注信息。
   vi. 单击**确定并提交**，完成原子指标的创建。

步骤四：创建派生指标

1. 在**规范建模**页面，单击左侧导航栏中的**派生指标**图标。
2. 在**新建派生指标**页面，单击图标。
3. 创建店铺和顾客维度sum_sales_price_1d的派生指标。
   i. 在**新建派生指标**页面，选择**原子指标**为交易域下的sum_sales_price。

ii. 单击**下一步**。

iii. 在**创建派生指标**页面，配置参数。

参数	描述
统计粒度	添加统计粒度： a. 选择dim_store 店铺下的fct_store_sales_rc_di.dim_store。 b. 单击**新建统计粒度**。 c. 选择dim_customer 客户下的fct_store_sales_rc_di.dim_customer。
统计周期	选择**最近1天**。
业务限定	本教程中无需配置。

iv. 单击**预生成派生指标**。

v. 在**编辑派生指标**区域，确认输入信息后，单击**提交**。

vi. 单击**提交**，提交派生指标。

vii. 在**提交备注**对话框，输入备注信息。

viii. 单击**确定并提交**，完成派生指标的创建。

4. 创建店铺和顾客维度的sum_list_price_1d的派生指标。

i. 在**新建派生指标**页面，选择**原子指标**为交易域下的sum_list_price。

ii. 单击**下一步**。

iii. 在**创建派生指标**页面，配置参数。

参数	描述
统计粒度	添加统计粒度： a. 选择dim_store 店铺下的fct_store_sales_rc_di.dim_store。 b. 单击**新建统计粒度**。 c. 选择dim_customer 客户下的fct_store_sales_rc_di.dim_customer。
统计周期	选择**最近1天**。
业务限定	本教程中无需配置。

iv. 单击**预生成派生指标**。

v. 在**编辑派生指标**区域，确认输入信息后，单击**提交**。

vi. 单击**提交**，提交派生指标。

vii. 在**提交备注**对话框，输入备注信息。

viii. 单击**确定并提交**，完成派生指标的创建。

步骤五：查看汇总逻辑表

1. 在**规范建模**页面，单击左侧导航栏中的**汇总逻辑表**图标。
2. 在**汇总逻辑表**页面，单击**dws_customer**，查看汇总逻辑表**dws_customer**下的派生指标。

3. 在**汇总逻辑表**页面，单击**dws_store**，查看汇总逻辑表**dws_store**下的派生指标。

3.6. 步骤五：发布任务

本文为您介绍如何发布已创建的任务至生产环境。

操作步骤

1. 登录Dataphin控制台。
2. 在Dataphin控制台页面，选择工作区地域后，单击**进入Dataphin>>**。
3. 进入**发布**页面。
   i. 在Dataphin首页，单击顶部菜单栏的**研发**。
   ii. 单击项目名称后的图标，单击**Dev**页签，选择**dqe_demo_dev**项目。

   如果您当前访问的是**dqe_demo_dev**项目，则无需选择项目。
   iii. 在数据**开发**页面，单击顶部菜单栏的**发布**。
4. 在**发布**页面，单击**规范建模**页签。
5. 发布维度和维度逻辑表。
   i. 在**规范建模**页签，选中**dim_customer**、**dim_customer**、**dim_store**和**dim_store**。

   ii. 单击页面下方的**发布**。
   iii. 在**发布**对话框，单击**确定**。
6. 发布派生指标、原子指标和事实逻辑表。
   i. 在**规范建模**页签，选中**sum_list_price_1d**、**sum_sales_price_1d**、**sum_list_price**、**sum_sales_price**和**fct_store_sales_di**。

   ii. 单击页面下方的**发布**。
   iii. 在**发布**对话框，单击**确定**。
7. 发布汇总逻辑表。
   i. 在**规范建模**页签，选中**dws_store**和**dws_customer**。
   ii. 单击页面下方的**发布**。
   iii. 在**发布**对话框，单击**确定**。
8. 查看发布结果。
   i. 单击左侧导航栏的**发布记录列表**。

ii. 单击**规范建模**页签。

iii. 在**规范建模**页签，查看发布状态。

3.7. 步骤六：补数据

本文为您介绍如何为维度逻辑表、事实逻辑表和汇总逻辑表补数据。

前提条件

完成任务的发布，详情请参见步骤五：发布任务。

背景信息

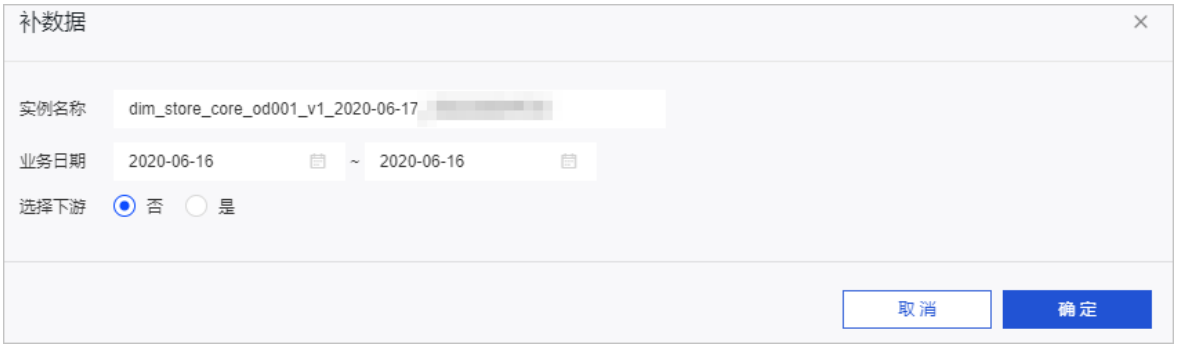
您需要为数据表dim_customer、dim_store、fct_store_sales_rec_di、dws_store和dws_customer进行补数据。数据表的补数据操作相同，下文以dim_store为例介绍。

操作步骤

1. 登录Dataphin控制台。
2. 在Dataphin控制台页面，选择工作区地域后，单击**进入Dataphin>>**。
3. 进入生产环境的**逻辑表运维**页面。
   i. 在Dataphin首页，单击顶部菜单栏的**研发**。
   ii. （可选）在数据**开发**页面，单击项目名称后的图标，在**Prod**页签下选择**dqe_demo**项目。

   如果您当前访问的是**dqe_demo**项目，则无需选择项目。
   iii. 在数据**开发**页面，单击顶部菜单**运维**。
   iv. 在**运维**页面，单击**逻辑表运维**页签。系统默认进入**逻辑表任务**页面。
4. 在**逻辑表任务**页面，打开维度逻辑表任务（**dim_store**）文件夹后，单击维度逻辑表（**dim_store**）节点。
5. 在右侧的DAG图中，右键单击维度逻辑任务节点，选择**补数据**。

6. 在**补数据**对话框，本教程中的**业务日期**为2020-06-16~2020-06-16**选择下游**选择**否**。



7. 单击**确定**。
8. 查看补数据实例的运行日志。
   i. 单击**全局运维**页签。

ii. 在**全局运维**页签，单击左侧导航栏的**离线补数据实例**图标。

iii. 在**离线补数据实例**页面，打开维度逻辑表补数据实例文件夹**dim_store**后，再单击维度逻辑表补数据实例节点。

iv. 在右侧的DAG图中，右键单击维度逻辑表**dim_store**节点，选择**查看运行日志**。

v. 在**运行日志**页面，查看运行日志。

3.8. 步骤七：验证数据

本文为您介绍如何通过即席查询来验证数据。

前提条件

完成维度逻辑表、汇总逻辑表和事实逻辑表补数据操作，详情请参见步骤六：补数据。

背景信息

即席查询模块为您提供主题式数据查询功能。

操作步骤

1. 登录Dataphin控制台。
2. 在Dataphin控制台页面，选择工作区地域后，单击**进入Dataphin>>**。
3. 进入**即席查询**页面。
   i. 在Dataphin首页，单击顶部菜单栏的**研发**。
   ii. 在数据**开发**页面，单击项目名称后的图标，在**Dev**页签下选择**dqe_demo_dev**项目。

   如果您当前访问的是**dqe_demo_dev**项目，则无需选择项目。
   iii. 在数据**开发**页面，单击**即席查询**页签。
4. 在**即席查询**页面，单击图标。
5. 在**新建文件**对话框，配置参数。

参数	描述
名称	输入adhoc。
描述	输入简单描述，例如验证数据。
选择目录	选择临时代码。

6. 单击**确定**。
7. 在**代码编写**页面，编写如下代码。

```
select * from LD_dqe_demo.dws_store where ds > 0;
select * from LD_dqe_demo.dws_customer where ds > 0;
```

8. 保存并执行即席查询。
   i. 单击页面右上角的图标，保存即席查询。
   ii. 单击页面右上角的执行图标，执行即席查询。
9. SQL查询语句执行成功后，在**Result**页签查看返回结果：
   - 统计粒度为customer（客户）的目录销售和实际销售的汇总数据。

cust_id	sum_sales_price_1d	sum_list_price_1d	ds
11121	208	190	20200616
11161	286	267	20200616
11181	286	258	20200616

   - 统计粒度为store（店铺）的目录销售和实际销售的汇总数据。

store_id	sum_sales_price_1d	sum_list_price_1d	ds
10001	208	197	20200616
10003	221	198	20200616
20004	351	320	20200616

3.9. 步骤八：创建质量规则

本文为您介绍如何创建数据表的质量校验规则。

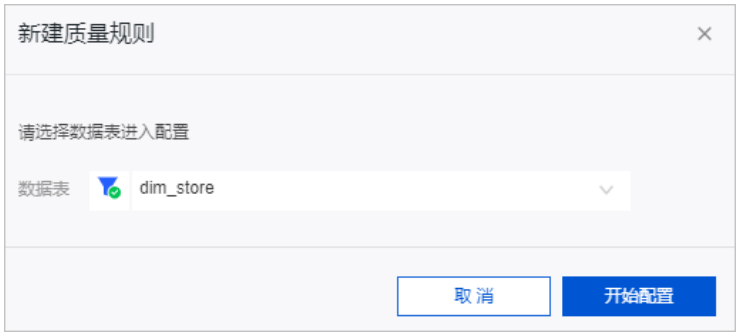
背景信息

本教程中的数据表包括dim_customer、dim_store、fct_store_sales_di、dws_store和dws_customer，下文以维度逻辑表（dws_store）为例。

步骤一 创建质量规则

1. 登录Dataphin控制台。
2. 在Dataphin控制台页面，选择工作区地域后，单击**进入Dataphin>>**。

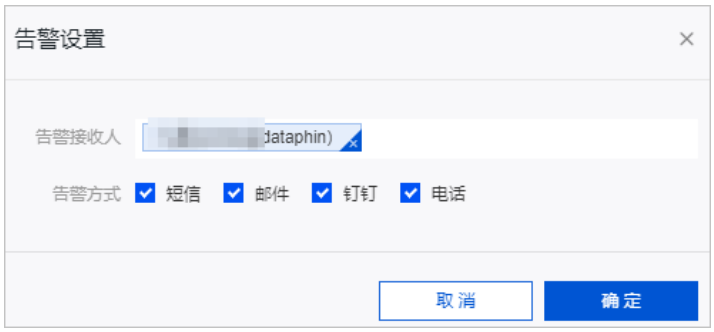
3. 在Dataphin首页，单击顶部菜单栏中的**资产**。
4. 在数据**资产**页面，单击顶部菜单栏的**质量**。
5. 在数据**质量**页面，单击左侧导航栏的**质量规则**。
6. 在**质量规则**页面，单击**新建质量规则**。
7. 在**新建质量规则**对话框中，选择数据表。



   - 单击图标后，**业务板块**选择为LD_dqe_demo、**所属项目**选择为LD_dqe_demo、**表类型**选择为维度逻辑表，在下拉列表中选择dim_store。
   - 在搜索框中输入dim_store，快速筛选数据表。
8. 单击**开始配置**。

步骤二 设置告警

1. 在**DIM_STORE质量规则配置**页面，单击**告警设置**后的图标。
2. 在**告警设置**对话框中，选择**告警接收人**及**告警方式**。



3. 单击**确定**。

步骤三 创建分区表达式

1. 在**DIM_STORE质量规则配置**页面，单击**质量规则配置**后的**新建分区表达式**。
2. 在**新建分区表达式**对话框中，**分区表达式类型**选择为**最近一天**。

3. 单击**保存**。

步骤四 配置质量规则

1. 在**DIM_STORE质量规则配置**页面，单击**分区表达式**后的**新建质量规则**。
2. 在**新建规则**对话框中，配置参数。

参数	描述
对象名称	选择为字段：store_id。
规则强度	选择为弱规则。
模板类型	选择为字段唯一值个数期望校验。
趋势	选择为**固定值**。
对比规则	选择**小于目标值**。
目标值	输入11112。

3. 单击**保存**。
4. 试跑质量规则。
   i. 选中新建的质量规则。

ii. 单击**规则试跑**。

试跑完成后，鼠标悬停至图标，单击**试跑日志**。在**运行日志**页面，查看运行日志。

5. 质量规则试跑成功后，打开**校验开关**，质量校验规则即可生效。

3.10. 步骤九：查看质量报告

本文为您介绍如何查看维度逻辑表的质量校验报告。

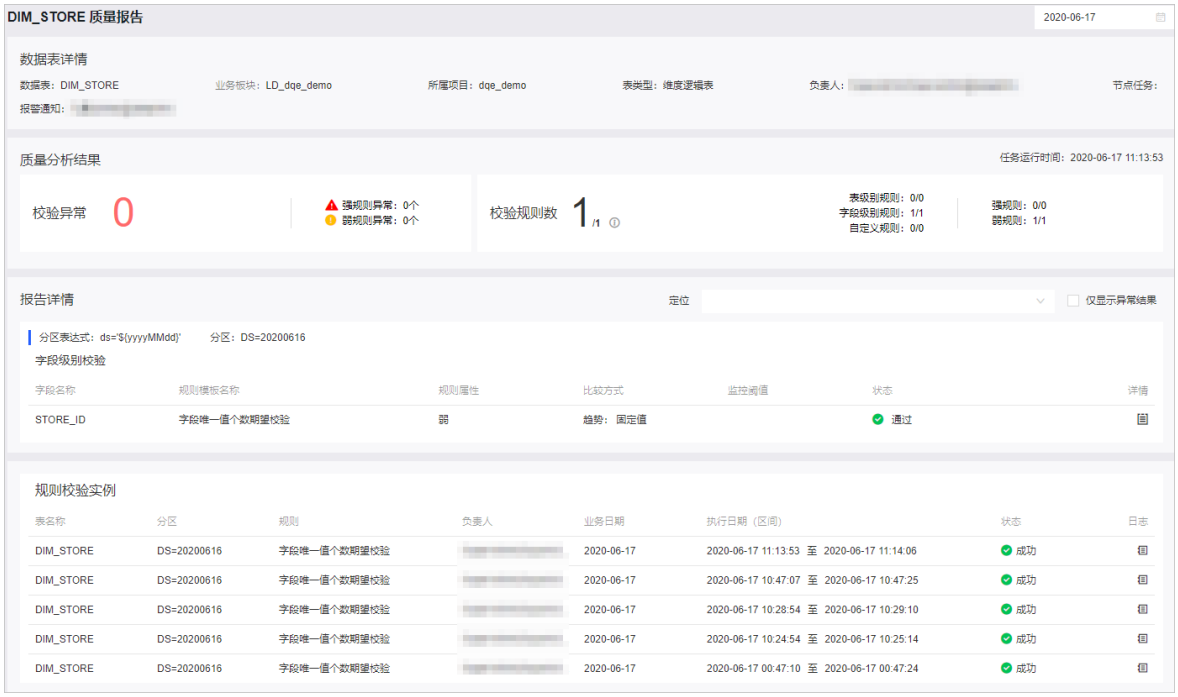
前提条件

完成质量规则的创建，详情请参见步骤八：创建质量规则。

操作步骤

1. 登录Dataphin控制台。
2. 在Dataphin控制台页面，选择工作区地域后，单击**进入Dataphin>>**。
3. 在Dataphin首页，单击顶部菜单栏中的**资产**。
4. 在数据**资产**页面，单击顶部菜单栏的**质量**。
5. 在数据**质量**页面，单击左侧导航栏的**校验记录**。
6. 在**校验记录**页面，查询维度逻辑表（DIM_STORE）。

7. 单击维度逻辑表（DIM_STORE）校验记录下的**质量报告**。
8. 在**DIM_STORE质量报告**页面，查看**数据表详情**、**质量分析结果**、**报告详情**和**规则校验实例**信息。


DIM_STORE 质量报告
2020-06-17
数据表详情
数据表：DIM_STORE
业务板块：LD_dqe_demo
所属项目：dqe_demo
表类型：维度逻辑表
负责人：
节点任务：
报警通知：
质量分析结果
任务运行时间：2020-06-17 11:13:53
校验异常 0
强规则异常：0个
弱规则异常：0个
校验规则数 1 /1
表级别规则：0/0
字段级别规则：1/1
自定义规则：0/0
强规则：0/0
弱规则：1/1
报告详情
定位
仅显示异常结果
分区表达式：ds='${yyyyMMdd}'
分区：DS=20200616
字段级别校验
字段名称
规则模板名称
规则属性
比较方式
监控阈值
状态
详情
STORE_ID
字段唯一值个数期望校验
弱
趋势：固定值
通过
规则校验实例
表名称
分区
规则
负责人
业务日期
执行日期（区间）
状态
日志
DIM_STORE DS=20200616 字段唯一值个数期望校验 2020-06-17 2020-06-17 11:13:53 至 2020-06-17 11:14:06 成功
DIM_STORE DS=20200616 字段唯一值个数期望校验 2020-06-17 2020-06-17 10:47:07 至 2020-06-17 10:47:25 成功
DIM_STORE DS=20200616 字段唯一值个数期望校验 2020-06-17 2020-06-17 10:28:54 至 2020-06-17 10:29:10 成功
DIM_STORE DS=20200616 字段唯一值个数期望校验 2020-06-17 2020-06-17 10:24:54 至 2020-06-17 10:25:14 成功
DIM_STORE DS=20200616 字段唯一值个数期望校验 2020-06-17 2020-06-17 00:47:10 至 2020-06-17 00:47:24 成功