WeData Data Development Platform
Quick Start

Copyright Notice

Trademark Notice

Tencent Cloud

Service Notice

This document provides an overview of the as–is details of Tencent Cloud's products and services in their entirety or part. The descriptions of certain products and services may be subject to adjustments from time to time.

The commercial contract concluded by you and Tencent Cloud will provide the specific types of Tencent Cloud products and services you purchase and the service standards. Unless otherwise agreed upon by both parties, Tencent Cloud does not make any explicit or implied commitments or warranties regarding the content of this document.

Contact Us

We are committed to providing personalized pre–sales consultation and technical after–sale support. Don't hesitate to contact us at 4009100100 or 95716 for any inquiries or concerns.

Contents

Quick Start

Overall Introduction

Last updated：2025-04-18 15:44:55

> **Note**
>
> The product features involved in this case are only the commonly used features within the module. If you need a more detailed understanding of all the features of the module, please view WeData Operation Guide.
>
> Besides the document, you can also learn the content of "Quick Start" through Tencent Cloud WeData Big Data Development and Governance Training Camp.

Background

This document is the basic usage documentation of Tencent Cloud WeData. The target is to help you quickly understand WeData and have a basic concept of the entire process of business data processing.

This document takes the order data synchronization analysis scenario as an example, concatenates the features of each module such as data table structure design, data integration, data development, data quality, and data service, and helps you complete your initial experience of the WeData end-to-end process.

Learning through This Document, You Can Understand the Following Content:

- Understand the entire process of business data development.
- Understand the role of each product module and upstream and downstream collaboration.
- Learn about basic concepts of data table structure design.
- Master the offline data synchronization process.
- Master the offline data development process.
- Master the data quality inspection process.
- Master the data service development process.

Roles and Division of Roles Involved in This Documentation

- **Enterprise administrator:**
  - Responsible for registering and authenticating Tencent Cloud accounts.
  - Responsible for building the network environment.
  - Responsible for purchasing various cloud resources, including: EMR, WeData, MySQL, data service resources.
  - Responsible for creating sub-accounts, projects, and data tables.
  - Responsible for adding sub-accounts and binding data sources in WeData.
- **data developer**

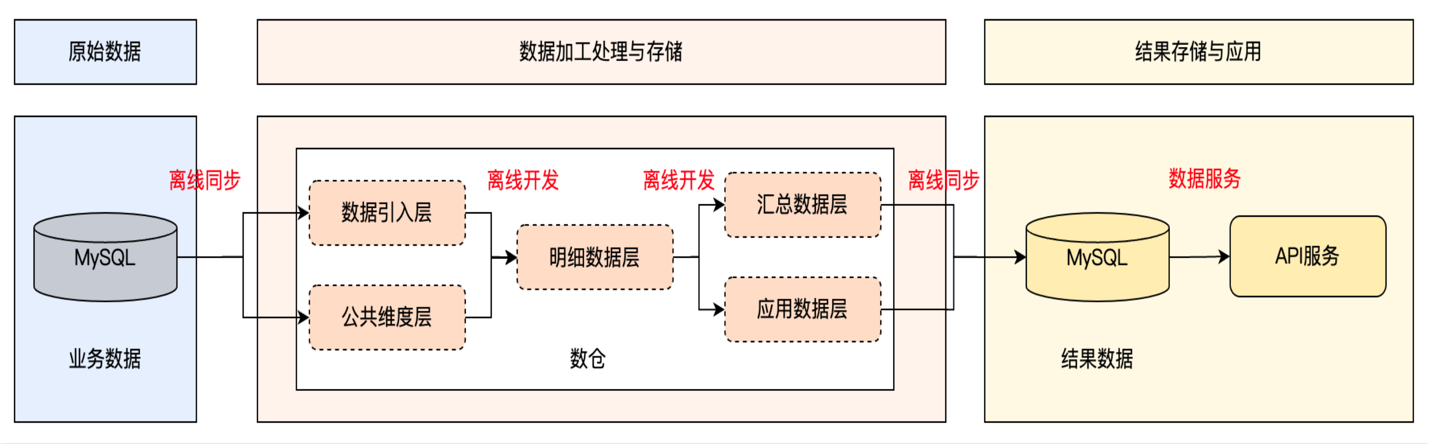
- ○ Responsible for data table structure design.
- ○ Responsible for the Data Integration module.
- ○ Responsible for the data development module.
- ○ Responsible for the data quality module.
- ○ Responsible for the data service module.

Scenario Introduction

In a certain e-commerce platform, the business side **expects to learn about the sales performance of different categories in different cities through analysis of order data** so that it can adjust the operational promotion strategy targeting different cities.

Holistic Picture of the Process

Throughout the entire product chain of WeData products, stages such as raw data research, data processing and storage, and result storage and application will be involved.



Preparations

Last updated: 2025-04-18 15:45:59

Before starting WeData data development, we need to make the following preparations first:

> **Note**
> The following steps involve resource purchasing and payment, which need to be performed by an enterprise administrator.

Flowchart

The specific operations include:

Steps	Description
Sign up for a Tencent Cloud account	• Sign up for a Tencent Cloud account. • Real-name authentication (enterprise authentication recommended). • Create a Tencent Cloud sub-account.
Prepare the network environment	• Create a new VPC. • Bind the subnet. • Apply for a public IP. • Purchase a Public NAT Gateway, bind it to the VPC, and bind the public IP.
Prepare the engine resource environment (Taking Tencent Cloud EMR as an example)	• Purchase an EMR cluster. • Purchase WeData, including integration resources and scheduling resources. • Create a project and bind EMR, bind integration resources, and scheduling resources. • In the WeData project, add a user.
Prepare the business data resource environment (Taking TencentDB for MySQL as an example)	• Purchase MySQL and initialize the business data. • In the WeData project, add a data source.

Signing up for Tencent Cloud account

All the cloud resources involved in this tutorial are purchased through a Tencent Cloud account. Please use the same main Tencent Cloud account. If you already have a Tencent Cloud account and have completed enterprise authentication, please skip this step.

Role: Enterprise Administrator.

Signing up for Tencent Cloud account

Go to Tencent Cloud registration page. You can register by scanning the QR code with WeChat or using an email. For the detailed registration process, please refer to Tencent Cloud Registration Guide.

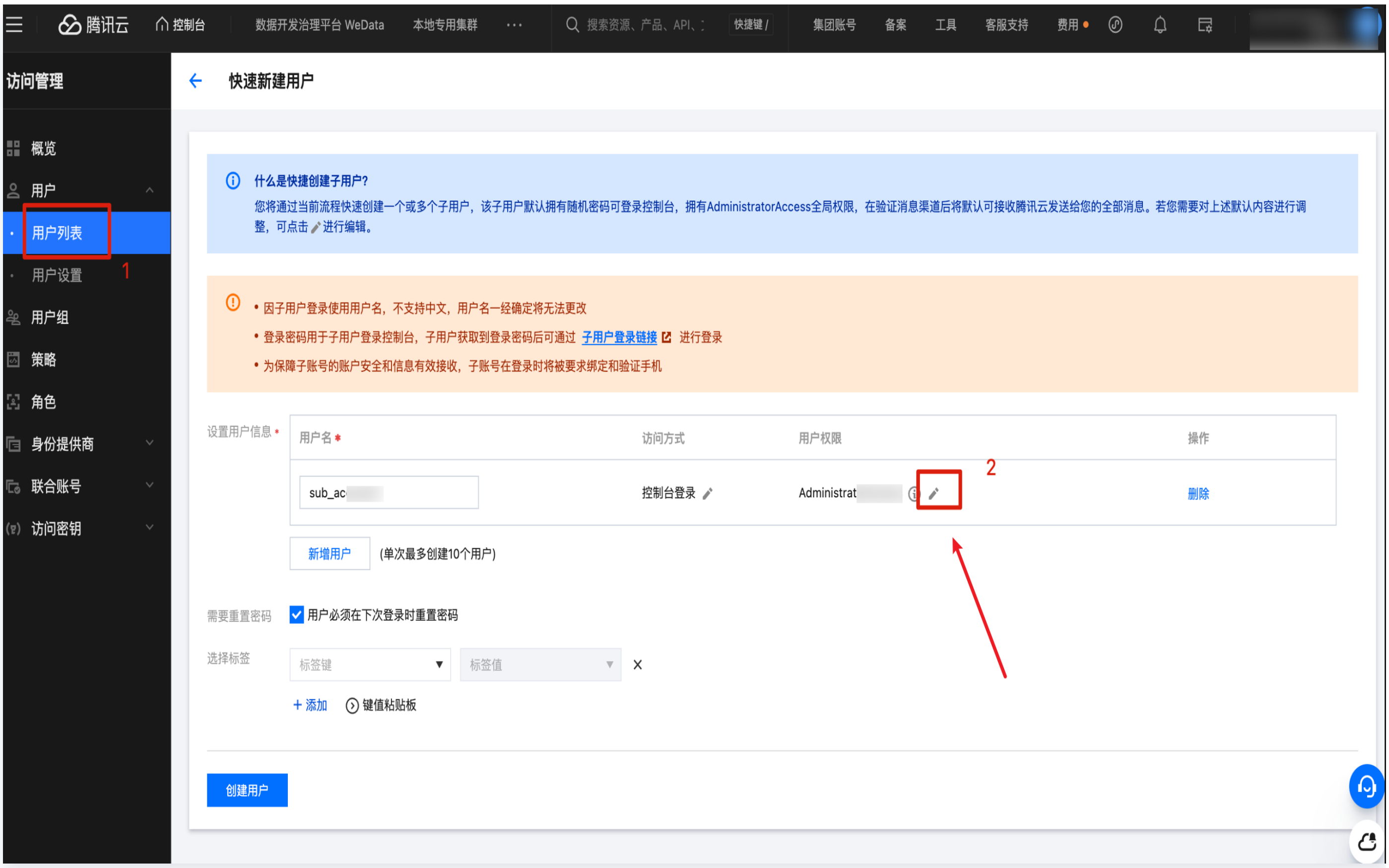
Enterprise Identity Verification Guide

After completing the registration, you need to go through enterprise authentication. The available authentication methods are as follows. For the detailed registration process, please refer to Tencent Cloud Enterprise Authentication.

Authentication Method	Authentication Duration	Note
WeChat Public Platform Authentication	Instant completion	Enterprises that have registered a WeChat Official Account and completed WeChat identity verification can use this method for immediate authentication.
Enterprise legal person WeChat scan code authentication	Instant completion	Use the personal WeChat of the enterprise legal person to scan the code for authentication. After the legal person authorizes via WeChat scan, the authentication is completed.
Enterprise legal person Face Recognition authentication	Instant completion	Use the personal WeChat of the enterprise legal person to scan the code for Face Recognition. After passing Face Recognition, the authentication is completed.
Tencent Cloud recharge authentication	1 business day	Transfer a small verification amount (less than 1 RMB) generated randomly by the system from the enterprise bank account (the amount will be added to the balance). Once Tencent Cloud receives the transfer, the authentication is completed.
Enterprise remittance authentication	1-5 business days	Enter the enterprise bank account information. After Tencent Cloud successfully transfers the money, enter the transferred amount to complete the authentication.

Create a Tencent Cloud sub-account

1. Go to the **Tencent Cloud console** > User List > **Create User** > **Rapid creation.** Modify user permissions, click the **Edit Icon** for User Permissions.



快速新建用户

什么是快捷创建子用户?
您将通过当前流程快速创建一个或多个子用户，该子用户默认拥有随机密码可登录控制台，拥有AdministratorAccess全局权限，在验证消息渠道后将默认可接收腾讯云发送给您的全部消息。若您需要对上述默认内容进行调整，可点击 进行编辑。

- 因子用户登录使用用户名，不支持中文，用户名一经确定将无法更改
- 登录密码用于子用户登录控制台，子用户获取到登录密码后可通过 子用户登录链接 进行登录
- 为保障子账号的账户安全和信息有效接收，子账号在登录时将被要求绑定和验证手机

设置用户信息 *

用户名 *	访问方式	用户权限	操作
sub_ac[redacted]	控制台登录	Administrat[redacted]	删除

新增用户 (单次最多创建10个用户)

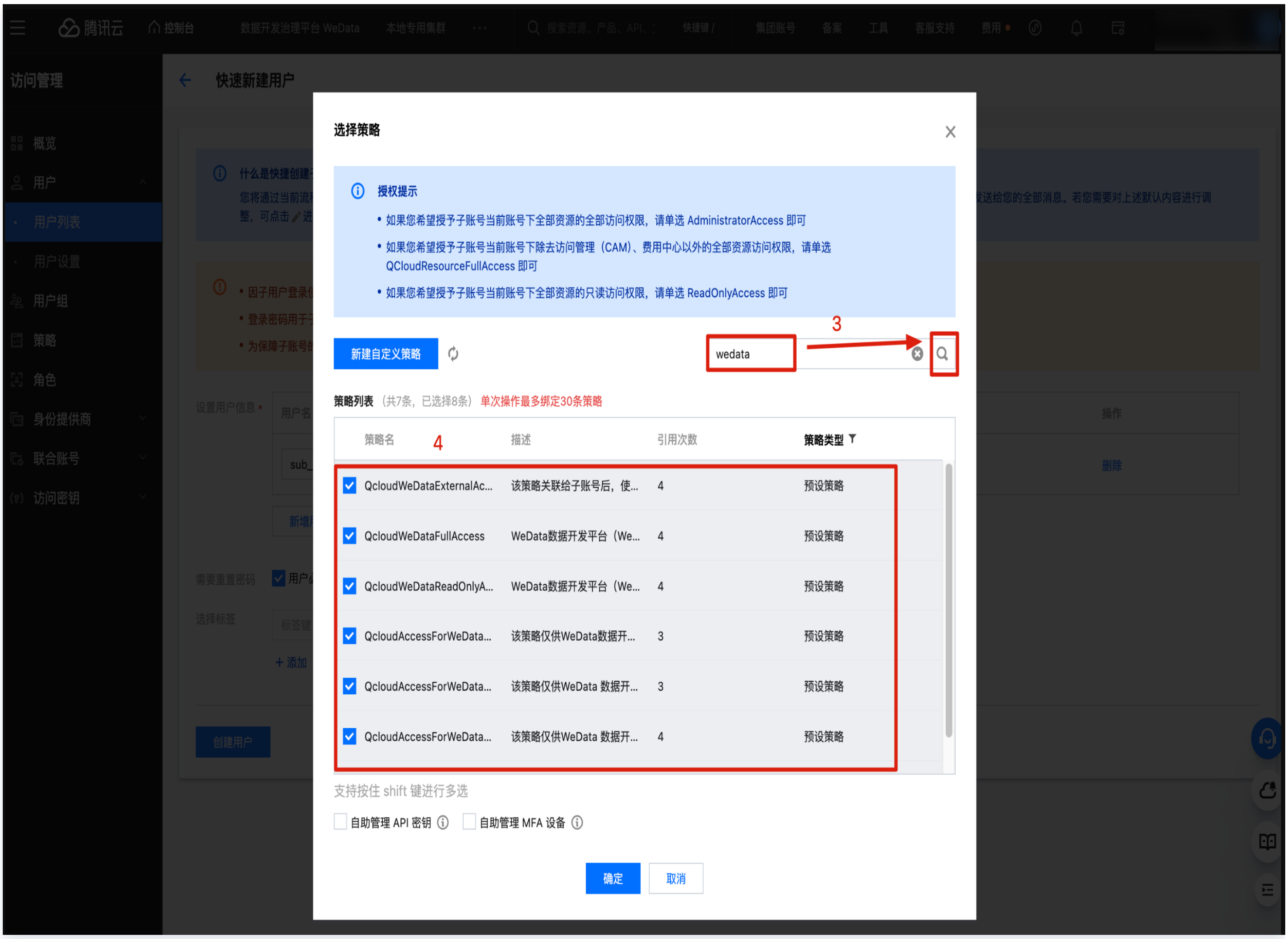
需要重置密码 ☑ 用户必须在下次登录时重置密码

选择标签 标签键 标签值 X

+ 添加 键值粘贴板

创建用户

2. Enter wedata, click the search icon. After selecting all policies, click **OK**.



选择策略

授权提示

- 如果您希望授予子账号当前账号下全部资源的全部访问权限，请单选 AdministratorAccess 即可
- 如果您希望授予子账号当前账号下除去访问管理（CAM）、费用中心以外的全部资源访问权限，请单选 QCloudResourceFullAccess 即可
- 如果您希望授予子账号当前账号下全部资源的只读访问权限，请单选 ReadOnlyAccess 即可

wedata

策略列表（共7条，已选择8条）单次操作最多绑定30条策略

	策略名	描述	引用次数	策略类型
☑	QcloudWeDataExternalAc...	该策略关联给子账号后，使...	4	预设策略
☑	QcloudWeDataFullAccess	WeData数据开发平台（We...	4	预设策略
☑	QcloudWeDataReadOnlyA...	WeData数据开发平台（We...	4	预设策略
☑	QcloudAccessForWeData...	该策略仅供WeData数据开...	3	预设策略
☑	QcloudAccessForWeData...	该策略仅供WeData 数据开...	3	预设策略
☑	QcloudAccessForWeData...	该策略仅供WeData 数据开...	4	预设策略

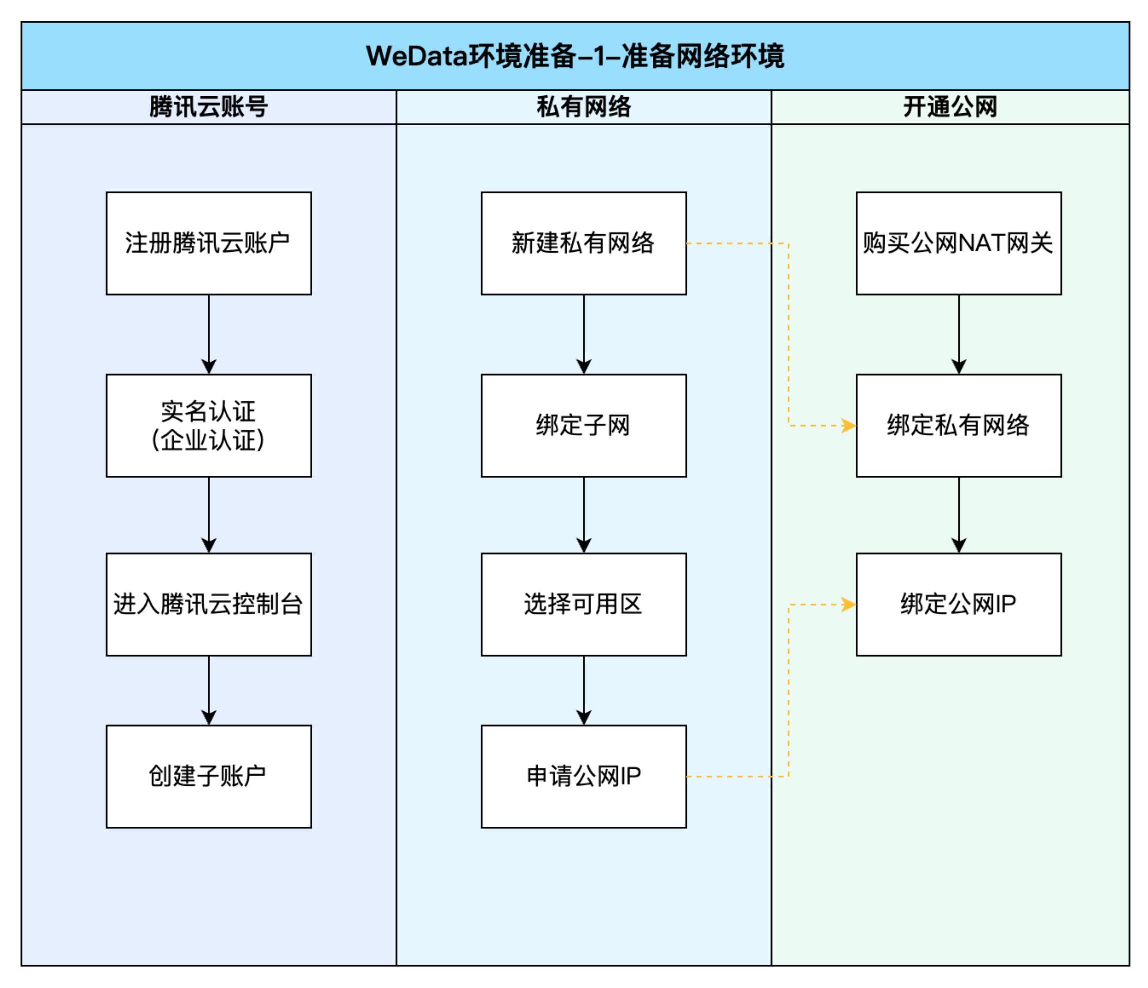
支持按住 shift 键进行多选

□ 自助管理 API 密钥 □ 自助管理 MFA 设备

Prepare the network environment

This tutorial involves multiple cloud resources. To ensure network connectivity, you need to set up a VPC environment.

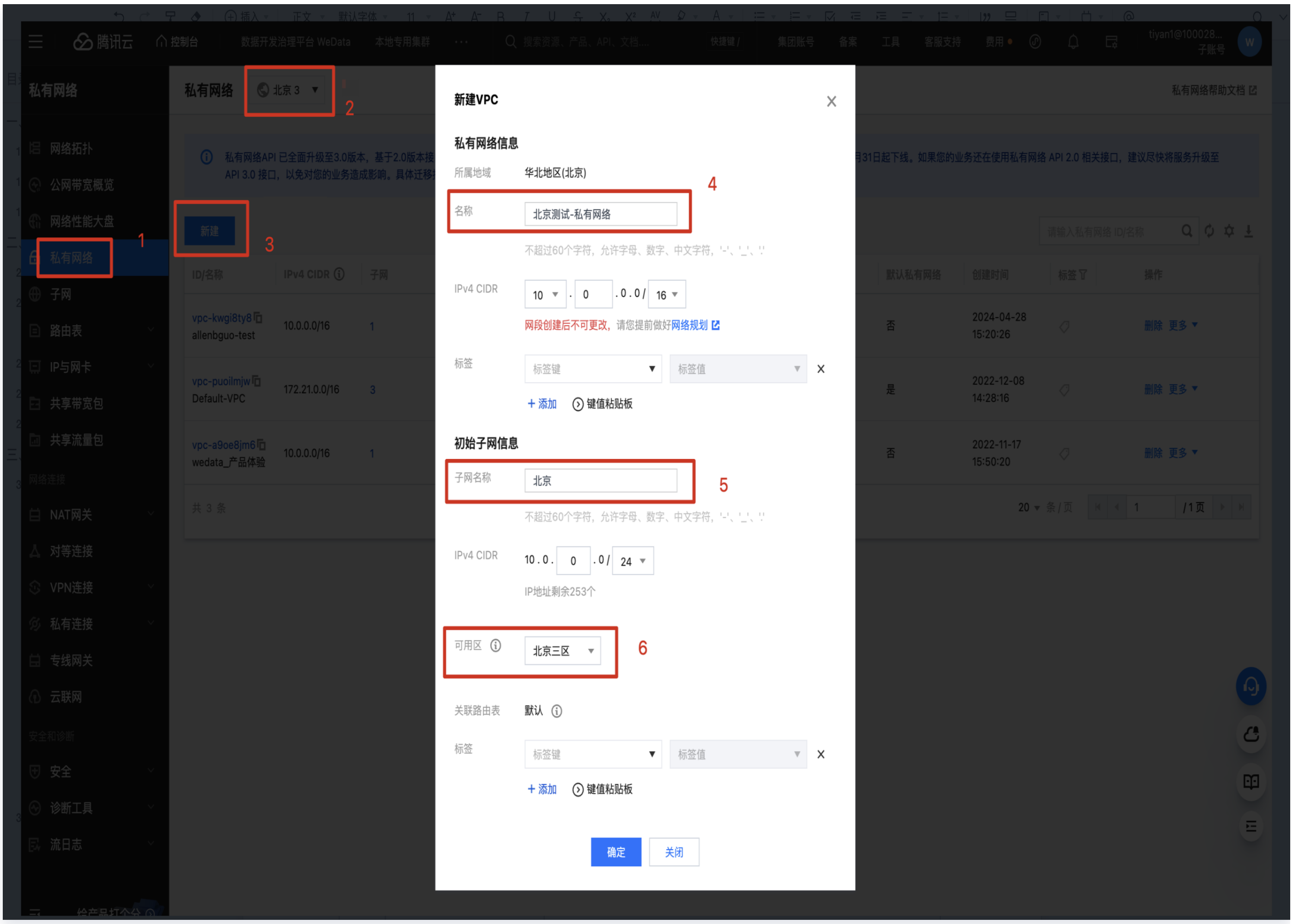
- **Role:** Enterprise Administrator.
- **Account:** Tencent Cloud primary account.
- **Steps:**



Create a VPC

1. Log in to Tencent Cloud VPC Console. At the top of the **VPC** page, select the VPC's region, for example, select Beijing, and click **Create.**
2. Enter the new VPC interface, fill in the VPC information and initial subnet information. After completing the form, click **OK.**
   - VPC Name: You can name it anything for easy distinction. Example: Beijing - VPC.
   - Subnet Name: You can name it anything for easy distinction. It is recommended to match the optional zone below. Example: Beijing Zone 3.
   - Optional Zone: You can choose any zone, for example, Beijing Zone 3. When purchasing other resources later, if this zone is not available, you can add a subnet in the VPC.

> **Note**
> - Choose the Beijing region. This is just an example. It is recommended to select a region that is closer to you.
> - All resources purchased later in this tutorial will be in the Beijing region, so please choose carefully.



新建VPC

私有网络信息

所属地域 华北地区(北京)

名称 北京测试-私有网络

不超过60个字符，允许字母、数字、中文字符，'-'、'_'、'.'

IPv4 CIDR 10 . 0 . 0 . 0 / 16

网段创建后不可更改，请您提前做好网络规划

标签 标签键 标签值

+添加 键值粘贴板

初始子网信息

子网名称 北京

不超过60个字符，允许字母、数字、中文字符，'-'、'_'、'.'

IPv4 CIDR 10 . 0 . 0 . 0 / 24

IP地址剩余253个

可用区 北京三区

关联路由表 默认

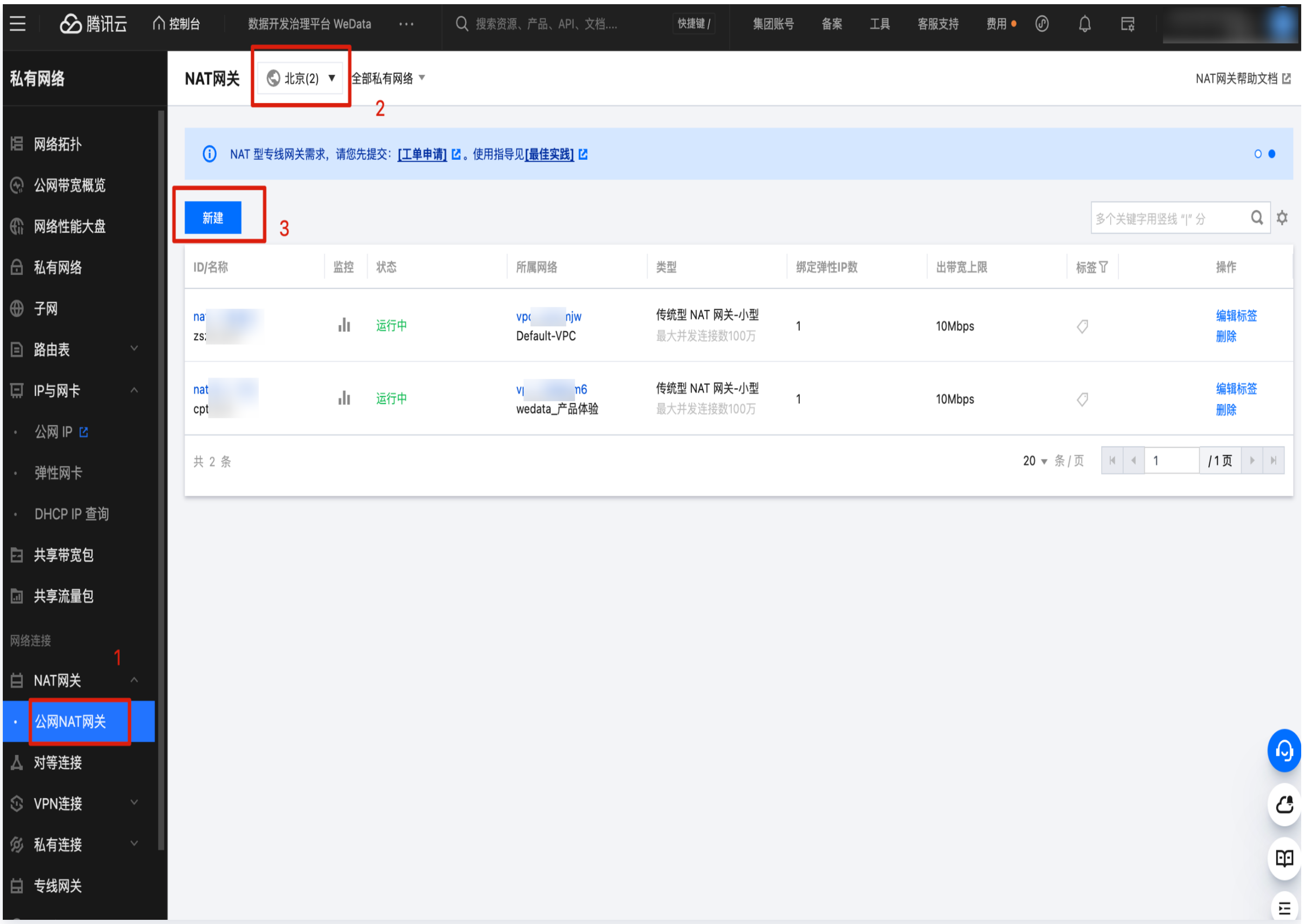
标签 标签键 标签值

+添加 键值粘贴板

确定 关闭

Purchase a NAT Gateway

1. Navigate to the Tencent Cloud Public Network NAT Gateway page, at the top of the **NAT Gateway** page, select the VPC region. For example: select Beijing, then click **Create.**

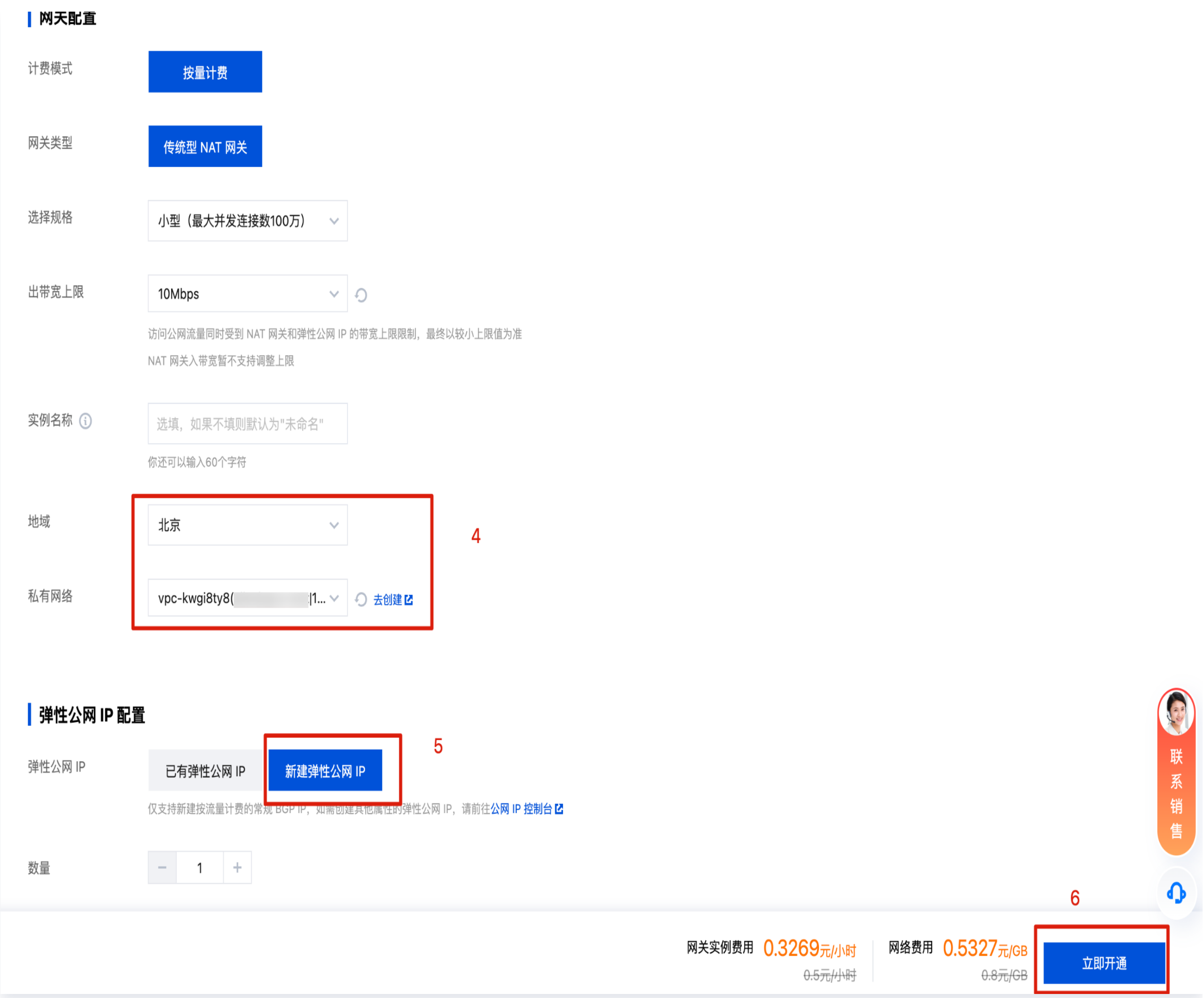


NAT 型专线网关需求，请您先提交：[工单申请]。使用指导见[最佳实践]

ID/名称	监控	状态	所属网络	类型	绑定弹性IP数	出带宽上限	标签	操作
na[redacted] zs[redacted]		运行中	vp[redacted]njw Default-VPC	传统型 NAT 网关-小型 最大并发连接数100万	1	10Mbps		编辑标签 删除
nat[redacted] cpt[redacted]		运行中	v[redacted]n6 wedata_产品体验	传统型 NAT 网关-小型 最大并发连接数100万	1	10Mbps		编辑标签 删除

共 2 条　20 条 / 页　1 /1 页

2. If you do not have a NAT Gateway, please go to the purchase page. After selecting the configuration, click **Enable Now**. Verify the bill, and complete the payment to enable it.
   - Region: Select Beijing,
   - VPC: Select the newly created VPC,
   - EIP: Select a new EIP. If you have already applied for a public IP, you can bind it here directly.



网关配置

计费模式 按量计费

网关类型 传统型 NAT 网关

选择规格 小型（最大并发连接数100万）

出带宽上限 10Mbps

访问公网流量同时受到 NAT 网关和弹性公网 IP 的带宽上限限制，最终以较小上限值为准

NAT 网关入带宽暂不支持调整上限

实例名称 选填，如果不填则默认为"未命名"

你还可以输入60个字符

地域 北京

私有网络 vpc-kwgi8ty8([redacted]|1...

弹性公网 IP 配置

弹性公网 IP 已有弹性公网 IP 新建弹性公网 IP

仅支持新建按流量计费的常规 BGP IP，如需创建其他属性的弹性公网 IP，请前往公网 IP 控制台

数量 1

网关实例费用 0.3269元/小时 ~~0.5元/小时~~

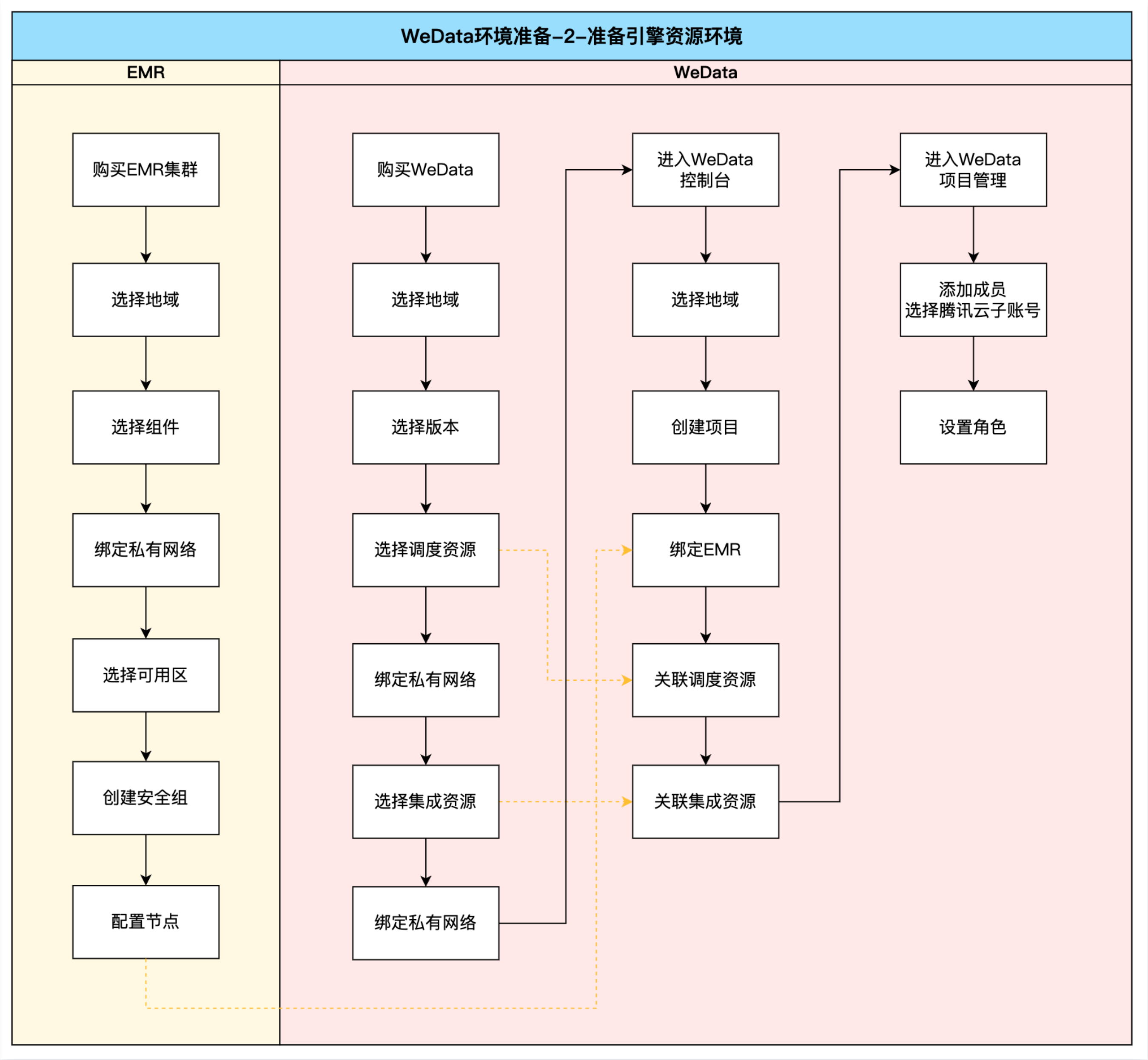
网络费用 0.5327元/GB ~~0.8元/GB~~

Prepare the engine resource environment

WeData, as a data development and management platform, needs to be bound with Tencent Cloud Big Data Suite as the data storage and computing engine, such as Tencent Cloud EMR, DLC, TCHouse, etc.

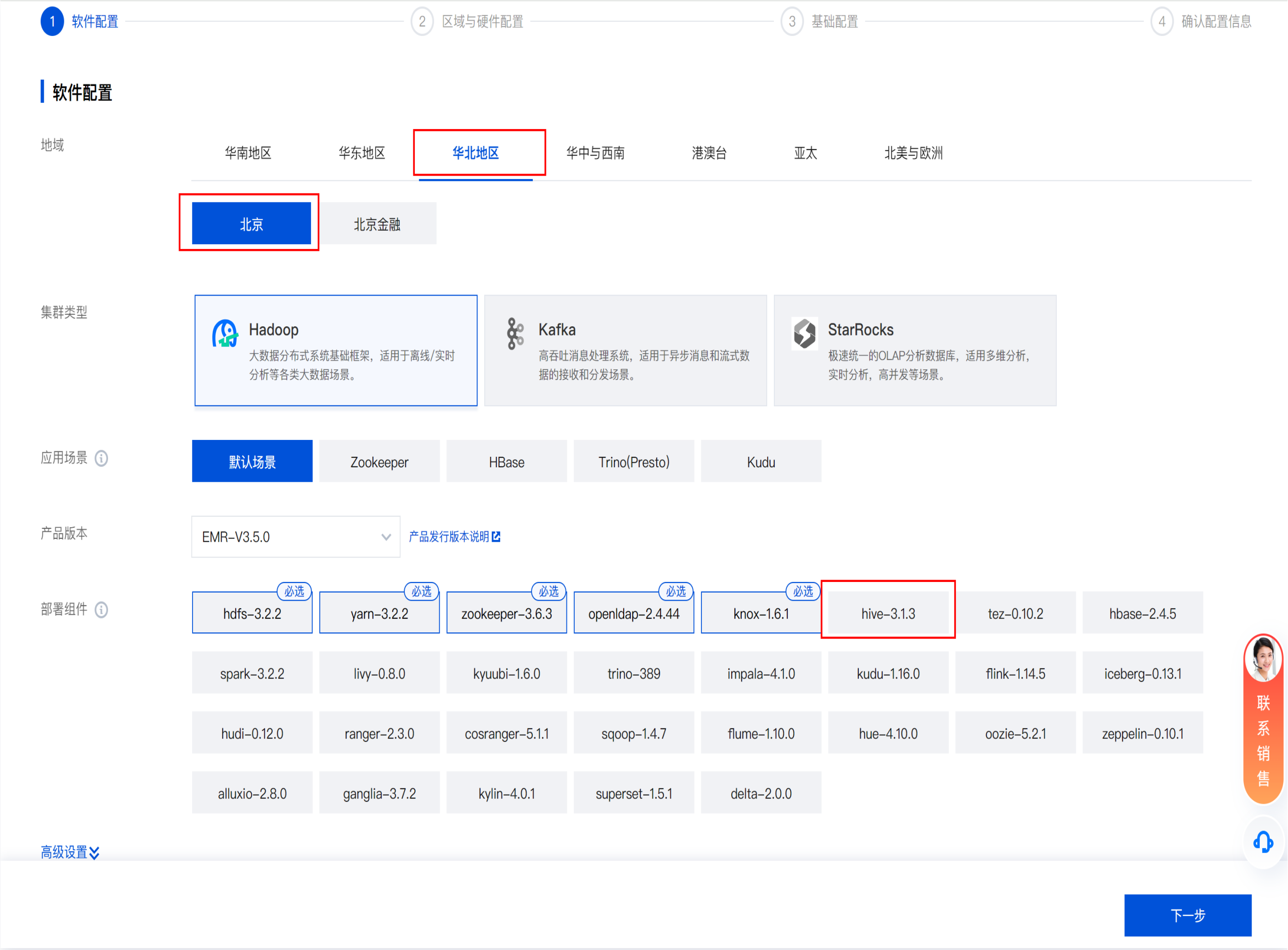
In this tutorial, EMR is used as an example to introduce WeData's data synchronization and development process. Therefore, we need to purchase an EMR environment on Tencent Cloud first.

- **Role:** Enterprise Administrator.
- **Account:** Tencent Cloud primary account.
- **Steps:**



Purchase EMR

1. Enter the Tencent Cloud EMR purchase page, first select the software configuration. After selection, click **Next**.
   - Region: Select North China - Beijing
   - Scenarios: Default scenario
   - Deployment Components: Choose Hive-3.1.3. In this tutorial, Hive is used as the storage and computing engine.



1 软件配置 — 2 区域与硬件配置 — 3 基础配置 — 4 确认配置信息

软件配置

地域: 华南地区 | 华东地区 | **华北地区** | 华中与西南 | 港澳台 | 亚太 | 北美与欧洲

北京 | 北京金融

集群类型:
- Hadoop: 大数据分布式系统基础框架，适用于离线/实时分析等各类大数据场景。
- Kafka: 高吞吐消息处理系统，适用于异步消息和流式数据的接收和分发场景。
- StarRocks: 极速统一的OLAP分析数据库，适用多维分析，实时分析，高并发等场景。

应用场景: **默认场景** | Zookeeper | HBase | Trino(Presto) | Kudu

产品版本: EMR-V3.5.0　产品发行版本说明

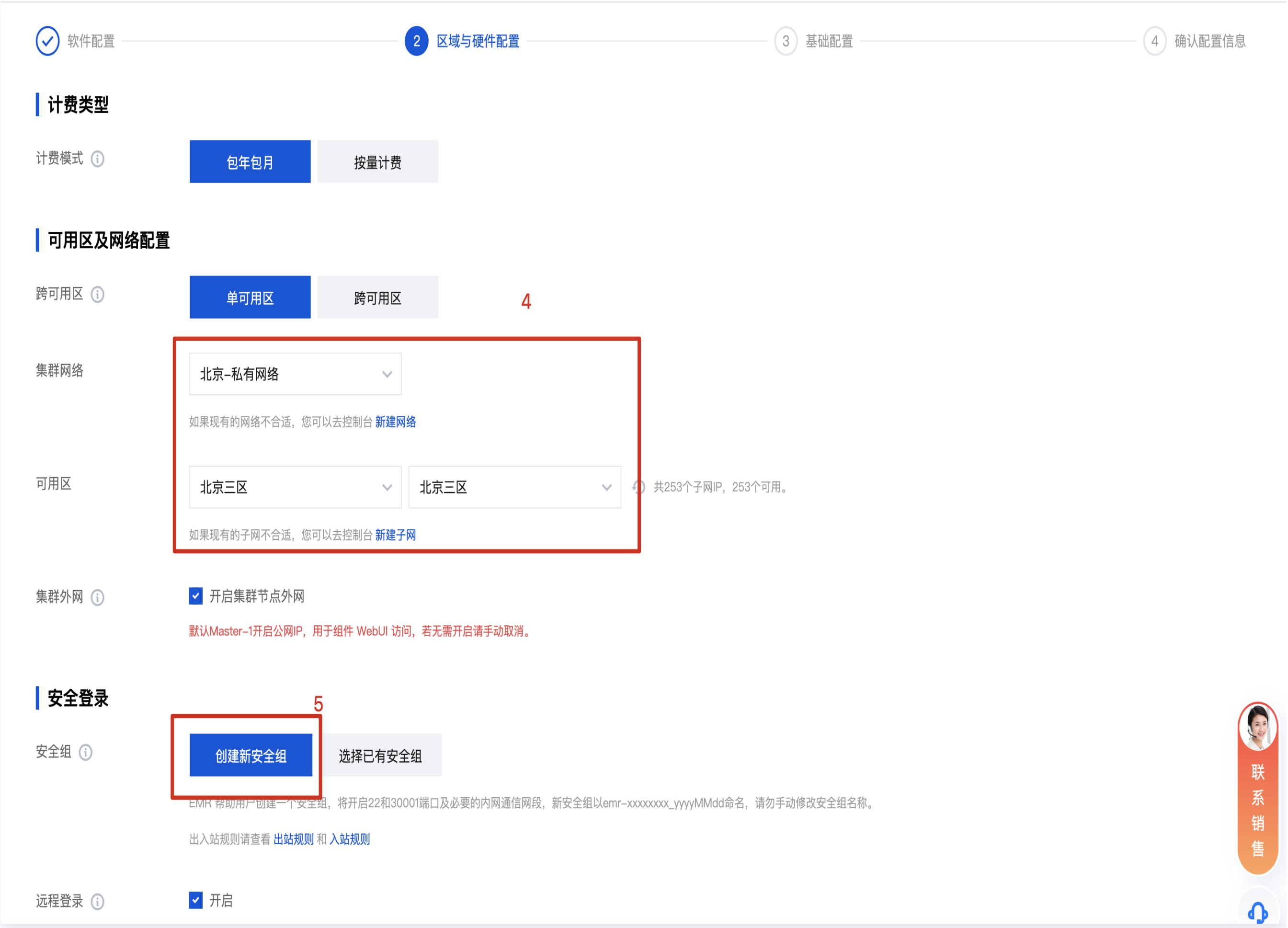
部署组件:

hdfs-3.2.2 必选	yarn-3.2.2 必选	zookeeper-3.6.3 必选	openldap-2.4.44 必选	knox-1.6.1 必选	hive-3.1.3	tez-0.10.2	hbase-2.4.5
spark-3.2.2	livy-0.8.0	kyuubi-1.6.0	trino-389	impala-4.1.0	kudu-1.16.0	flink-1.14.5	iceberg-0.13.1
hudi-0.12.0	ranger-2.3.0	cosranger-5.1.1	sqoop-1.4.7	flume-1.10.0	hue-4.10.0	oozie-5.2.1	zeppelin-0.10.1
alluxio-2.8.0	ganglia-3.7.2	kylin-4.0.1	superset-1.5.1	delta-2.0.0			

高级设置

下一步

2. In the second step, select the region and hardware configuration. After selection, click **Next** .
   - Cluster Network: Select the newly created VPC.
   - Availability Zone: Select the availability zone where the subnet of the VPC is located. If it is not available, return to the VPC page to bind the subnet.
   - Security Group: Here, the default is to create a new security group.



计费类型

计费模式 ⓘ 包年包月 | 按量计费

可用区及网络配置

跨可用区 ⓘ 单可用区 | 跨可用区

集群网络 北京-私有网络

如果现有的网络不合适，您可以去控制台 新建网络

可用区 北京三区 | 北京三区 共253个子网IP，253个可用。

如果现有的子网不合适，您可以去控制台 新建子网

集群外网 ⓘ ☑ 开启集群节点外网

默认Master-1开启公网IP，用于组件 WebUI 访问，若无需开启请手动取消。

安全登录

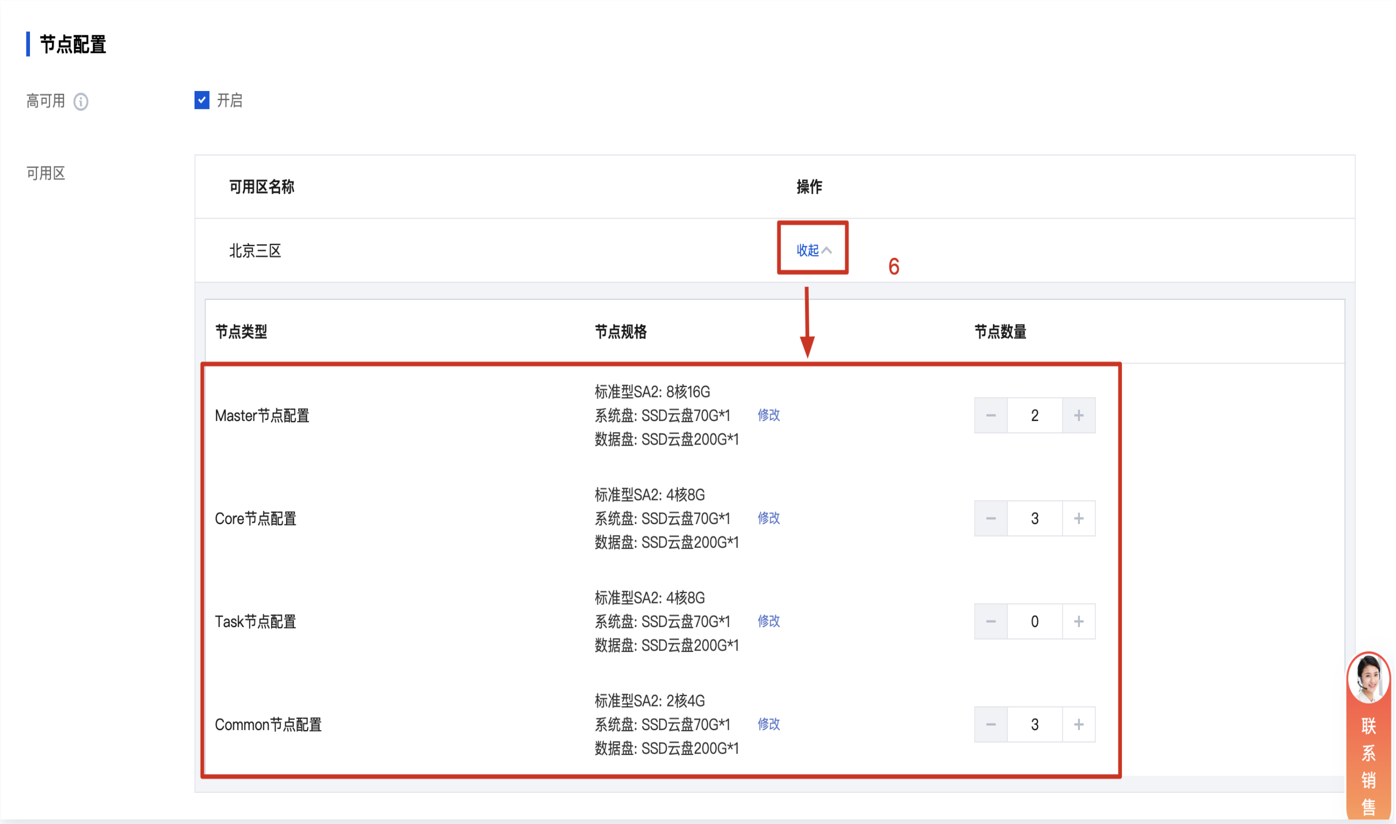
安全组 ⓘ 创建新安全组 | 选择已有安全组

EMR 帮助用户创建一个安全组，将开启22和30001端口及必要的内网通信网段，新安全组以emr-xxxxxxxx_yyyyMMdd命名，请勿手动修改安全组名称。

出入站规则请查看 出站规则 和 入站规则

远程登录 ⓘ ☑ 开启

3. Enter the node configuration page, expand the details, set the number of nodes, and use the default configuration.



节点配置

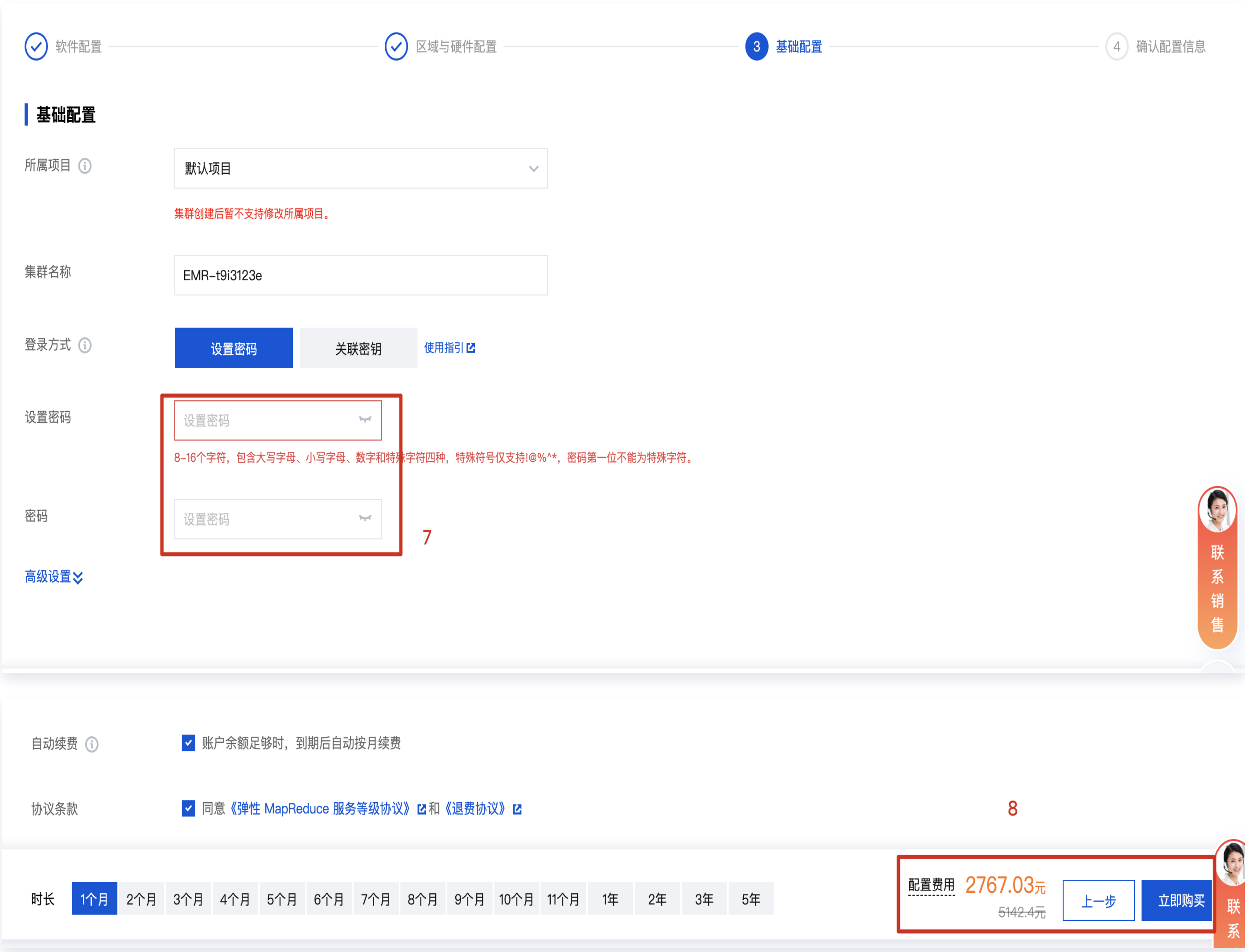
高可用 ⓘ ☑ 开启

可用区

可用区名称	操作
北京三区	收起 ^

节点类型	节点规格	节点数量
Master节点配置	标准型SA2: 8核16G 系统盘: SSD云盘70G*1 修改 数据盘: SSD云盘200G*1	− 2 +
Core节点配置	标准型SA2: 4核8G 系统盘: SSD云盘70G*1 修改 数据盘: SSD云盘200G*1	− 3 +
Task节点配置	标准型SA2: 4核8G 系统盘: SSD云盘70G*1 修改 数据盘: SSD云盘200G*1	− 0 +
Common节点配置	标准型SA2: 2核4G 系统盘: SSD云盘70G*1 修改 数据盘: SSD云盘200G*1	− 3 +

4. Enter the basic configuration interface, set the server password, check **auto-renewal** and **Terms of Service**, then click **Purchase Now**. Verify the bill and complete the payment to activate. You can refer to Password Setting Format.



软件配置 — 区域与硬件配置 — 3 基础配置 — 4 确认配置信息

基础配置

所属项目：默认项目

集群创建后暂不支持修改所属项目。

集群名称：EMR-t9i3123e

登录方式：设置密码 | 关联密钥 | 使用指引

设置密码：设置密码

8-16个字符，包含大写字母、小写字母、数字和特殊字符四种，特殊符号仅支持!@%^*，密码第一位不能为特殊字符。

密码：设置密码

7

高级设置

自动续费：☑ 账户余额足够时，到期后自动按月续费

协议条款：☑ 同意《弹性 MapReduce 服务等级协议》和《退费协议》

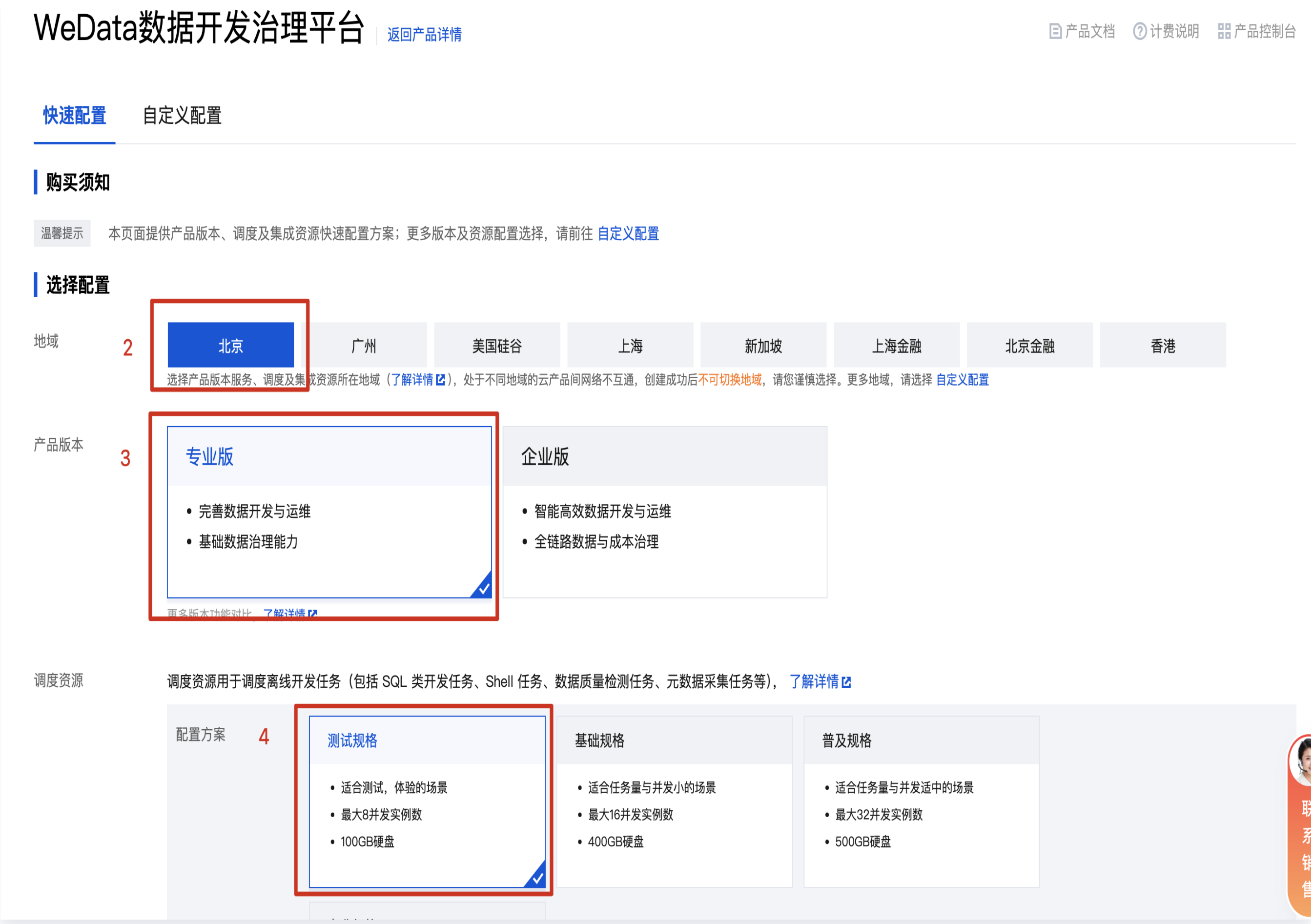
8

时长：1个月 | 2个月 | 3个月 | 4个月 | 5个月 | 6个月 | 7个月 | 8个月 | 9个月 | 10个月 | 11个月 | 1年 | 2年 | 3年 | 5年

配置费用 2767.03元 ~~5142.4元~~ 上一步 立即购买

Purchase WeData

1. Go to the Tencent Cloud WeData Purchase Page, complete the quick configuration, click **Purchase Now**, verify the bill, and complete the payment to activate.
   - Region: Select Beijing. It is recommended to choose a region closer to you.
   - Product Version: Select the professional version. For details, refer to WeData Version Differences.
   - Scheduling Resources: Select the test specification. For detailed information about scheduling resources, refer to Scheduling Resources Billing Overview.
   - Scheduling Resources Network: Select the newly created VPC.
   - Configuration Scheme: Select the basic specification. For detailed information about scheduling resources, refer to Integration Resources Billing Overview.
   - Network: Select the newly created VPC.



WeData数据开发治理平台 | 返回产品详情

购买须知

温馨提示 本页面提供产品版本、调度及集成资源快速配置方案；更多版本及资源配置选择，请前往 自定义配置

选择配置

地域 2

选择产品版本服务、调度及集成资源所在地域（了解详情），处于不同地域的云产品间网络不互通，创建成功后不可切换地域，请您谨慎选择。更多地域，请选择 自定义配置

产品版本 3

专业版
- 完善数据开发与运维
- 基础数据治理能力

企业版
- 智能高效数据开发与运维
- 全链路数据与成本治理

更多版本功能对比，了解详情

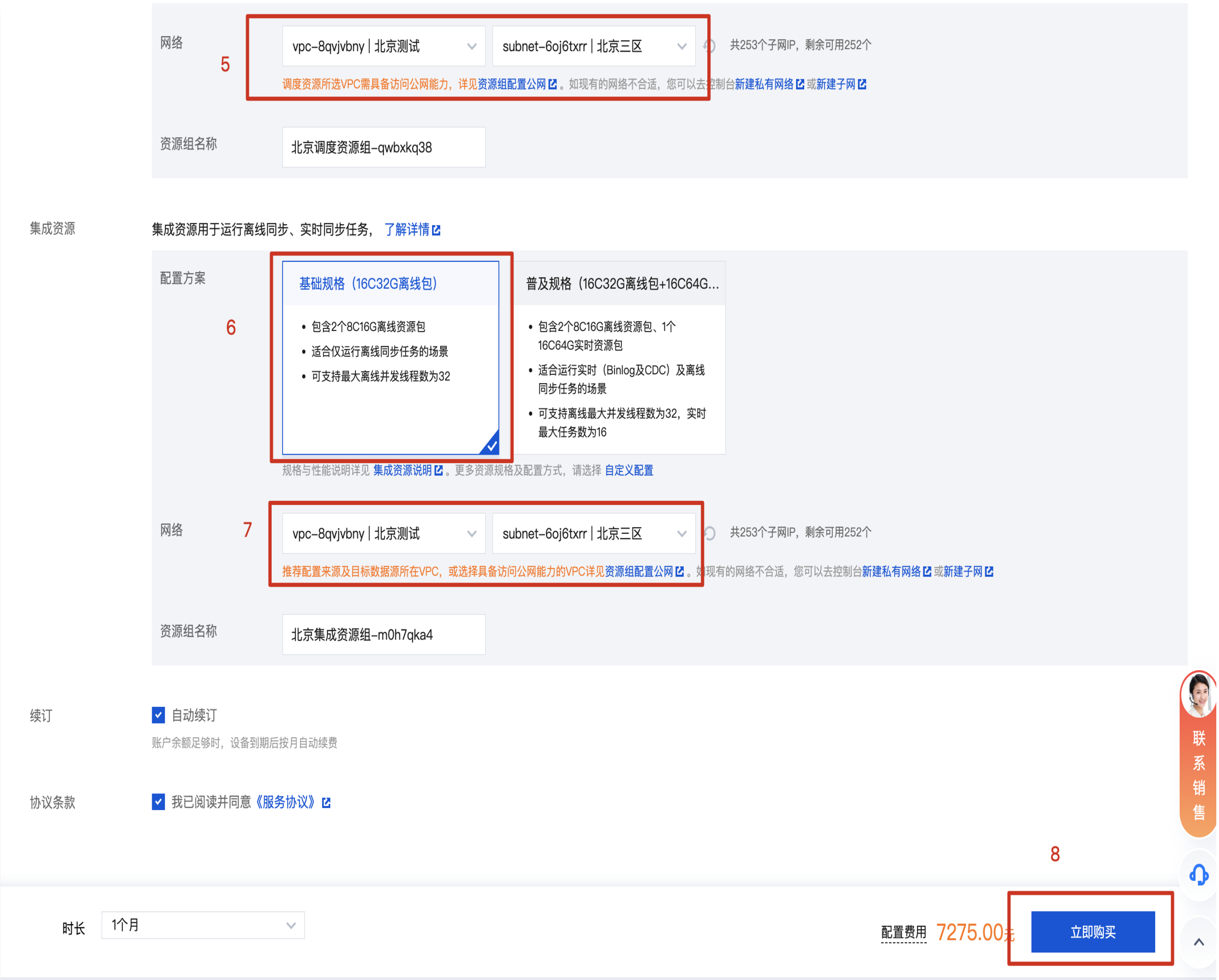
调度资源 调度资源用于调度离线开发任务（包括 SQL 类开发任务、Shell 任务、数据质量检测任务、元数据采集任务等），了解详情

配置方案 4

测试规格
- 适合测试，体验的场景
- 最大8并发实例数
- 100GB硬盘

基础规格
- 适合任务量与并发小的场景
- 最大16并发实例数
- 400GB硬盘

普及规格
- 适合任务量与并发适中的场景
- 最大32并发实例数
- 500GB硬盘



网络　vpc-8qvjvbny | 北京测试　subnet-6oj6txrr | 北京三区　共253个子网IP，剩余可用252个

调度资源所选VPC需具备访问公网能力，详见资源组配置公网。如现有的网络不合适，您可以去控制台新建私有网络或新建子网

资源组名称　北京调度资源组-qwbxkq38

集成资源　集成资源用于运行离线同步、实时同步任务，了解详情

配置方案

基础规格（16C32G离线包）
- 包含2个8C16G离线资源包
- 适合仅运行离线同步任务的场景
- 可支持最大离线并发线程数为32

普及规格（16C32G离线包+16C64G...
- 包含2个8C16G离线资源包、1个16C64G实时资源包
- 适合运行实时（Binlog及CDC）及离线同步任务的场景
- 可支持离线最大并发线程数为32，实时最大任务数为16

规格与性能说明详见 集成资源说明。更多资源规格及配置方式，请选择 自定义配置

网络　vpc-8qvjvbny | 北京测试　subnet-6oj6txrr | 北京三区　共253个子网IP，剩余可用252个

推荐配置来源及目标数据源所在VPC，或选择具备访问公网能力的VPC详见资源组配置公网。如现有的网络不合适，您可以去控制台新建私有网络或新建子网

资源组名称　北京集成资源组-m0h7qka4

续订　☑ 自动续订

账户余额足够时，设备到期后按月自动续费

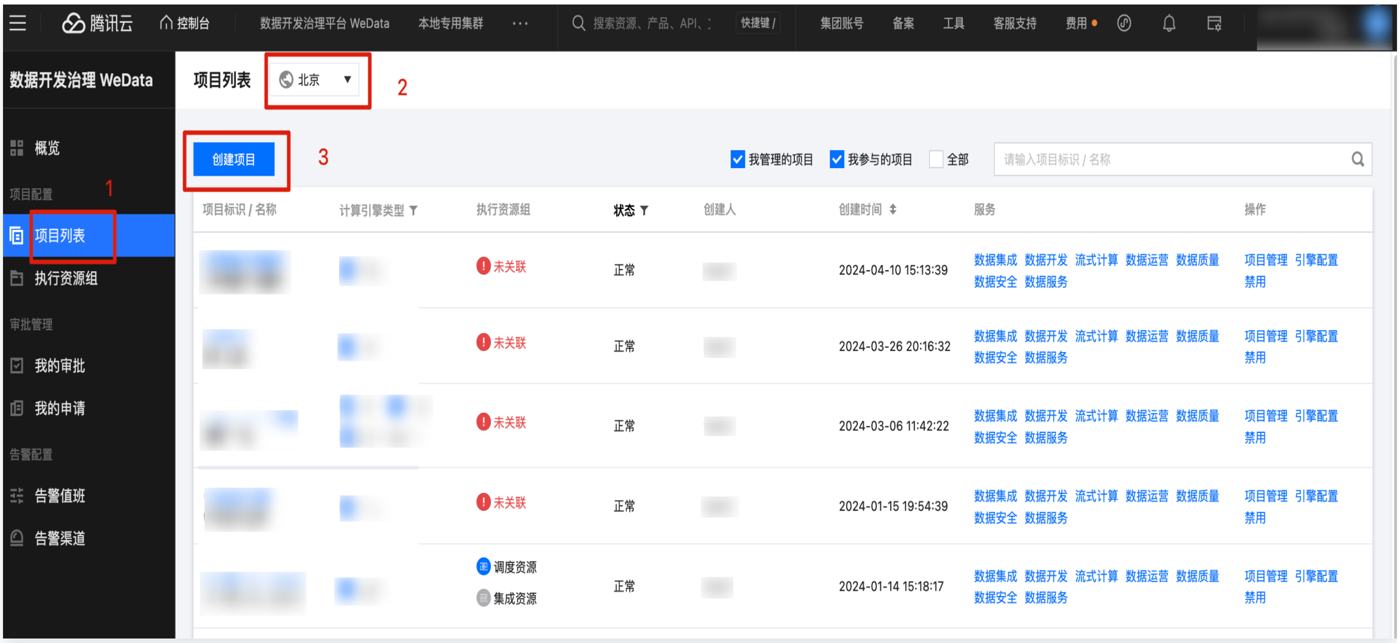
协议条款　☑ 我已阅读并同意《服务协议》

时长　1个月

配置费用　7275.00元

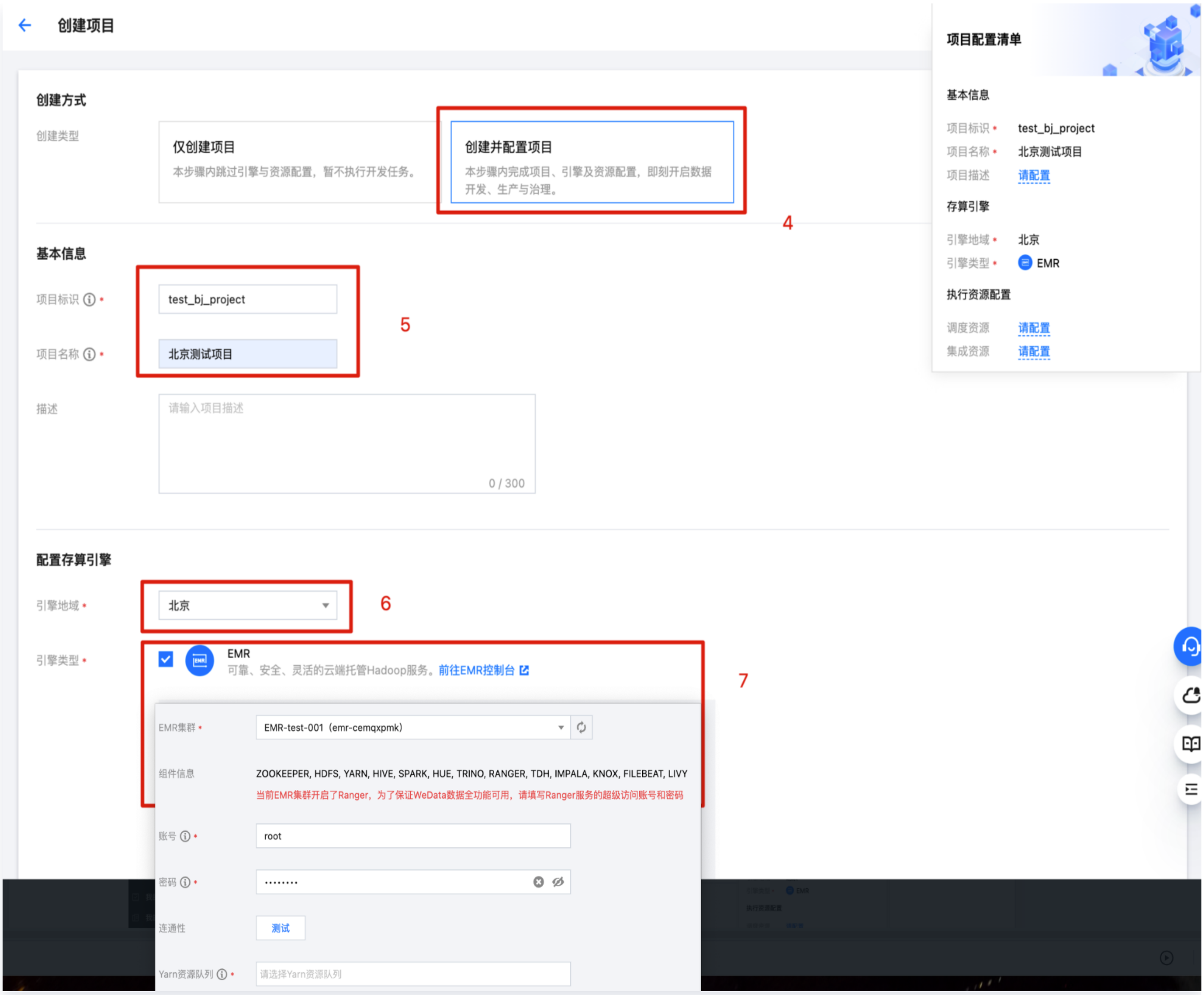
Create a project in WeData

1. Log in to Tencent Cloud WeData Console, click **Project List** on the left-side menu, enter the Project List interface, select the top region as **Beijing**, then click **Create Project**.



项目标识 / 名称	计算引擎类型	执行资源组	状态	创建人	创建时间	服务	操作
[redacted]	[redacted]	未关联	正常	[redacted]	2024-04-10 15:13:39	数据集成 数据开发 流式计算 数据运营 数据质量 数据安全 数据服务	项目管理 引擎配置 禁用
[redacted]	[redacted]	未关联	正常	[redacted]	2024-03-26 20:16:32	数据集成 数据开发 流式计算 数据运营 数据质量 数据安全 数据服务	项目管理 引擎配置 禁用
[redacted]	[redacted]	未关联	正常	[redacted]	2024-03-06 11:42:22	数据集成 数据开发 流式计算 数据运营 数据质量 数据安全 数据服务	项目管理 引擎配置 禁用
[redacted]	[redacted]	未关联	正常	[redacted]	2024-01-15 19:54:39	数据集成 数据开发 流式计算 数据运营 数据质量 数据安全 数据服务	项目管理 引擎配置 禁用
[redacted]	[redacted]	调度资源 集成资源	正常	[redacted]	2024-01-14 15:18:17	数据集成 数据开发 流式计算 数据运营 数据质量 数据安全 数据服务	项目管理 引擎配置 禁用

2. Enter the project creation interface, select and fill in the relevant information, then click **Confirm** to complete the project creation.
   - **Creation Method:** Select to create and configure the project.
   - **Basic Information:**
     - Project Identification: Enter any text for easy distinction, e.g., test_bj_project.
     - Project Name: Enter any text for easy distinction, e.g., Beijing Test Project.
   - **Configure Storage and Computing Engine:** Engine Region: Select Beijing.
   - **Engine Type:** Select **EMR.**
     - EMR Cluster: Bind it directly from the dropdown. If you've completed the purchase in step 2.3.1, the EMR cluster name will be displayed here.
     - Account: The default is root.
     - Password: The one set during Purchase EMR.
     - Connectivity: Please click **Testing.**
     - Yarn Resource Queue: The default is `default`.
     - Engine Metadata Collection: Yes.



创建项目

创建方式

创建类型

仅创建项目
本步骤内跳过引擎与资源配置，暂不执行开发任务。

创建并配置项目
本步骤内完成项目、引擎及资源配置，即刻开启数据开发、生产与治理。

4

基本信息

项目标识 * test_bj_project

项目名称 * 北京测试项目

5

描述 请输入项目描述 0 / 300

配置存算引擎

引擎地域 * 北京

6

引擎类型 * ☑ EMR
可靠、安全、灵活的云端托管Hadoop服务。前往EMR控制台

7

EMR集群 * EMR-test-001（emr-cemqxpmk）

组件信息 ZOOKEEPER, HDFS, YARN, HIVE, SPARK, HUE, TRINO, RANGER, TDH, IMPALA, KNOX, FILEBEAT, LIVY

当前EMR集群开启了Ranger，为了保证WeData数据全功能可用，请填写Ranger服务的超级访问账号和密码

账号 * root

密码 * ·········

连通性 测试

Yarn资源队列 * 请选择Yarn资源队列

项目配置清单

基本信息

项目标识 * test_bj_project

项目名称 * 北京测试项目

项目描述 请配置

存算引擎

引擎地域 * 北京

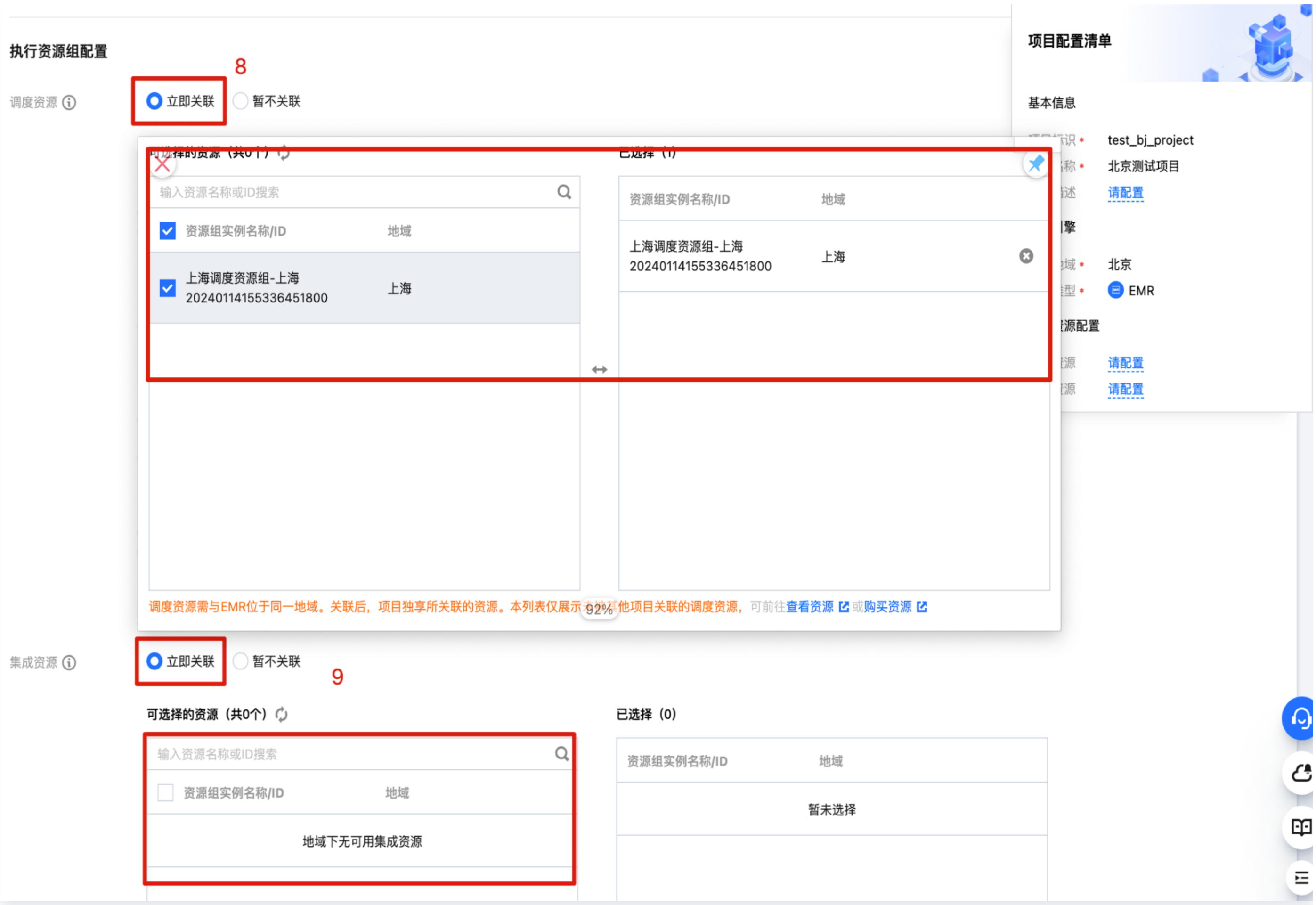
引擎类型 * EMR

执行资源配置

调度资源 请配置

集成资源 请配置

- **Scheduling Resources:** Select **Immediately Associate,** check the previously created scheduling resources for binding , and the available resource groups will be displayed here.
- **Integration Resources:** Select **Immediately Associate,** check the previously created integration resources for binding , and the available resource groups will be displayed here.



执行资源组配置

8

调度资源 ⓘ ● 立即关联 ○ 暂不关联

可选择的资源（共0个）

输入资源名称或ID搜索

☑ 资源组实例名称/ID	地域
☑ 上海调度资源组-上海 20240114155336451800	上海

已选择（1）

资源组实例名称/ID	地域
上海调度资源组-上海 20240114155336451800	上海

调度资源需与EMR位于同一地域。关联后，项目独享所关联的资源。本列表仅展示未被其他项目关联的调度资源，可前往查看资源或购买资源

集成资源 ⓘ ● 立即关联 ○ 暂不关联

9

可选择的资源（共0个）

输入资源名称或ID搜索

□ 资源组实例名称/ID	地域
地域下无可用集成资源	

已选择（0）

资源组实例名称/ID	地域
暂未选择	

项目配置清单

基本信息

test_bj_project

北京测试项目

请配置

北京

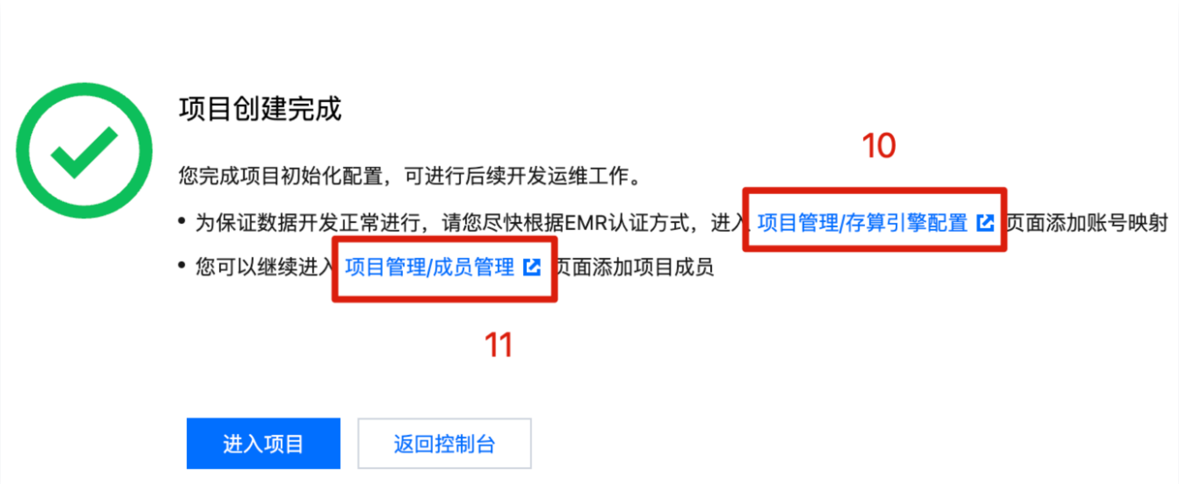
EMR

源配置

请配置

请配置

3. After the project is created, you can configure the account and add members by clicking **Project Management/Storage and Computing Engine Configuration** and **Project Management/Member Management.**

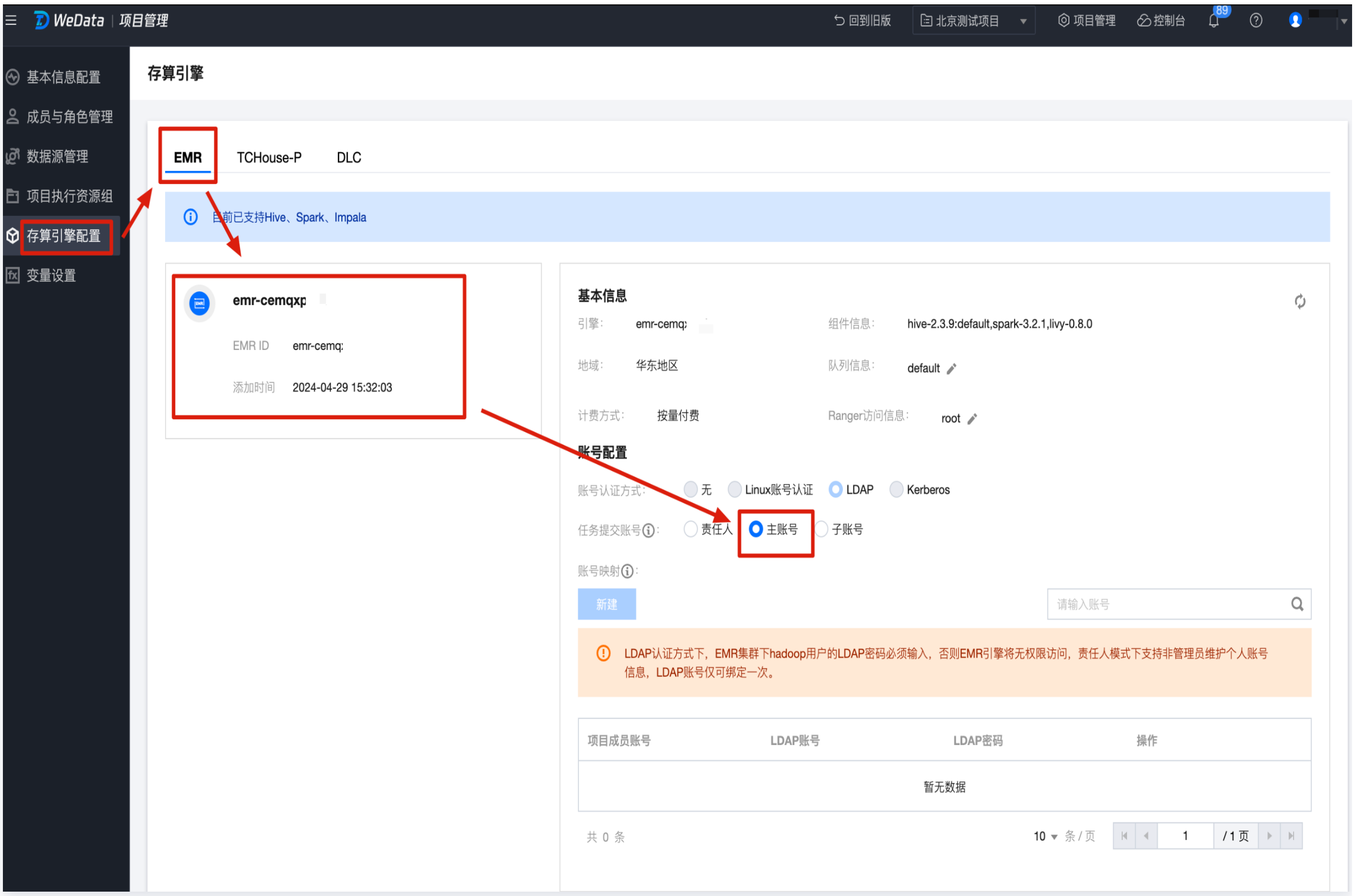


项目创建完成

您完成项目初始化配置，可进行后续开发运维工作。

- 为保证数据开发正常进行，请您尽快根据EMR认证方式，进入 项目管理/存算引擎配置 页面添加账号映射
- 您可以继续进入 项目管理/成员管理 页面添加项目成员

4. **Storage and Computing Engine Configuration:** Enter the storage and computing engine configuration interface, set EMR as the **primary account.**



存算引擎

目前已支持Hive、Spark、Impala

emr-cemqxp

EMR ID emr-cemq

添加时间 2024-04-29 15:32:03

基本信息

引擎: emr-cemq

组件信息: hive-2.3.9:default,spark-3.2.1,livy-0.8.0

地域: 华东地区

队列信息: default

计费方式: 按量付费

Ranger访问信息: root

账号配置

账号认证方式: ○ 无 ○ Linux账号认证 ● LDAP ○ Kerberos

任务提交账号: ○ 责任人 ● 主账号 ○ 子账号

账号映射:

请输入账号

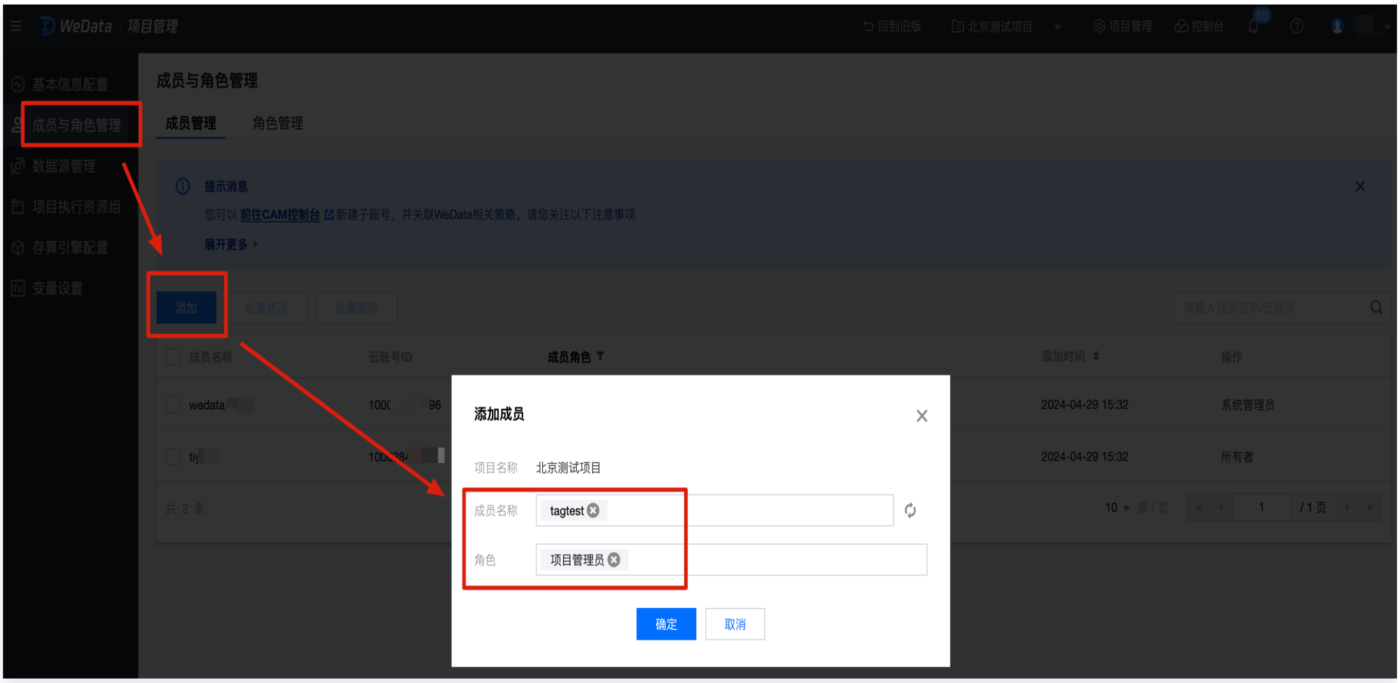
LDAP认证方式下，EMR集群下hadoop用户的LDAP密码必须输入，否则EMR引擎将无权限访问，责任人模式下支持非管理员维护个人账号信息，LDAP账号仅可绑定一次。

项目成员账号	LDAP账号	LDAP密码	操作
暂无数据			

共 0 条 10 条 / 页 1 / 1 页

5. **Member Configuration:** In the Member and Role Management page, click **Add,** enter the Add Member page, and add the Tencent Cloud sub-account as the project administrator.

> **Note**
> This sub-account can perform subsequent data synchronization and data development operations.



添加成员

项目名称 北京测试项目

成员名称 tagtest

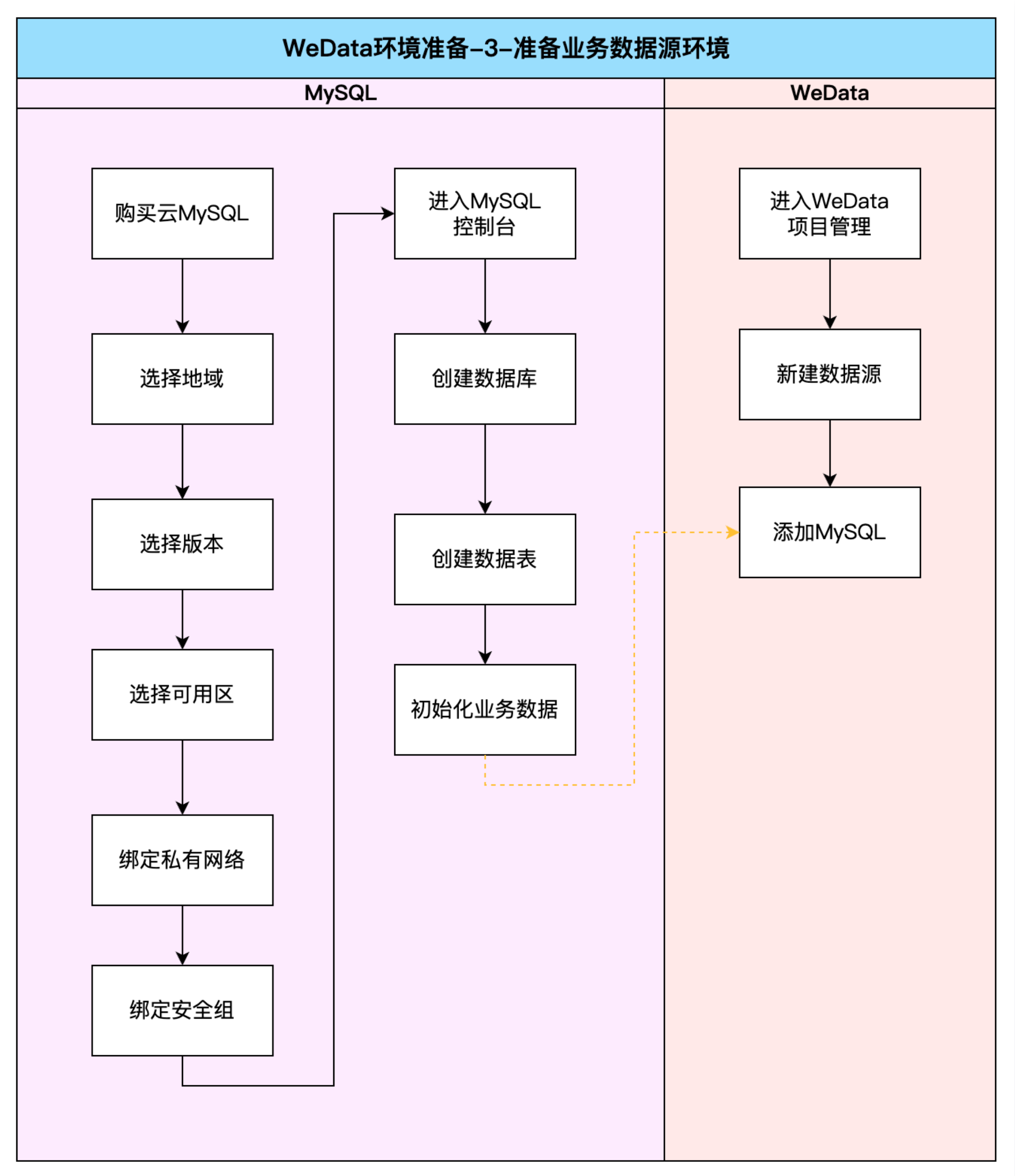
角色 项目管理员

Prepare the business data resource environment

In this tutorial, we simulate an e-commerce mall order data synchronization and analysis scenario, so we need to prepare the original data of the e-commerce mall.

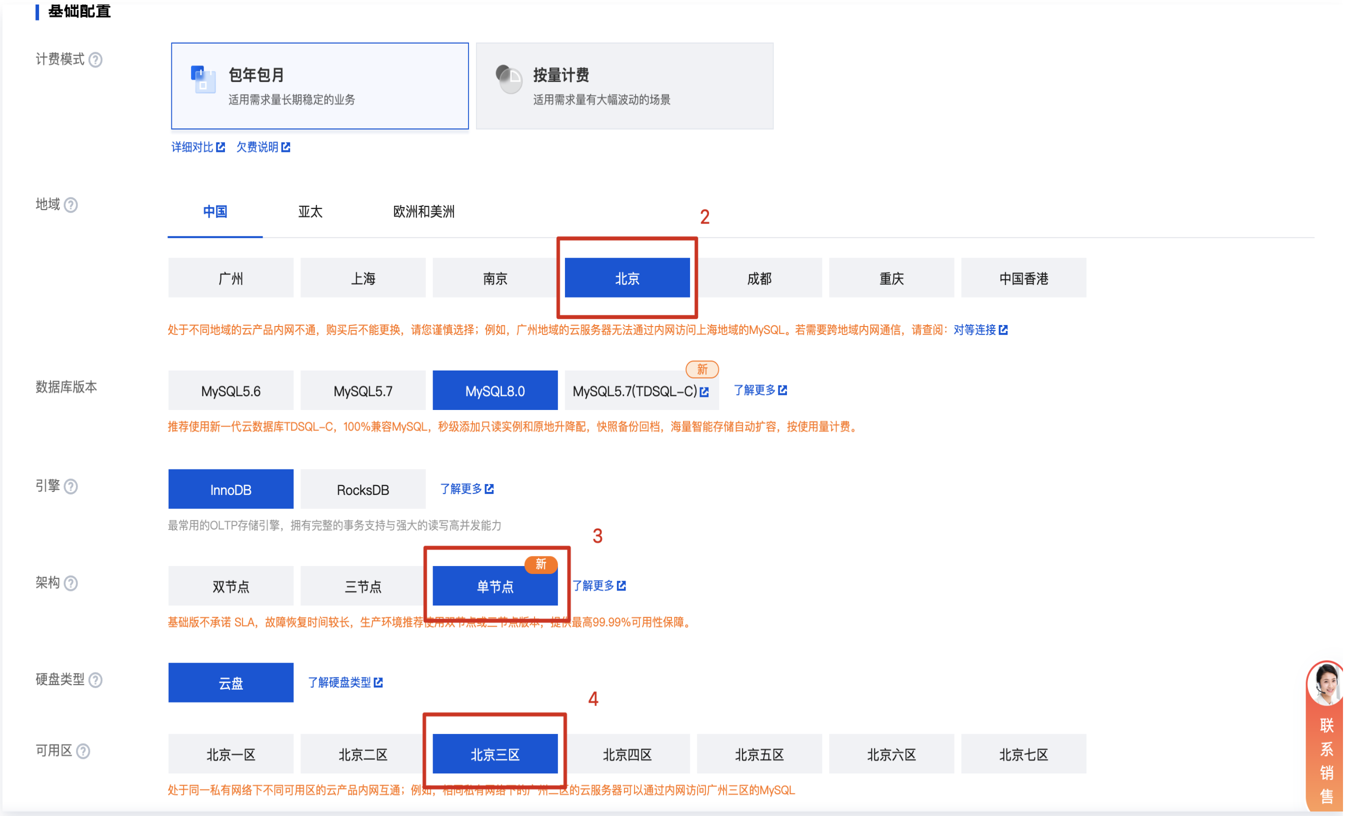
In this tutorial, Tencent Cloud MySQL is used as an example to introduce WeData's data synchronization process. Therefore, we need to purchase a MySQL database on Tencent Cloud first.

- **Role:** Enterprise Administrator.
- **Account:** Tencent Cloud primary account.
- **Steps:**



Purchase MySQL

1. Go to the TencentDB for MySQL purchase page, complete the quick configuration, click **Purchase Now**, verify the bill, and complete the payment to activate.
   - Region: Select Beijing (this is just an example, you may choose a region closer to you).
   - Architecture: Select **Single-node**
   - Availability Zone: Select **Beijing Zone 3** , choose the availability zone where the subnet of the VPC is located.



基础配置

计费模式：包年包月（适用需求量长期稳定的业务） / 按量计费（适用需求量有大幅波动的场景）

详细对比 欠费说明

地域：中国 亚太 欧洲和美洲

广州 上海 南京 北京 成都 重庆 中国香港

处于不同地域的云产品内网不通，购买后不能更换，请您谨慎选择；例如，广州地域的云服务器无法通过内网访问上海地域的MySQL。若需要跨地域内网通信，请查阅：对等连接

数据库版本：MySQL5.6 MySQL5.7 MySQL8.0 MySQL5.7(TDSQL-C) 新 了解更多

推荐使用新一代云数据库TDSQL-C，100%兼容MySQL，秒级添加只读实例和原地升降配，快照备份回档，海量智能存储自动扩容，按使用量计费。

引擎：InnoDB RocksDB 了解更多

最常用的OLTP存储引擎，拥有完整的事务支持与强大的读写高并发能力

架构：双节点 三节点 单节点 新 了解更多

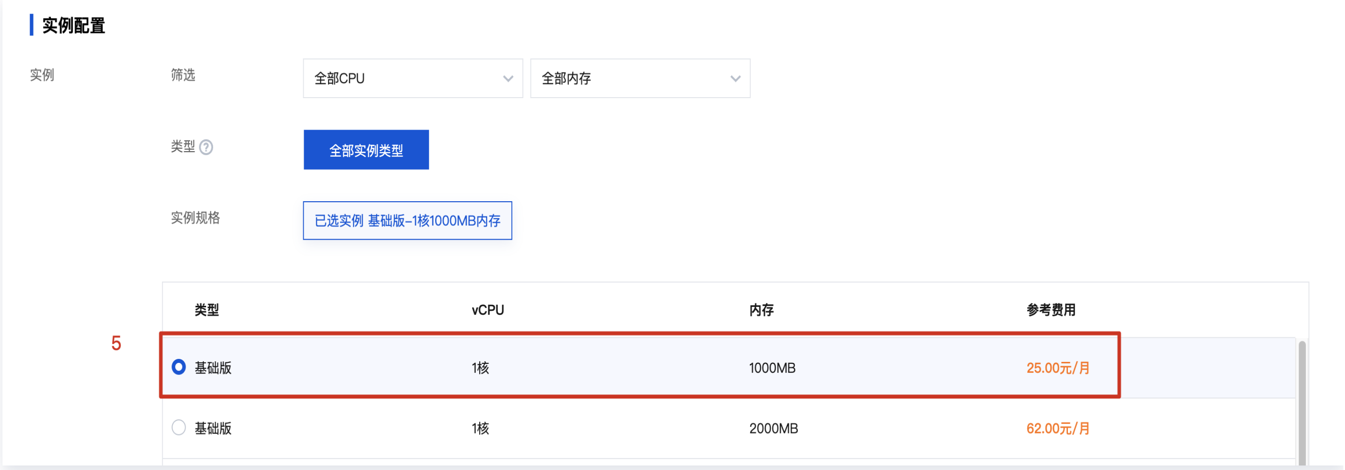
基础版不承诺 SLA，故障恢复时间较长，生产环境推荐使用双节点或三节点版本，提供最高99.99%可用性保障。

硬盘类型：云盘 了解硬盘类型

可用区：北京一区 北京二区 北京三区 北京四区 北京五区 北京六区 北京七区

处于同一私有网络下不同可用区的云产品内网互通；例如，相同私有网络下的广州二区的云服务器可以通过内网访问广州三区的MySQL

- Instance specifications: Select **Basic**



实例配置

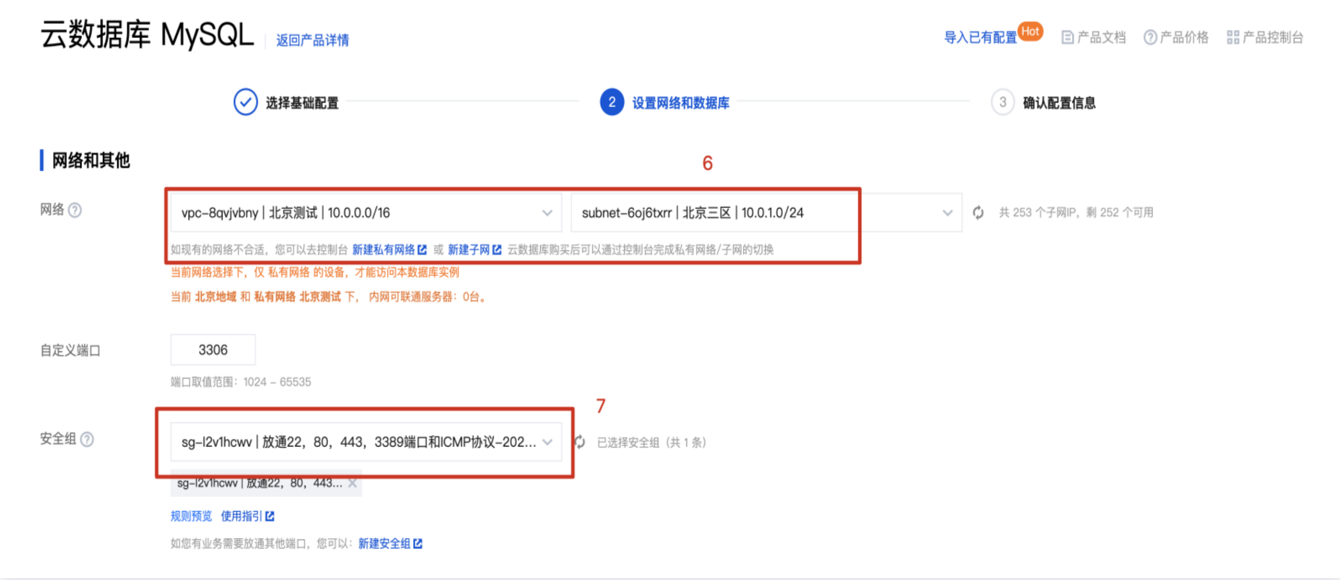
实例 筛选：全部CPU 全部内存

类型：全部实例类型

实例规格：已选实例 基础版-1核1000MB内存

类型	vCPU	内存	参考费用
● 基础版	1核	1000MB	25.00元/月
○ 基础版	1核	2000MB	62.00元/月

- Network: Select the newly created VPC mentioned above.
- Security Group: Select the default security group created earlier.



云数据库 MySQL 返回产品详情

导入已有配置 Hot 产品文档 产品价格 产品控制台

选择基础配置 — 2 设置网络和数据库 — 3 确认配置信息

网络和其他

6

网络：vpc-8qvjvbny | 北京测试 | 10.0.0.0/16　subnet-6oj6txrr | 北京三区 | 10.0.1.0/24　共 253 个子网IP，剩 252 个可用

如现有的网络不合适，您可以去控制台 新建私有网络 或 新建子网 云数据库购买后可以通过控制台完成私有网络/子网的切换

当前网络选择下，仅 私有网络 的设备，才能访问本数据库实例

当前 北京地域 和 私有网络 北京测试 下， 内网可联通服务器：0台。

自定义端口：3306

端口取值范围：1024 – 65535

7

安全组：sg-l2v1hcwv | 放通22，80，443，3389端口和ICMP协议-202...　已选择安全组（共 1 条）

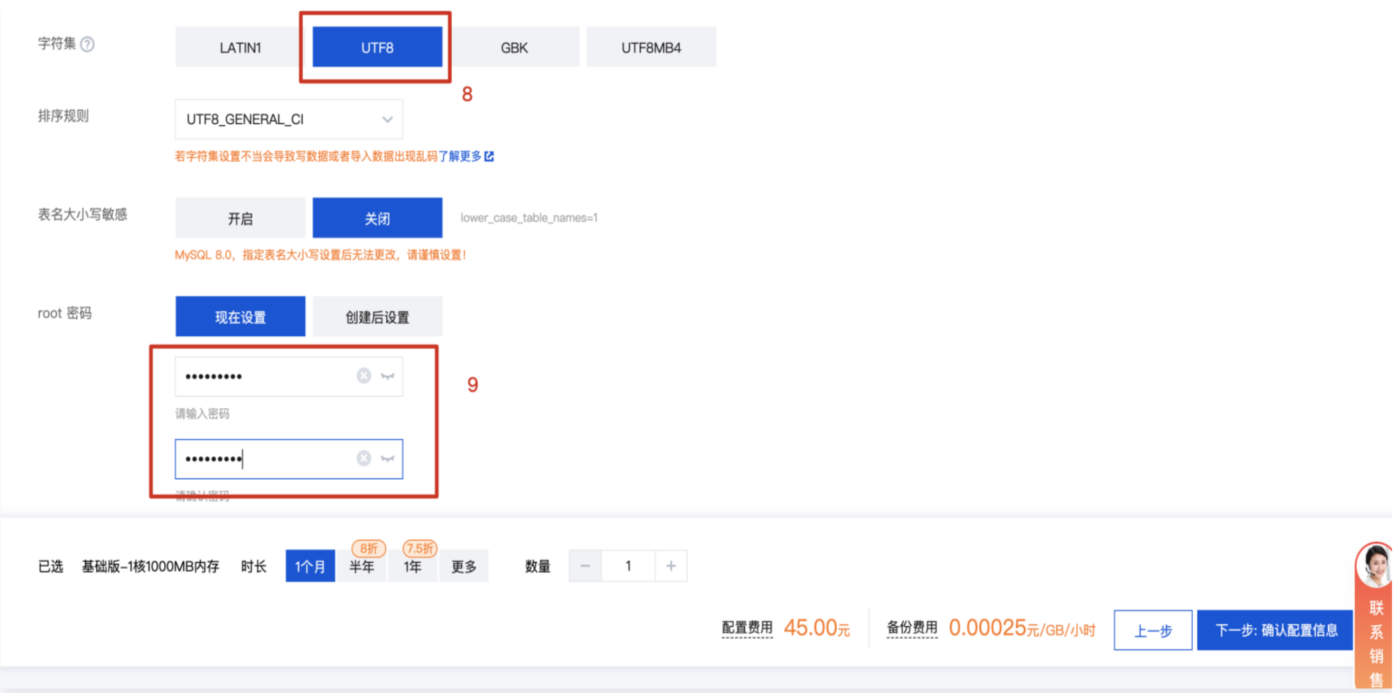
sg-l2v1hcwv | 放通22，80，443...

规则预览　使用指引

如您有业务需要放通其他端口，您可以：新建安全组

- Character Set: Select **UTF-8**
- Root Password: Set the root user's password.

2. After completing the settings, click **Next,** to confirm the configuration information.



字符集：LATIN1　UTF8　GBK　UTF8MB4

8

排序规则：UTF8_GENERAL_CI

若字符集设置不当会导致写数据或者导入数据出现乱码了解更多

表名大小写敏感：开启　关闭　lower_case_table_names=1

MySQL 8.0，指定表名大小写设置后无法更改，请谨慎设置！

root 密码：现在设置　创建后设置

••••••••••

请输入密码

••••••••••

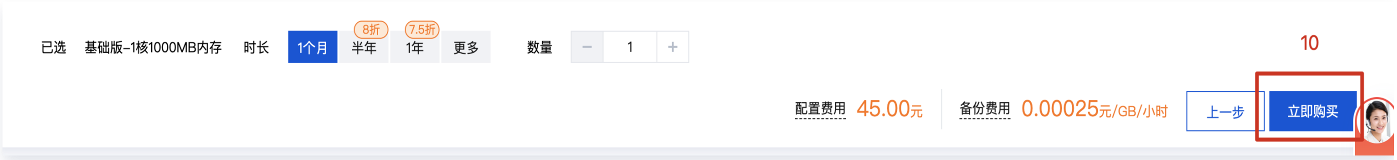
请确认密码

9

已选　基础版-1核1000MB内存　时长　1个月　8折 半年　7.5折 1年　更多　数量　1

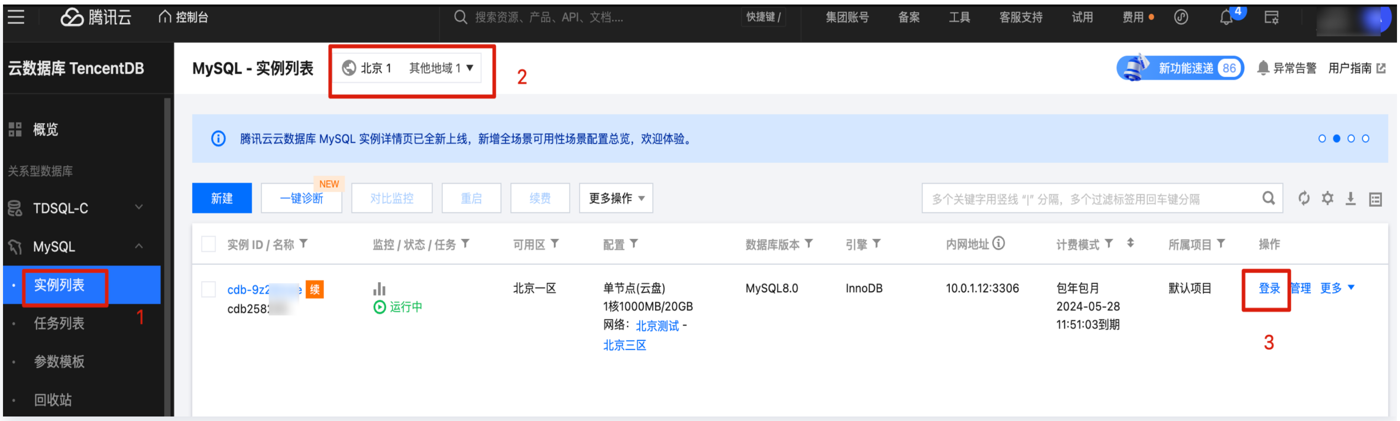
配置费用 45.00元　备份费用 0.00025元/GB/小时

3. Click **Buy Now** , verify the bill and complete the payment to activate.



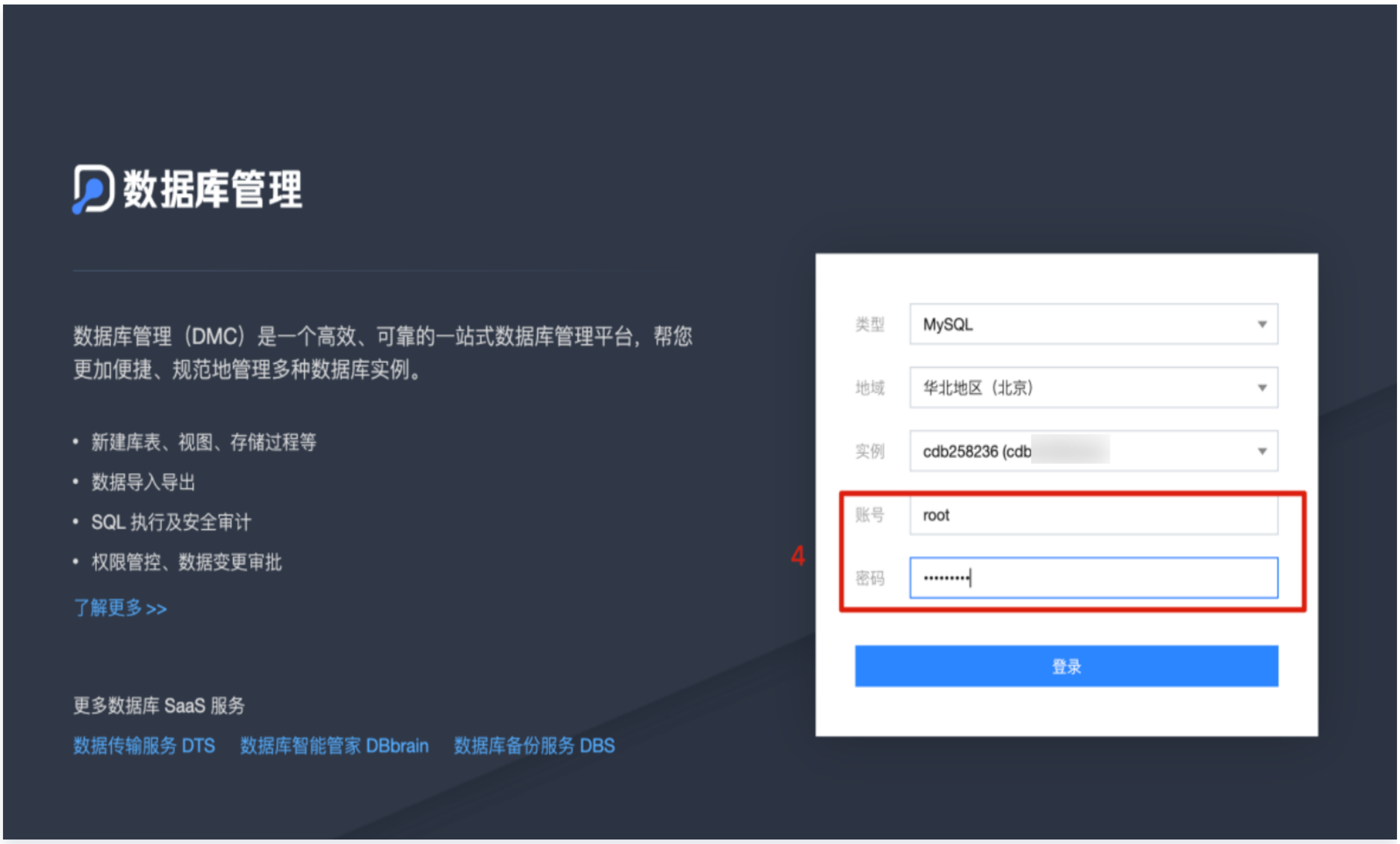
Initialize business data in MySQL

1. Enter the Tencent Cloud database MySQL Console , click the left-side menu **Instance List** to access the MySQL instance list page. Select the region at the top to **Beijing** , then click **Login to** .



实例 ID / 名称	监控 / 状态 / 任务	可用区	配置	数据库版本	引擎	内网地址	计费模式	所属项目	操作
cdb-9z2[redacted] 续 cdb258[redacted]	运行中	北京一区	单节点(云盘) 1核1000MB/20GB 网络：北京测试 - 北京三区	MySQL8.0	InnoDB	10.0.1.12:3306	包年包月 2024-05-28 11:51:03到期	默认项目	登录 管理 更多

2. Enter the DMC login interface, input your account and password, then click **Login to** .



类型 MySQL

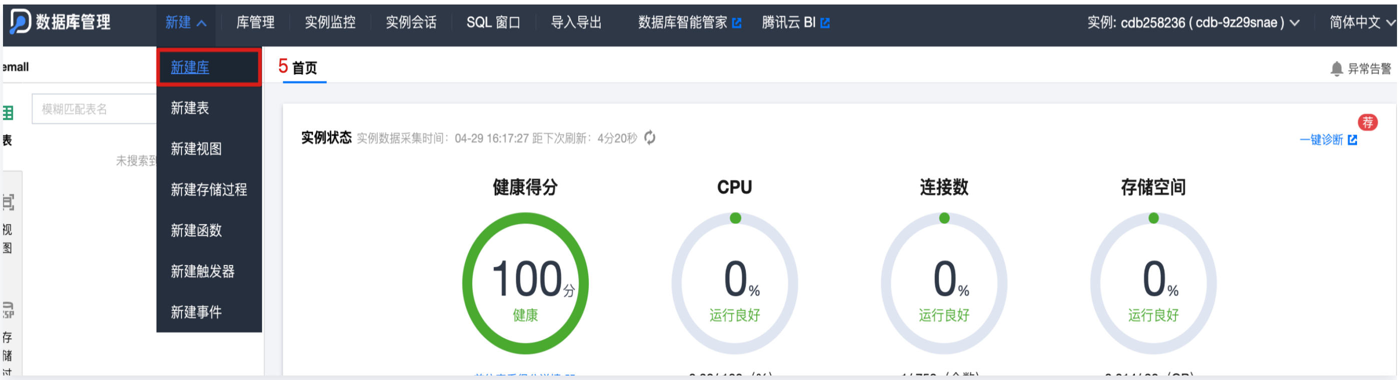
地域 华北地区（北京）

实例 cdb258236 (cdb[redacted]

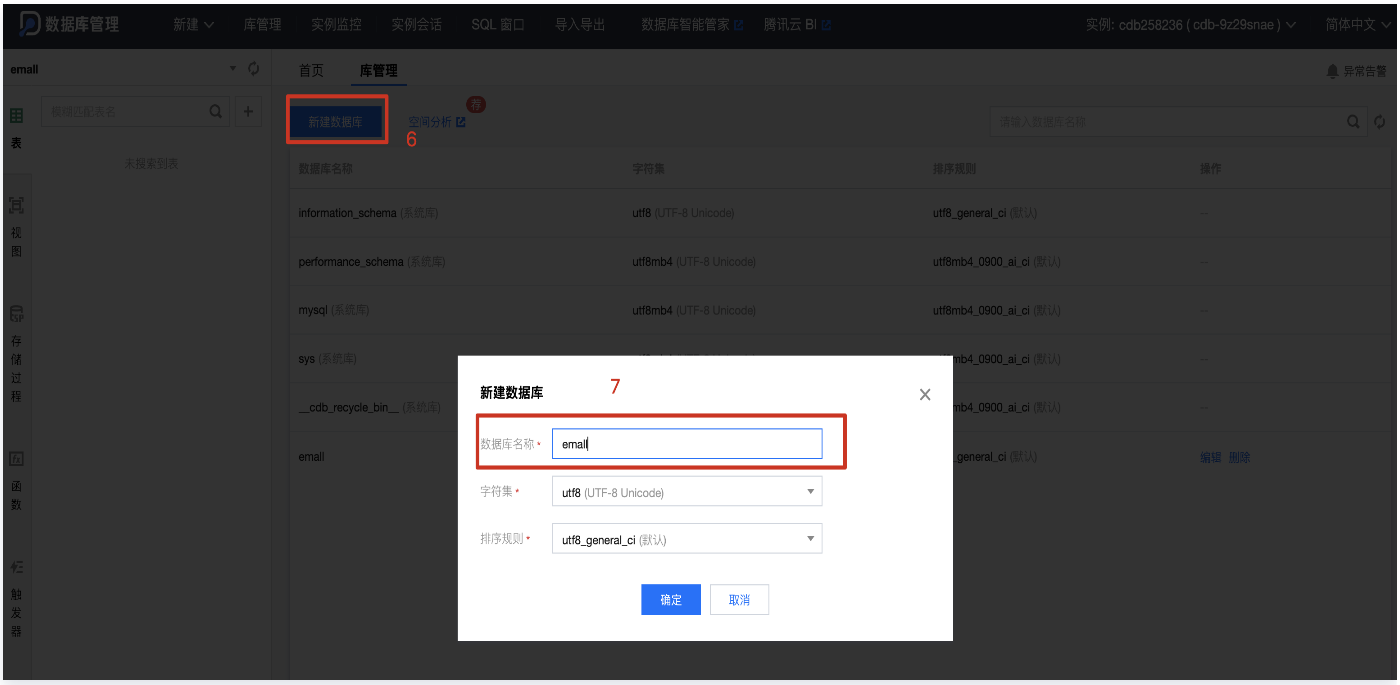
账号 root

密码 ••••••••

3. In the DMC interface, select the top menu **Create** > **Create Database** .



4. Enter the Create Database interface, click **Create New Database** to access the new database creation page, fill in the database name, suggested to be 'emall'. After completing, click **OK**.



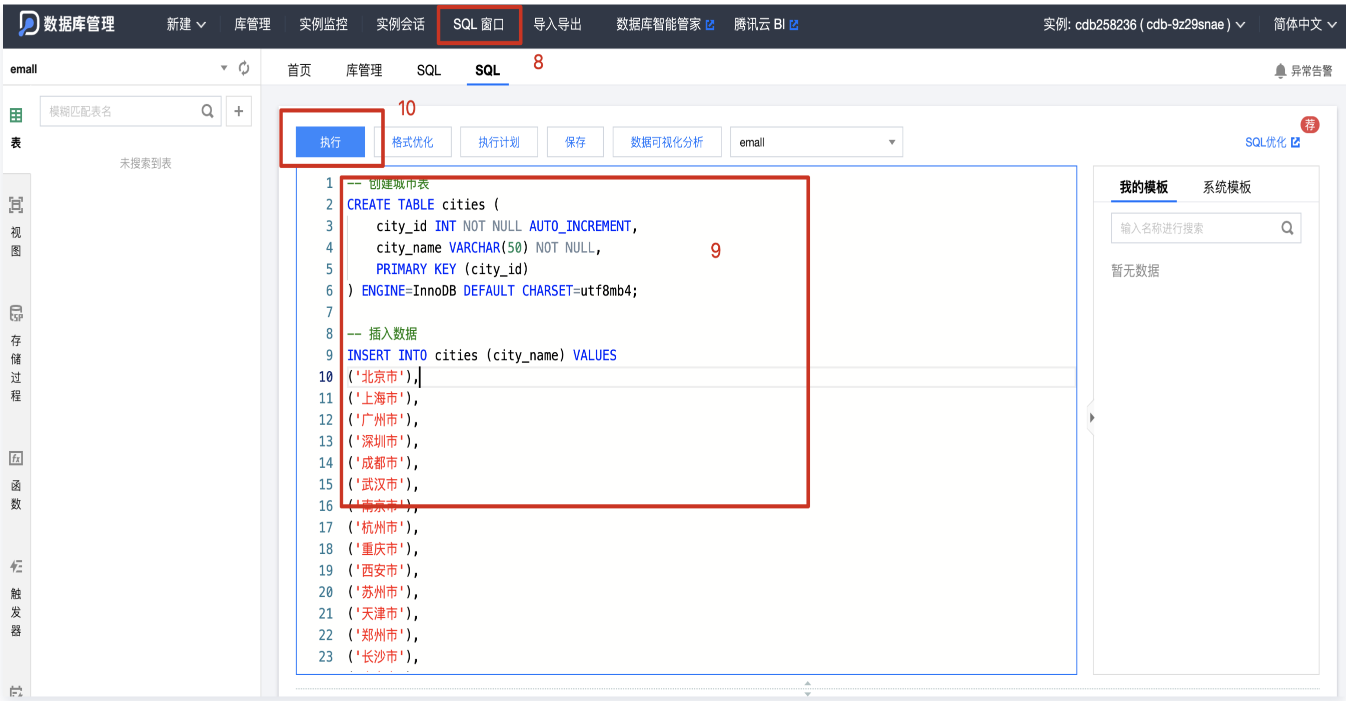
数据库名称	字符集	排序规则	操作
information_schema (系统库)	utf8 (UTF-8 Unicode)	utf8_general_ci (默认)	--
performance_schema (系统库)	utf8mb4 (UTF-8 Unicode)	utf8mb4_0900_ai_ci (默认)	--
mysql (系统库)	utf8mb4 (UTF-8 Unicode)	utf8mb4_0900_ai_ci (默认)	--
sys (系统库)		mb4_0900_ai_ci (默认)	--
__cdb_recycle_bin__ (系统库)		mb4_0900_ai_ci (默认)	--
emall		_general_ci (默认)	编辑 删除

数据库名称 * emall

字符集 * utf8 (UTF-8 Unicode)

排序规则 * utf8_general_ci (默认)

5. In the DMC interface, click the top menu **SQL Window** > **SQL** to enter the SQL interface. Quickly create tables by executing SQL statements.
   - Copy the following table creation SQL statements one by one. Each time you copy an SQL statement, click **Execute**. After execution, clear the SQL content before copying the next statement.



```
-- 创建城市表
CREATE TABLE cities (
    city_id INT NOT NULL AUTO_INCREMENT,
    city_name VARCHAR(50) NOT NULL,
    PRIMARY KEY (city_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- 插入数据
INSERT INTO cities (city_name) VALUES
('北京市'),
('上海市'),
('广州市'),
('深圳市'),
('成都市'),
('武汉市'),
('南京市'),
('杭州市'),
('重庆市'),
('西安市'),
('苏州市'),
('天津市'),
('郑州市'),
('长沙市'),
```

- The specific table creation statements are as follows:
  - **Create City Table (cities)**

```
-- Create cities table in MySQL
CREATE TABLE cities (
    city_id INT NOT NULL AUTO_INCREMENT,
    city_name VARCHAR(50) NOT NULL,
    PRIMARY KEY (city_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Insert data
INSERT INTO cities (city_name) VALUES
('Beijing 市'),
('Shanghai 市'),
('Guangzhou 市'),
('Shenzhen 市'),
('Chengdu 市'),
('武汉市'),
('Nanjing 市'),
('杭州市'),
('Chongqing 市'),
('西安市'),
('苏州市'),
('Tianjin 市'),
('郑州市'),
('长沙市'),
('青岛市'),
('沈阳市');
```

- **Create Product Category Table (categories)**

```sql
-- Create categories table
CREATE TABLE categories (
    category_id INT NOT NULL AUTO_INCREMENT,
    category_name VARCHAR(50) NOT NULL,
    PRIMARY KEY (category_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Insert data
INSERT INTO categories (category_name) VALUES
('电子产品'),
('家用电器'),
('服装鞋帽'),
('食品饮料'),
('图书音像'),
('运动户外'),
('家居建材'),
('母婴用品'),
('汽车用品');
```

- **Create Product Table (products)**

```sql
-- Create products table
CREATE TABLE products (
    product_id INT NOT NULL AUTO_INCREMENT,
    category_id INT NOT NULL,
    product_name VARCHAR(100) NOT NULL,
    PRIMARY KEY (product_id),
    FOREIGN KEY (category_id) REFERENCES categories(category_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Insert data
INSERT INTO products (category_id, product_name) VALUES
(1, '智能手机'),
(1, '笔记本电脑'),
(1, '平板电脑'),
(2, '空调'),
(2, '洗衣机'),
(3, '男士外套'),
(3, '女士裙子'),
(4, '碳酸饮料'),
(4, '矿泉水'),
(5, '现代小说'),
(5, '历史书籍'),
(6, '跑步鞋'),
(6, '瑜伽垫'),
(7, '实木家具'),
(7, '床上用品'),
(8, '婴儿奶粉'),
(8, '儿童玩具');
```

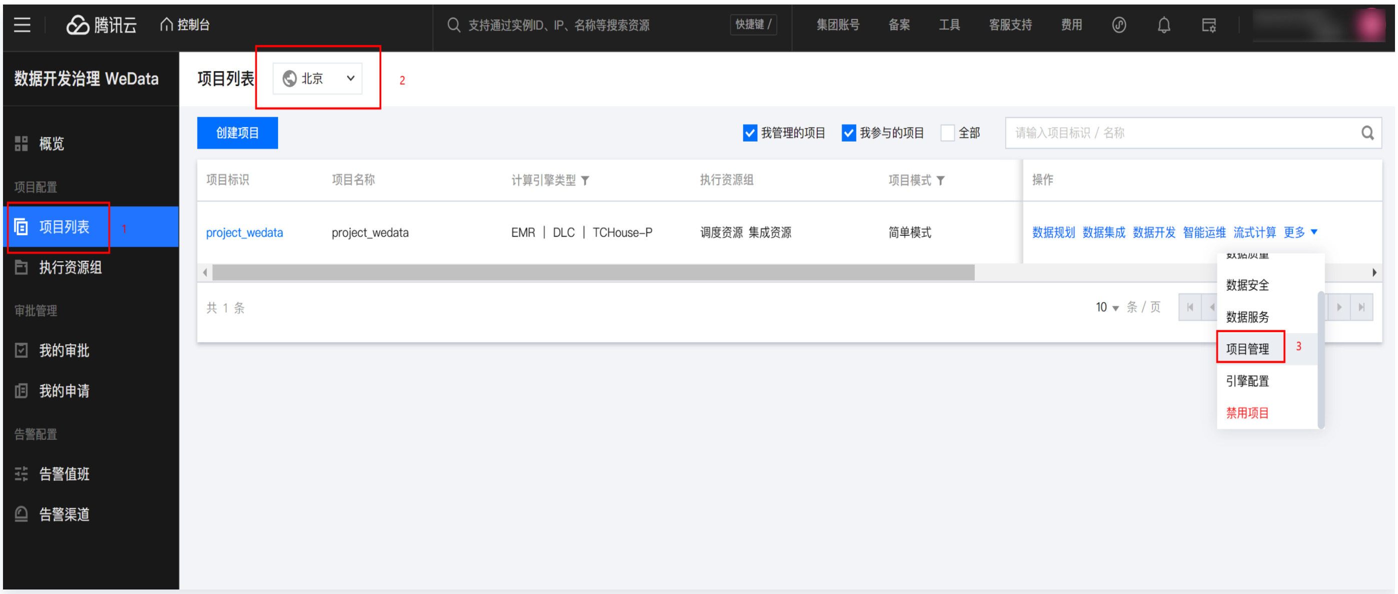
- Create Order Table (orders)

```
-- Create orders table
CREATE TABLE orders (
    order_id INT NOT NULL AUTO_INCREMENT,
    product_id INT NOT NULL,
    quantity INT NOT NULL CHECK (quantity > 0),
    unit_price DECIMAL(10, 2) NOT NULL,
    amount DECIMAL(10, 2) NOT NULL,
    order_time DATETIME NOT NULL,
    shipping_city_id INT NOT NULL,
    shipping_address TEXT NOT NULL,
    PRIMARY KEY (order_id),
    FOREIGN KEY (product_id) REFERENCES products(product_id),
    FOREIGN KEY (shipping_city_id) REFERENCES cities(city_id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- Inserting data
INSERT INTO orders (product_id, quantity, unit_price, amount,
order_time, shipping_city_id, shipping_address) VALUES
(1, 1, 4999.00, 4999.00, '2024-04-01 10:00:00', 1, 'Beijing Haidian
District'),
(2, 1, 6999.00, 6999.00, '2024-04-02 11:00:00', 2, 'Shanghai Pudong
New District'),
(3, 2, 3999.00, 7998.00, '2024-04-03 12:00:00', 3, 'Guangzhou Tianhe
District'),
(4, 1, 5999.00, 5999.00, '2024-04-04 13:00:00', 4, 'Shenzhen Nanshan
District'),
(5, 1, 999.00, 999.00, '2024-04-05 14:00:00', 5, 'Chengdu Wuhou
District'),
(6, 1, 699.00, 699.00, '2024-04-06 15:00:00', 6, 'Wuhan Jianghan
District'),
(7, 1, 2999.00, 2999.00, '2024-04-07 16:00:00', 7, 'Nanjing Gulou
District'),
(8, 1, 3999.00, 3999.00, '2024-04-08 17:00:00', 8, 'Hangzhou West Lake
District'),
(9, 1, 4999.00, 4999.00, '2024-04-09 18:00:00', 9, 'Chongqing Yuzhong
District'),
(10, 1, 1999.00, 1999.00, '2024-04-10 19:00:00', 10, 'Xi'an Beilin
District');
```

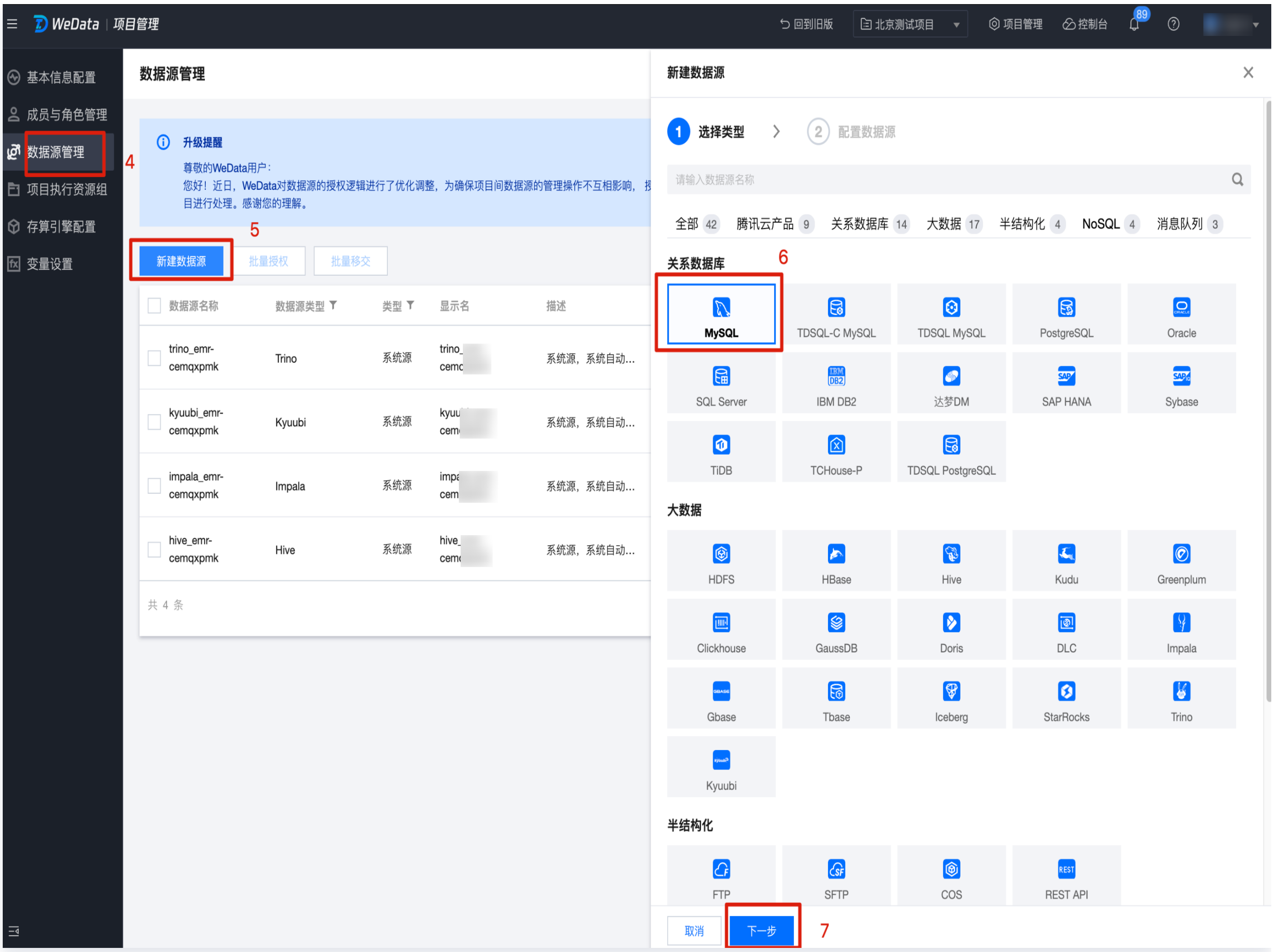
Bind MySQL in WeData

1. Log in to Tencent Cloud WeData Console , click on the left menu **Project List**, select the top region as Beijing, in the corresponding project operation column, click **Project Management.**



<table>
<tr><th>项目标识</th><th>项目名称</th><th>计算引擎类型</th><th>执行资源组</th><th>项目模式</th><th>操作</th></tr>
<tr><td>project_wedata</td><td>project_wedata</td><td>EMR | DLC | TCHouse–P</td><td>调度资源 集成资源</td><td>简单模式</td><td>数据规划 数据集成 数据开发 智能运维 流式计算 更多</td></tr>
</table>

2. In the Data Source Management interface, click **New Data Source** , select the data source type as MySQL, click **Next** .



数据源管理

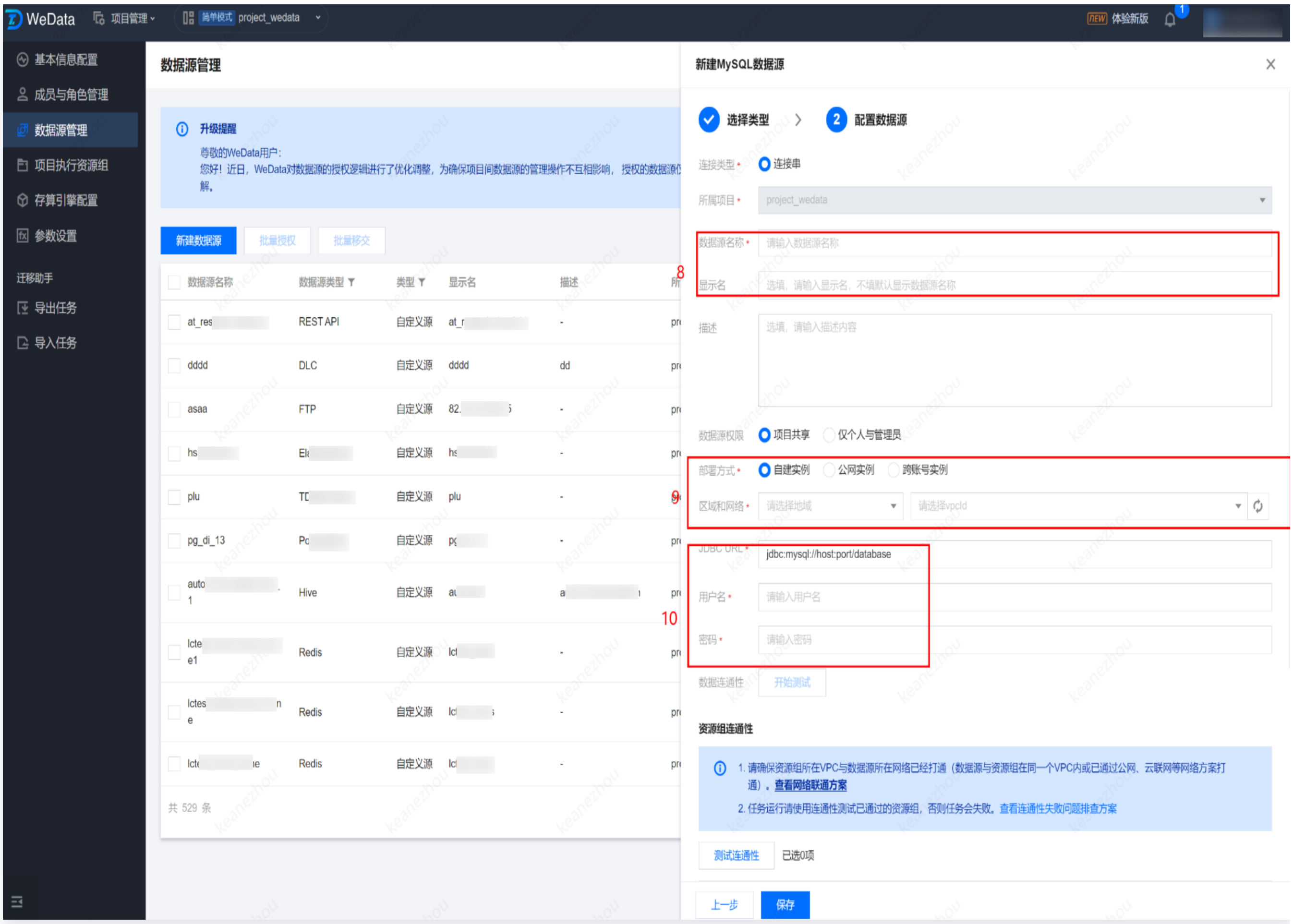
升级提醒

尊敬的WeData用户：
您好！近日，WeData对数据源的授权逻辑进行了优化调整，为确保项目间数据源的管理操作不互相影响，授
目进行处理。感谢您的理解。

数据源名称	数据源类型	类型	显示名	描述
trino_emr-cemqxpmk	Trino	系统源	trino_ cemc[redacted]	系统源，系统自动...
kyuubi_emr-cemqxpmk	Kyuubi	系统源	kyuu cem[redacted]	系统源，系统自动...
impala_emr-cemqxpmk	Impala	系统源	impa cem[redacted]	系统源，系统自动...
hive_emr-cemqxpmk	Hive	系统源	hive_ cem[redacted]	系统源，系统自动...

共 4 条

3. Enter the New MySQL Data Source page, fill in the relevant information, and click **Save** .
   - Data Source Name:
     - Data Source Name: Fill in as required for easy identification, for example: bj_mall.
     - Display Name: Fill in as required for easy identification, for example: Beijing—Test—Mall.
   - Instance Information:
     - Region: Select **Beijing.**
     - Select Instance: Select from the drop-down list.
   - Table Connection Information:
     - Database Name: emall
     - Username: root
     - Password: Enter the password set when purchasing MySQL mentioned above.



数据源管理

升级提醒

尊敬的WeData用户:
您好！近日，WeData对数据源的授权逻辑进行了优化调整，为确保项目间数据源的管理操作不互相影响，授权的数据源(
解。

新建数据源 批量授权 批量移交

□	数据源名称	数据源类型	类型	显示名	描述	所
□	at_res[redacted]	REST API	自定义源	at_r[redacted]	-	pr
□	dddd	DLC	自定义源	dddd	dd	pr
□	asaa	FTP	自定义源	82.[redacted]5	-	pr
□	hs[redacted]	El[redacted]	自定义源	hs[redacted]	-	pr
□	plu	TD[redacted]	自定义源	plu	-	pr
□	pg_di_13	Po[redacted]	自定义源	pg[redacted]	-	pr
□	auto[redacted]. 1	Hive	自定义源	au[redacted]	a[redacted]n	pr
□	lcte[redacted] e1	Redis	自定义源	lc[redacted]	-	pr
□	lctes[redacted]n e	Redis	自定义源	lc[redacted]s	-	pr
□	lct[redacted]ie	Redis	自定义源	lc[redacted]	-	pr

共 529 条

新建MySQL数据源

✓ 选择类型 > 2 配置数据源

连接类型 * ● 连接串

所属项目 * project_wedata

数据源名称 * 请输入数据源名称

显示名 选填，请输入显示名，不填默认显示数据源名称

描述 选填，请输入描述内容

数据源权限 ● 项目共享 ○ 仅个人与管理员

部署方式 * ● 自建实例 ○ 公网实例 ○ 跨账号实例

区域和网络 * 请选择地域 请选择vpcId

JDBC URL * jdbc:mysql://host:port/database

用户名 * 请输入用户名

密码 * 请输入密码

数据连通性 开始测试

资源组连通性

1. 请确保资源组所在VPC与数据源所在网络已经打通（数据源与资源组在同一个VPC内或已通过公网、云联网等网络方案打通）。查看网络联通方案
2. 任务运行请使用连通性测试已通过的资源组，否则任务会失败。查看连通性失败问题排查方案

测试连通性 已选0项

上一步 保存

All preparations are now complete. We will now officially start the data synchronization and data development sections of this tutorial. Subsequent operations can be performed using a Tencent Cloud Sub-account.

Data Table Structure Design

Last updated：2025-04-18 15:46:32

Business Research

Source Data Storage Location

Through researching the existing technical architecture of the mall system, it was found that the data is stored in a MySQL database.
Here, it is assumed that the TencentDB for MySQL database is used.

Target Business Scenario Analysis

By analyzing the target business scenarios: sales performance in various cities and categories, we need to obtain the following tables:

- Order Table: (At this point, ignore the design of subtables such as order details, assuming the order table includes product ID, product quantity, product price, shipping address, order time, etc.).
- Product Table: (At this point, ignore the design of subtables such as SKU, assuming the product table includes product ID, product category, etc.).
- City Table: (Assuming the geographic location coding table only goes down to the city level, the city table includes: city code, city name).
- Product Category Table: (Assuming there is only one level of category, the category table includes: category code, category name).

Actual Structure

The following are the actual structures of the researched Order Table and Product Table:

1. Order Table (orders)

Field name	Field Type	Field Length	Field Description	Sample Code
order_id	INT	10	Order ID, primary key, auto-increment	10001
product_id	INT	10	Product ID, foreign key	1001
quantity	INT	5	Quantity of Goods, positive integer	2
unit_price	DECIMAL(10,2)	-	Unit Price of Goods, round to two decimal places	99.99
amount	DECIMAL(10,2)	-	Subtotal Amount of Goods, which is quantity multiplied by the unit price	199.98

order_time	DATETIME	-	Order Time, accurate to the minute	"2024-04-04 10:30:00"
shipping_city_id	INT	10	Shipping Address City ID, foreign key	1101
shipping_address	TEXT	-	Shipping Address, includes province, city, district, and detailed address	"Chaoyang District, Beijing, XXX Community"

2. Product Table (products)

Field Code	Field Type	Field Length	Field Description	Sample Code
product_id	INT	10	Product ID,primary key,Auto-increment	1001
category_id	INT	10	Category ID,Foreign key	101
product_name	VARCHAR(100)	-	Product Name	"Smartphone"

3. City Table (cities)

Field Code	Field Type	Field Length	Field Description	Sample Code
city_id	INT	10	City Code,primary key,Auto-increment	1101
city_name	VARCHAR(50)	-	City Name	"Beijing City"

Product Category Table (categories)

Field Code	Field Type	Field Length	Field Description	Sample Code
category_id	INT	10	Category ID,primary key,Auto-increment	1
category_name	VARCHAR(50)	-	Category Name	"Electronic Products"

Architecture Design

Based on business scenario needs, the final business output involves Data Warehouse Layering and Data Table Structure.

Model Specification

Model specifications help teams unify data warehouse design rules, streamline the data development process, better accumulate data assets, and lay a foundation for building data services and data marts. During the design of data warehouse model specifications, multiple categories are included, such as data domain and principal domain.

In this scenario, the core objective is DataInLong and the data development process, and therefore detailed teaching of data model specifications is not covered in this tutorial.

Below are examples of model specifications related to this scenario:

Category	Chinese Description	English Name
Business Category	Sales	trade
Data Domain	Order Product	order product
Business process	Order Creation	ordercreate
Subject Domain	Product	product
Dimension	Date Region Category	date city category
Metrics	Sales Volume Sales Quantity	amount quantity

Data Warehouse Layering

1. Data Ingestion Layer ODS

Import raw data that hasn't undergone any processing into the data warehouse. The table structure in the ODS layer is consistent with the table structure in the original data system.

Therefore, we need to create 4 Hive tables based on the raw data (table creation operations are not needed here; this will be covered in subsequent lessons), with table structures identical to the MySQL source data tables.

The naming of the four tables is as follows:

- Order Table: ods_order_order
- Product Table: ods_product_product
- Category Table: ods_product_category
- City Table: ods_order_city

> **Note**
> The suggested naming convention is: ods_{data domain}_{self-definition content}.

2. Common Dimension Layer DIM

This section focuses on data synchronization logic, temporarily ignoring the design of the dimension layer.

Note
It is recommended to redundantly store the field attributes from the dimension tables in the detailed data tables.

3. Detailed Data Layer DWD

Build the most granular detailed data table. It is advisable to redundantly store some fields in this table to reduce the association between the detailed data table and the dimension table.

- Build a detailed table: Table creation is not required here; it will be covered in subsequent lessons.
- Product Sales Detail Report: dwd_trade_order_ordercreate_productsales.

Note
The suggested naming convention is: dwd_{business category}_{data domain}_{business process}_{self-definition content}.

Field Code	Field Type	Field Length	Field Description	Sample Code
order_id	INT	10	Order ID, primary key	10001
product_id	INT	10	Product ID	1001
category_id	INT	10	Category ID	101
category_name	STRING	50	Category Name	"Electronic Products"
product_name	STRING	50	Product Name	"Smartphone"
quantity	INT	5	Quantity of Goods, positive integer	2
unit_price	DECIMAL	10,2	Unit Price of Goods, round to two decimal places	99.99
amount	DECIMAL	10,2	Subtotal Amount of Goods, which is quantity multiplied by the unit price	199.98
order_time	DATETIME	-	Order Time, accurate to the minute	"2024-04-04 10:30:00"
shipping_city_id	INT	10	Shipping Address City ID, foreign key	1101
shipping_city_	STRIN	50	City Name	"Beijing City"

name	G			
shipping_address	TEXT	–	Shipping Address, includes province, city, district, and detailed address	"Chaoyang District, Beijing, XXX Community"
pt_date	STRING	50	Partition Field	"2024–04–01"

Additional Notes: Hive Table Partitioning

Partitioned Table Overview

Partitioning is an essential database optimization technique that improves performance, simplifies management, reduces costs, and enhances data availability and security by dividing datasets into smaller, logically independent parts.

Partitioning is especially important in big data scenarios.

Benefits of Hive Table Partitioning

The benefits of partitioning storage in Hive can be reflected in several aspects:

Advantage	Description
Improve Query Performance	By storing data in different partitions, queries can target specific partitions, avoiding the need to scan the entire table's data and significantly reducing query time.
Optimize Data Management	Partitioning is a logical way of organizing data, making it easier to maintain, clean, and perform bulk operations such as backup and recovery.
Horizontal Scaling	Partitions can horizontally distribute data storage pressure, physically distributing data to different storage units, enhancing system scalability.
Reduce Data Skew	In cases of uneven data distribution, partitioning can prevent data skew by avoiding situations where some partitions have too much data while others have too little.
Data Isolation	Partitions can be used for data isolation. For example, data can be divided into different partitions based on time, facilitating version control and historical data management.
Reduce Data Loading Time	During data loading or ETL processes, data can be loaded into specific partitions more quickly without operating on the entire table.
Save Storage Space	Partitions can help delete or archive old partition data, thereby saving storage space.
Parallel Processing	Partition tables can better leverage Hadoop's MapReduce parallel processing capabilities, as queries can be executed in parallel on different partitions.

Data Security and Access Control	Partitions can be used to implement finer-grained data access control, such as setting stricter access permissions for certain partitions.
Maintain Data Integrity	Partitions can ensure data integrity because each partition can have its own data integrity constraints.
Support Hot and Cold Data Layering	By partitioning, data can be classified into "hot data" and "cold data" based on its frequency of use, and different storage strategies can be applied.
Simplify Data ETL Process	During data extraction, transformation, and loading, partitions can simplify data organization and processing workflows.
Improve Data Availability	Partitions can improve data availability, as the unavailability of one partition does not affect access to other partitions.

Therefore, plan partition fields as early as possible when creating Hive tables.

4. Summary Data Layer DWS

Construct summary indicator data tables with business-use granularity.

- Build a summary table: Table creation is not required here; it will be covered in subsequent lessons.
- Daily Product Sales Summary Table: dws_trade_order_productsales_1d.
- The suggested naming convention is: dws_{business category}_{data domain}_{self-definition content}_{time period}.

Field Code	Field Type	Field Length	Field Description	Sample Code
order_date	DATE	-	Date	2021-04-01
city_id	INT	10	City ID	1
category_id	INT	10	Category ID	1
city_name	STRING	50	City Name	" Beijing "
category_name	STRING	50	Category Name	"Electronic Products"
quantity	INT	10	Total Product Sales	100
amount	DECIMAL	(10, 2)	Total Product Sales Volume	9999.99
pt_date	STRING	50	Partition Field	"2021-04-01"

5. Application Data Layer (ADS)

Build an indicator table for final business analysis requirements. Since this scenario is relatively simple, it will be temporarily ignored here.

Data Integration

Last updated：2025-04-18 15:48:35

We will sync the raw data to the data warehouse in this step.

Add a New Data Source

Raw Data Source: MySQL

We have bound the data source to the project. It can be ignored here.

Target Data Source: Hive

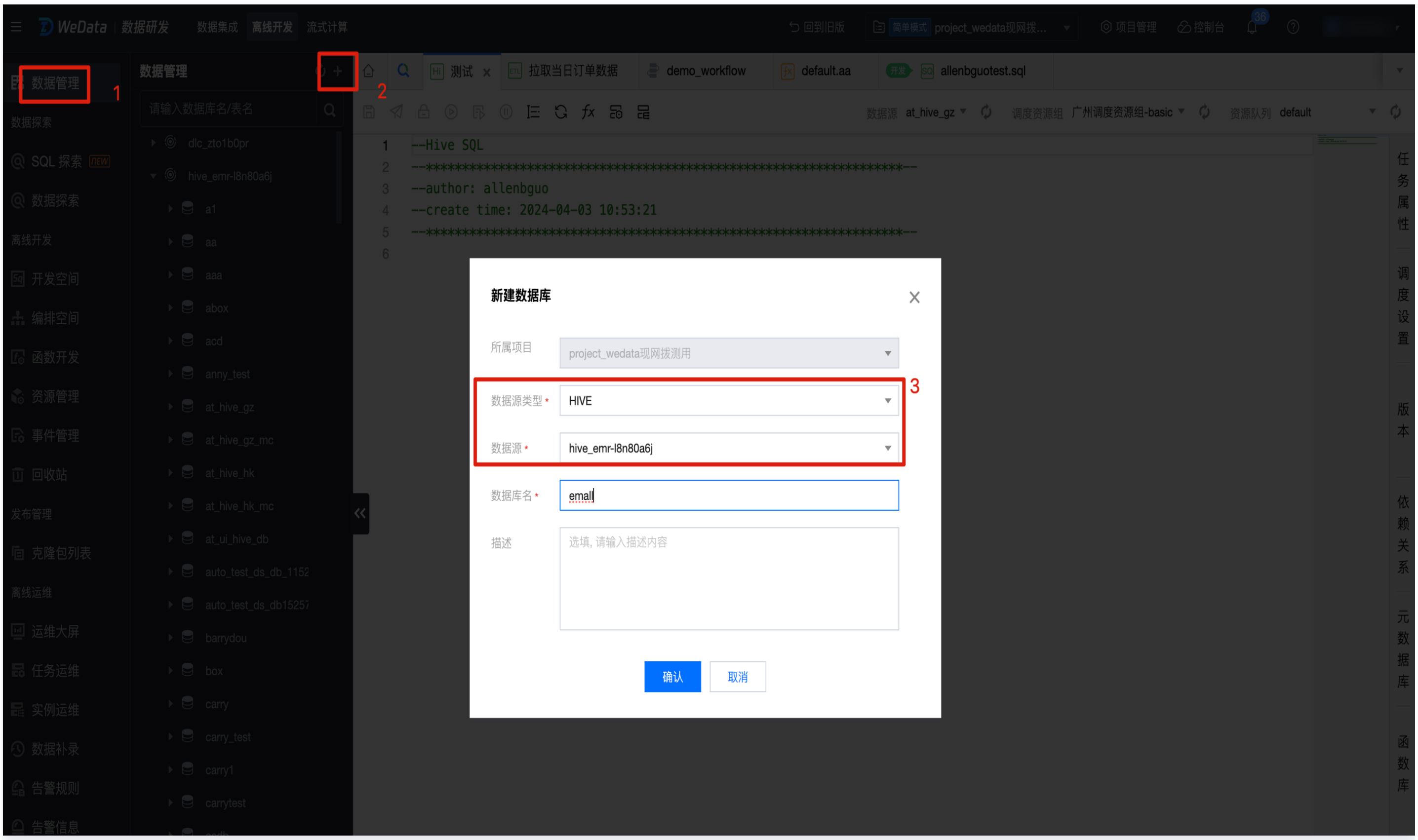
After you bind the storage-compute engine EMR, the system will collect the Hive data source in the EMR cluster within 10 minutes. Therefore, it is not necessary for you to actively bind the Hive data source.
However, we need to **create a database** in the Hive data source to store the collected original data.

Creating a Database

Enter **Data Development** > **Data Management**, click **+** to create a database, select and fill in the required content, and after completing, click **OK**.

- Select data source type: Hive.
- Data source selection: hive_emr-XXX.

> **Note**
> After you bind the storage-compute engine EMR, the system will collect the Hive data source in the EMR cluster within 10 minutes.



```
--Hive SQL
--**********************************************************************--
--author: allenbguo
--create time: 2024-04-03 10:53:21
--**********************************************************************--
```

新建数据库

所属项目: project_wedata现网拨测用

数据源类型*: HIVE

数据源*: hive_emr-l8n80a6j

数据库名*: emall

描述: 选填, 请输入描述内容

确认 取消

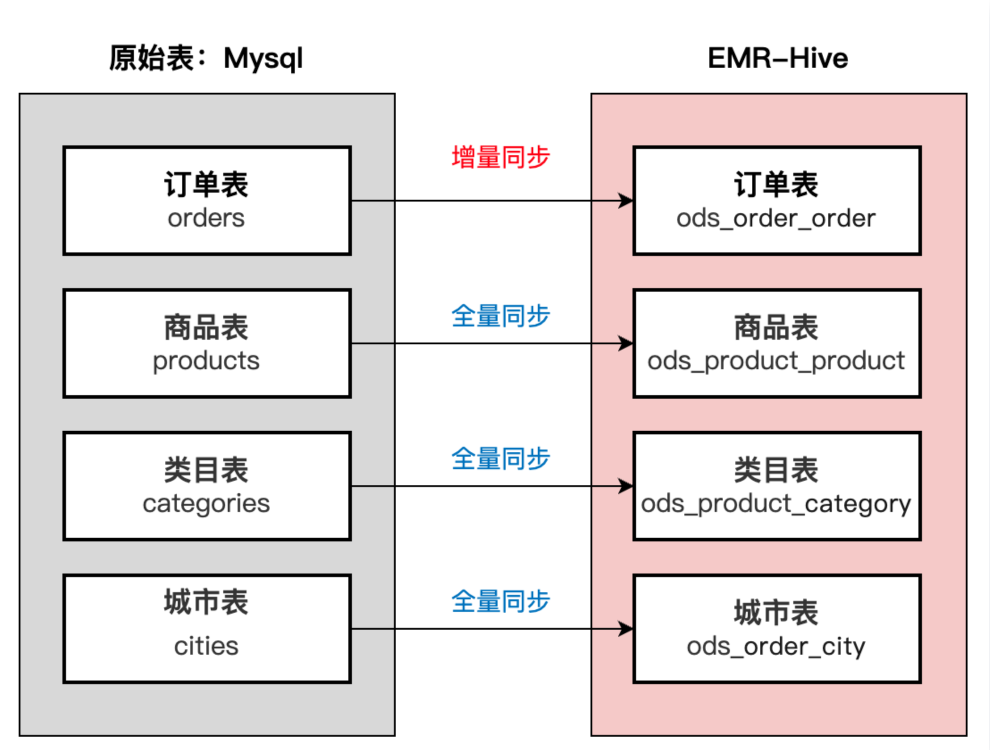
Offline Synchronization Task Design

We will now create an offline synchronization task to synchronize the original data in the MySQL data source to the Hive table in the EMR cluster.

We have already known from the above operations that we need to synchronize 4 raw data tables, which are:

No.	Table Name	Raw Data Source: MySQL	<Target Data Source: Hive Table Name>
1	Order Table	orders	ods_order_order
2	Product Table	products	ods_product_product
3	Category Table	categories	ods_product_category
4	City Table	cities	ods_order_city

The task development plan is designed as follows:



Additional Notes

Difference between full synchronization and incremental synchronization:

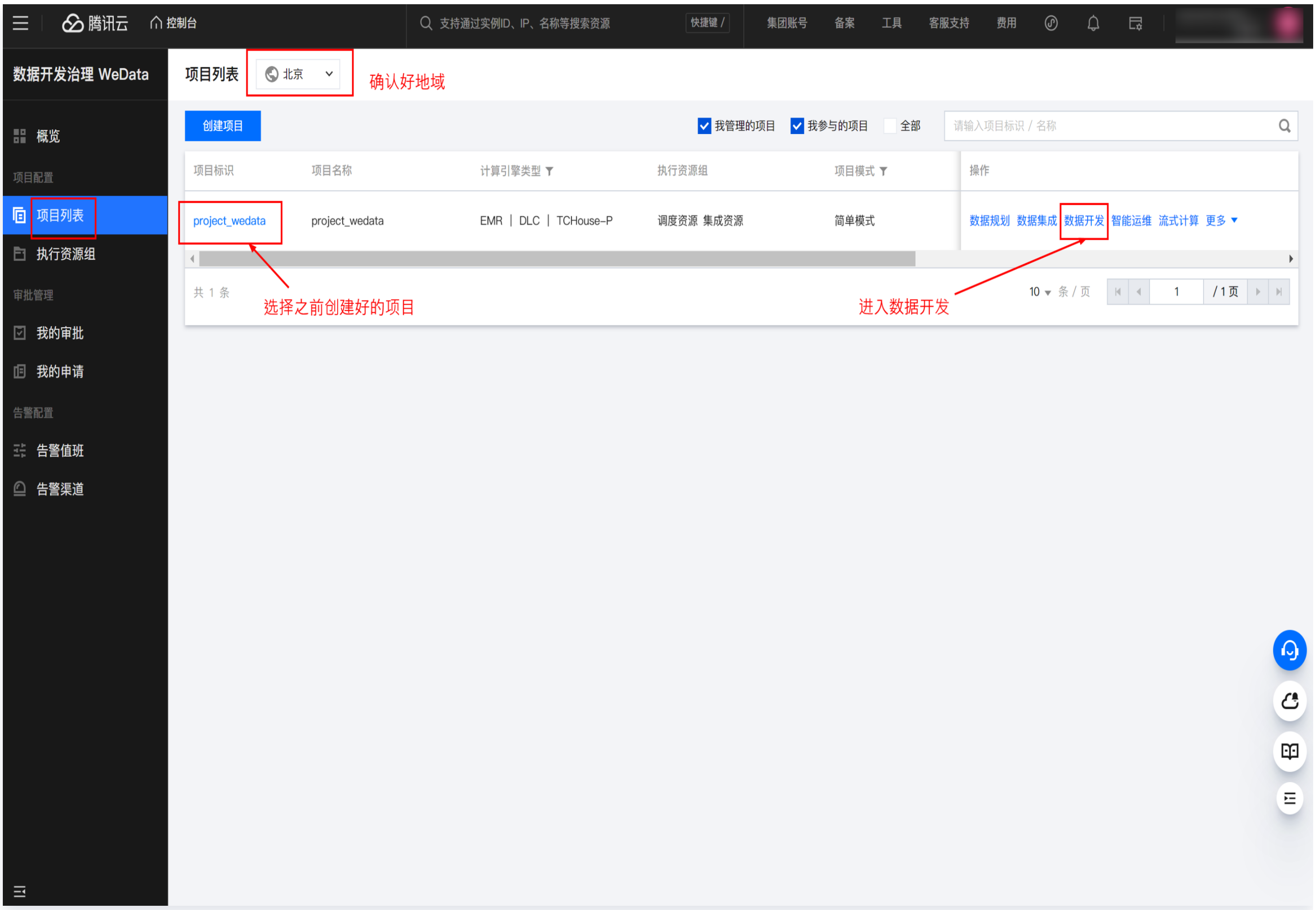
Name	Description
Full synchronization	**Definition:** Full synchronization refers to the process where the system transmits all data from two databases or data warehouses during each sync operation.
	Use cases: Full synchronization is usually suitable for small amounts of data or infrequent data change situations, as well as during initial sync or data migration.
	Strengths: • Simple and easy to implement: No need to track data changes, just directly copy all data. • Integrity: Underwrite the integrity and consistency of data, because all data is resynchronized.
	Drawbacks: • High time and resource consumption: A large amount of data needs to be transmitted, which takes a long time and occupies bandwidth. • High cost: For large data volume, more storage and computing resources may be needed.
Incremental synchronization	**Definition:** Incremental synchronization only synchronizes the data that has changed since the last synchronization, rather than all data.

	Use cases: Suitable for environments with large volumes of data or frequent updates.
	Strengths: - High efficiency: Synchronize only the changed data, saving time and bandwidth. - Low cost: Reduces the need for storage and computing resources. - Real-timeness: Can reflect the latest state of data more quickly.
	Drawbacks: - High complexity: A mechanism is required to track and record data changes. - Consistency issues may exist: If problems occur during the sync process, data inconsistency may result.
Summary: The choice of sync method depends on the specific application scenario and requirements. For small amounts of data with infrequent changes, full synchronization may be simpler and more efficient. Conversely, for large amounts of data with frequent updates, incremental synchronization can significantly improve efficiency and reduce costs. In practical applications, these two strategies are sometimes used in conjunction, for example, regularly conducting full synchronization to ensure data integrity, while using incremental synchronization in daily operations to improve efficiency.	

Offline Synchronization Task Development

Creating Workflow

1. Enter the offline development page from the project list.



项目列表 北京

确认好地域

☑ 我管理的项目 ☑ 我参与的项目 □ 全部

请输入项目标识 / 名称

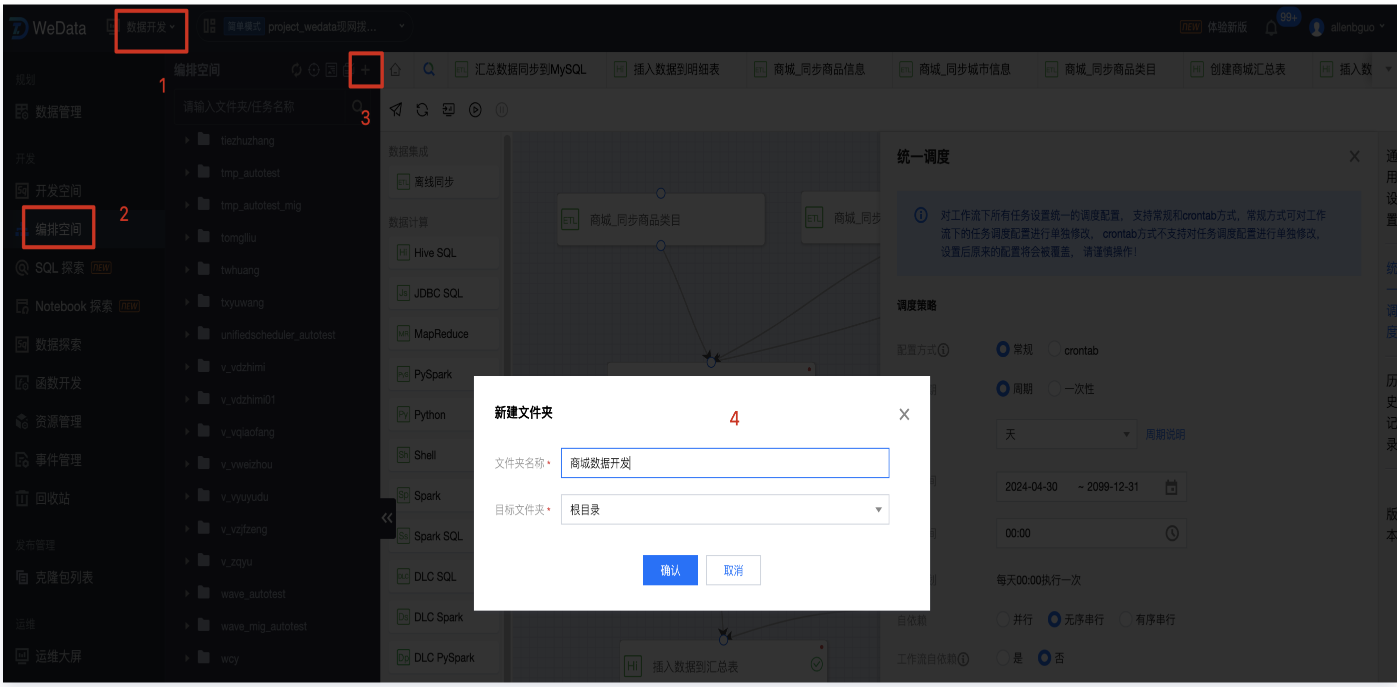
项目标识	项目名称	计算引擎类型	执行资源组	项目模式	操作
project_wedata	project_wedata	EMR | DLC | TCHouse-P	调度资源 集成资源	简单模式	数据规划 数据集成 数据开发 智能运维 流式计算 更多

共 1 条 10 条 / 页 1 / 1 页

选择之前创建好的项目

进入数据开发

2. Select offline **Data Development** > **Orchestration Space**, click **+** to create a new folder (named: Mall Data Task Development), and store subsequent development tasks.



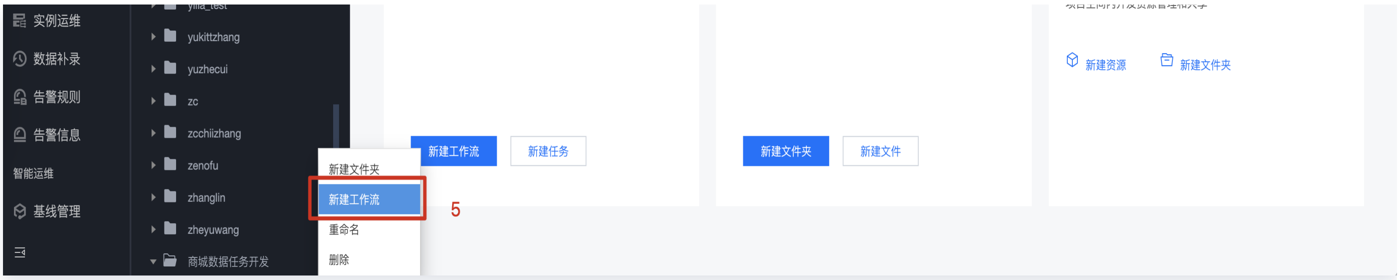
新建文件夹

文件夹名称 * 商城数据开发

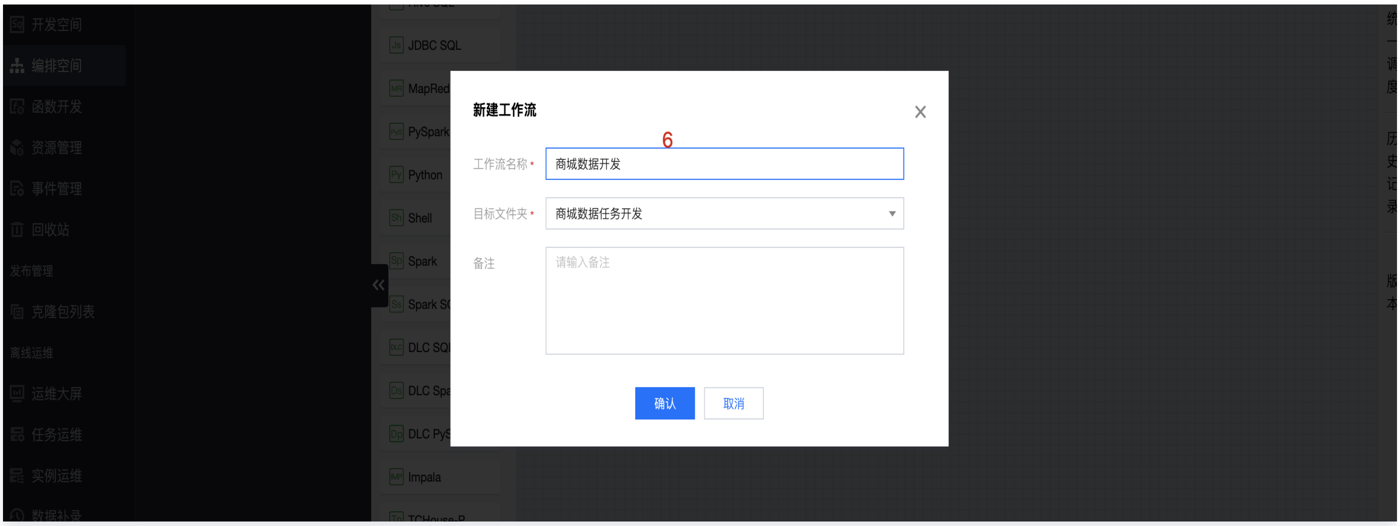
目标文件夹 * 根目录

确认 取消

3. Find the **created folder** > Right-click **create workflow.**



4. Create a new workflow named Mall Data Development, select the corresponding target folder, and click **Confirm.**



新建工作流

工作流名称 * 商城数据开发

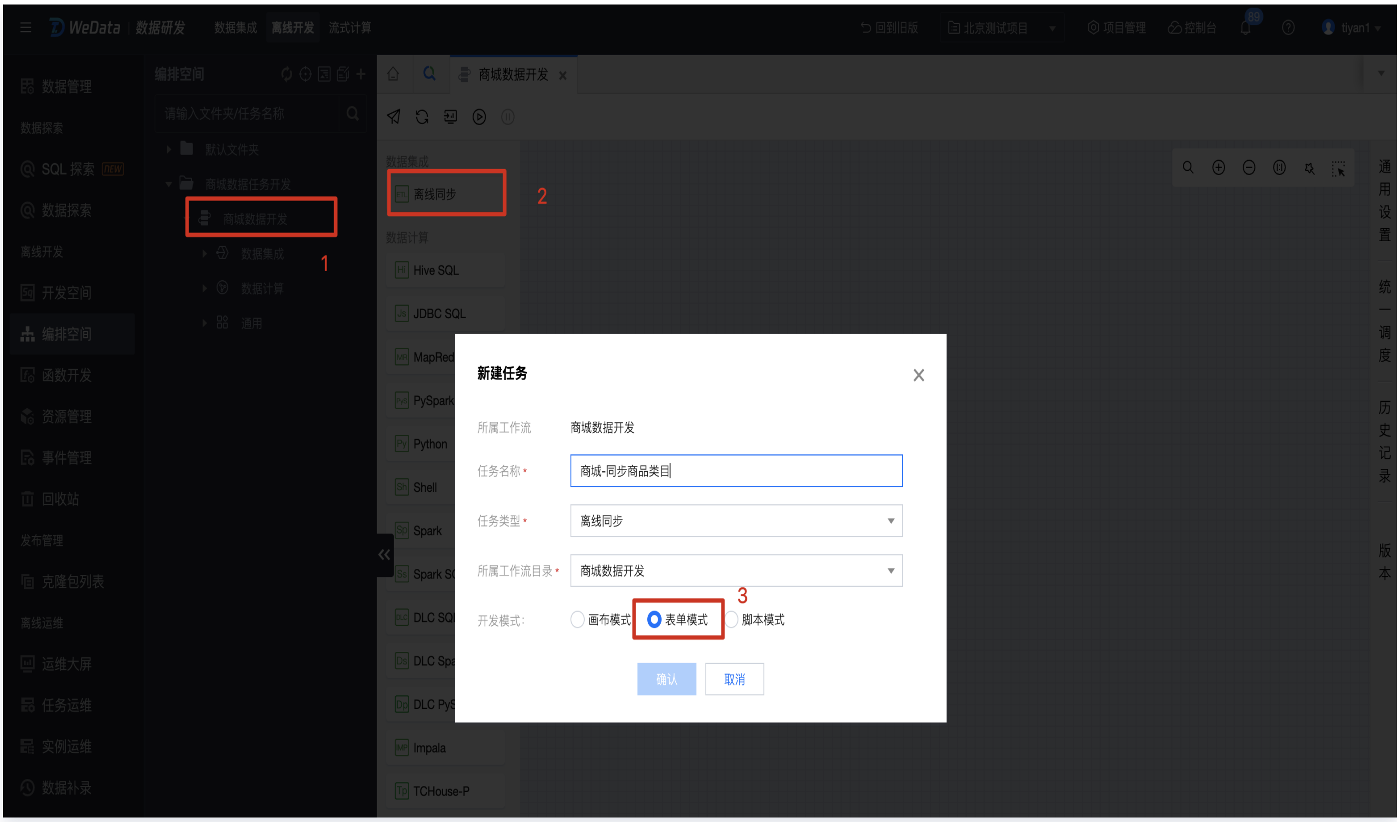
目标文件夹 * 商城数据任务开发

备注 请输入备注

Synchronize Category Table

First, we will synchronize the category table from MySQL to the Hive table.

1. Find the newly created **Mall Data Task Development** > **Mall Data Development** in the Orchestration Space. Click **Offline Sync.** Select the configuration mode (Task Name: Mall_Synchronize Product Category, Development Mode: Select **Form Mode** here), and click **Confirm.**



新建任务

所属工作流 商城数据开发

任务名称 * 商城-同步商品类目

任务类型 * 离线同步

所属工作流目录 * 商城数据开发

开发模式: ○ 画布模式 ● 表单模式 ○ 脚本模式

确认 取消

2. Configure the raw data source.

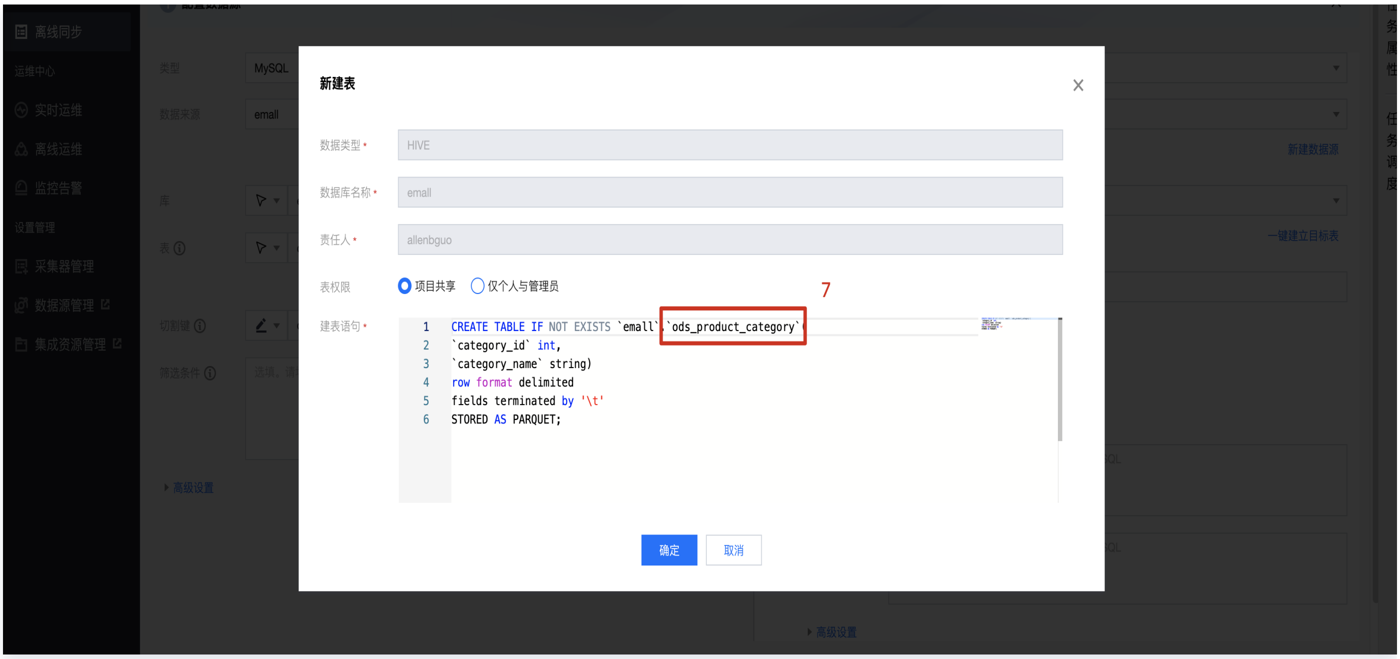
   2.1 Select the database and table where the raw data table is stored. Please select the MySQL data source added in the previous step. In the following text, when we introduce order table sync, we will introduce how to achieve incremental data synchronization by setting filter conditions.

   > **Notes:**
   > Since the category table is basic information and the amount of data is generally small, we do not need to set filter conditions (i.e., Where statements) here.

   2.2 Configure the target data source. Select the Hive table where you need to store data. Please select the Hive data source and database added in the previous step.

   - Type: Hive
   - Data destination: search hive_emr
   - Database: emall (the database created in the previous step)

   2.3 Establish a target table. Here, use **create target tables with one click** to replicate the MySQL Table Structure.



新建表

数据类型* HIVE

数据库名称* email

责任人* allenbguo

表权限 ● 项目共享 ○ 仅个人与管理员

建表语句*

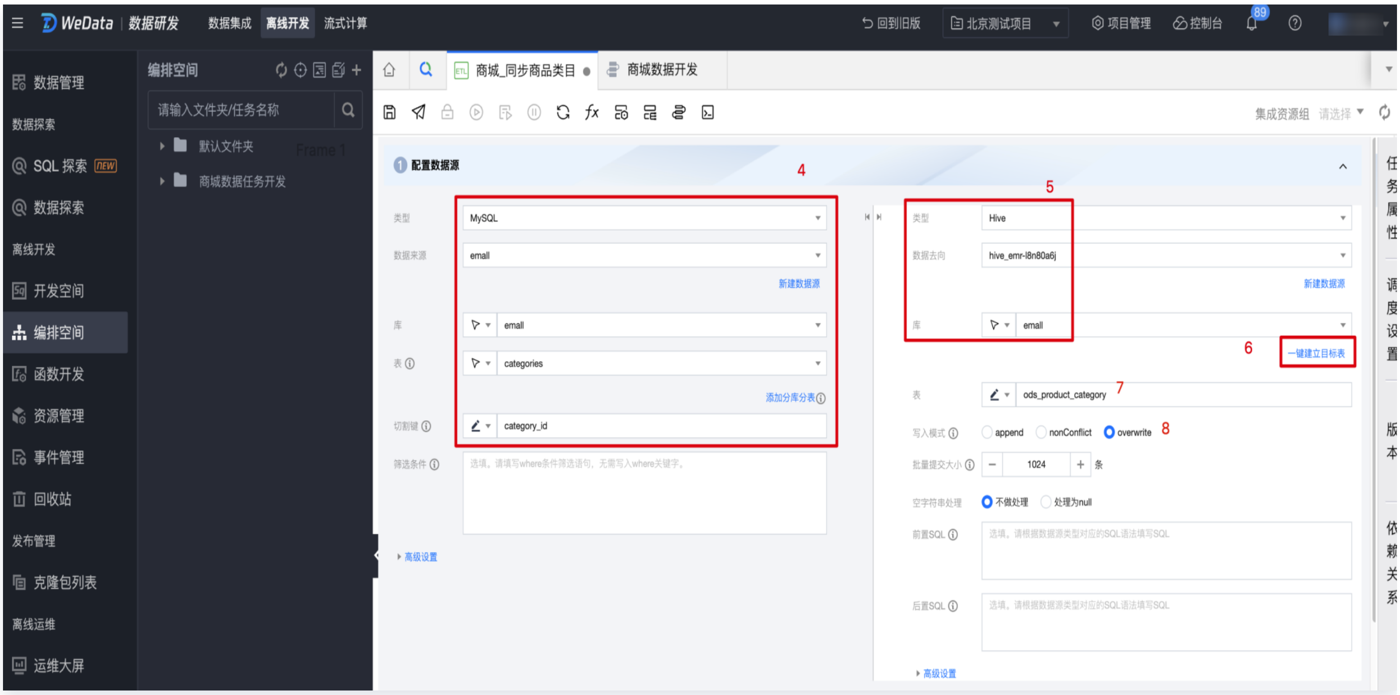
```
CREATE TABLE IF NOT EXISTS `email`.`ods_product_category`
`category_id` int,
`category_name` string)
row format delimited
fields terminated by '\t'
STORED AS PARQUET;
```

Notes:

Please modify the table creation statement in the pop-up box. Modify table name: ods_product_category.

2.4 Selection table: ods_product_category

2.5 Since the category table is basic information, here we select **overwrite**, that is, overwrite update every time.



配置数据源

类型 MySQL

数据来源 email

库 email

表 categories

切割键 category_id

筛选条件 选填，请填写where条件筛选语句，无需写入where关键字。

类型 Hive

数据去向 hive_emr-l8n80a6j

库 email

表 ods_product_category

写入模式 ○ append ○ nonConflict ● overwrite

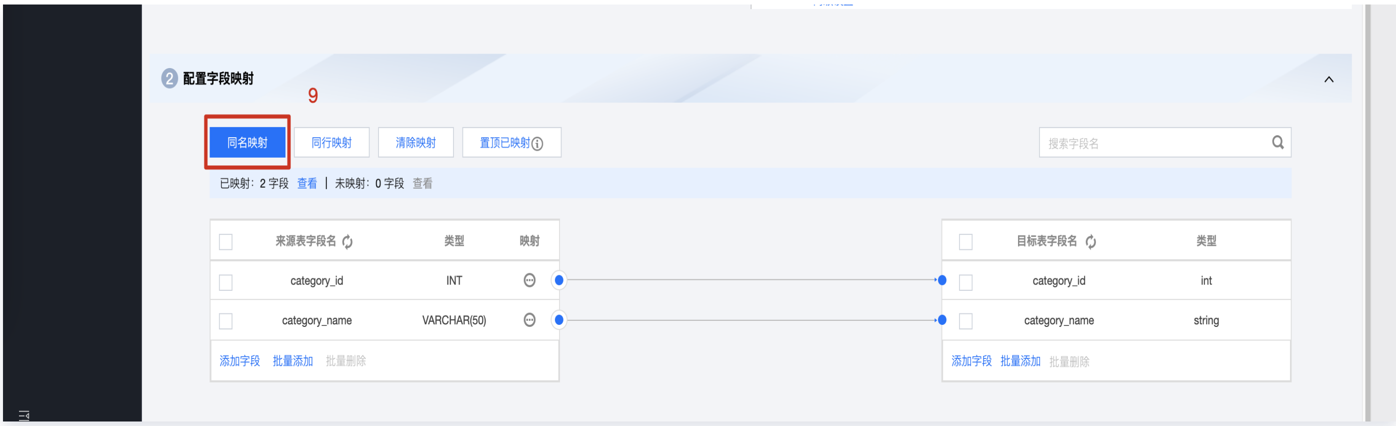
批量提交大小 1024 条

空字符串处理 ● 不做处理 ○ 处理为null

前置SQL 选填。请根据数据源类型对应的SQL语法填写SQL

后置SQL 选填。请根据数据源类型对应的SQL语法填写SQL

3. Configure field mappings. Here, we need to map the fields of the source table to the fields of the target table one-to-one. Since our table structures are identical, we can use **same name mapping**.



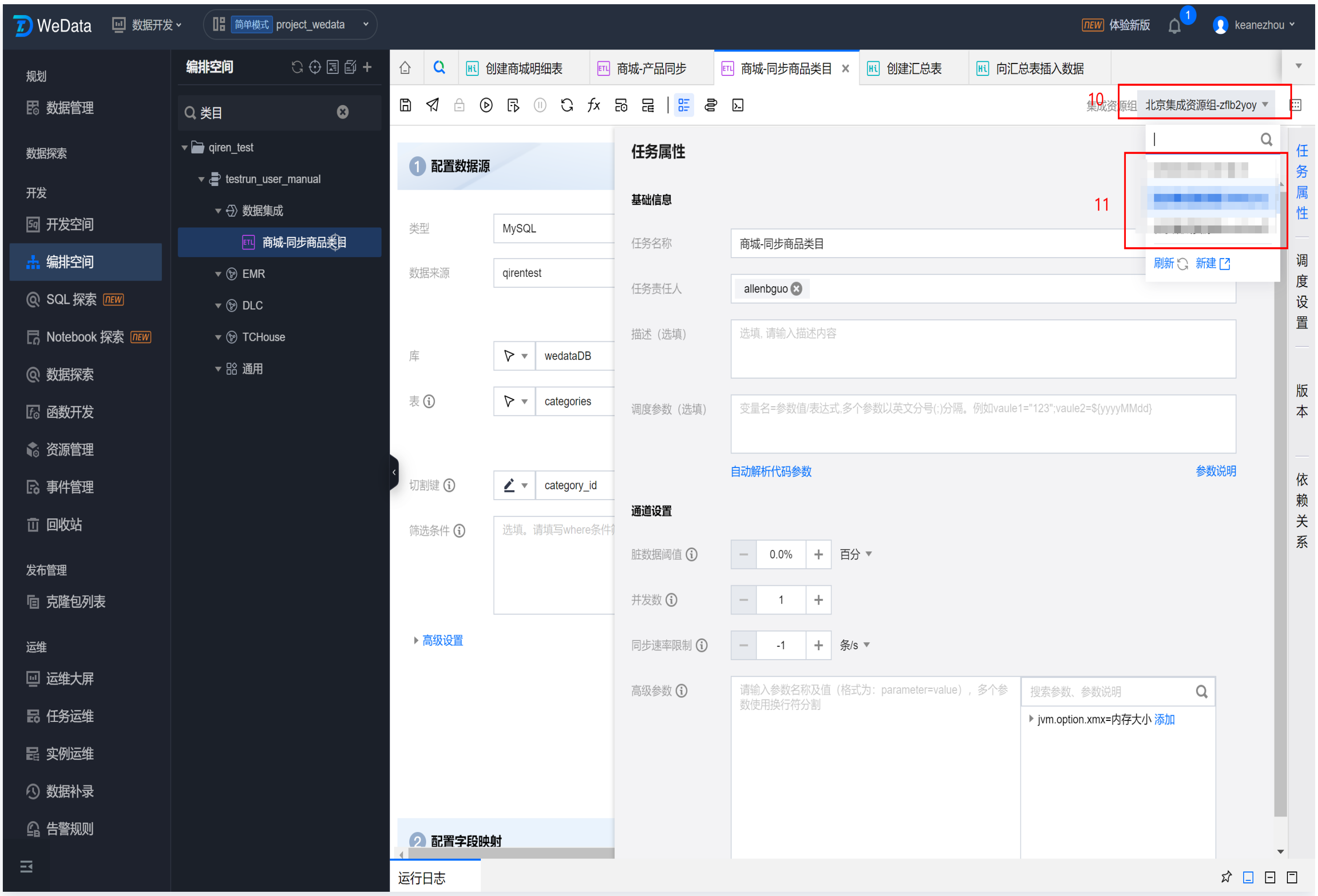
已映射：2 字段 查看 | 未映射：0 字段 查看

来源表字段名	类型	映射
category_id	INT	⊖
category_name	VARCHAR(50)	⊖

目标表字段名	类型
category_id	int
category_name	string

4. Set task properties. Select the integration resource group required at runtime.

> **Notes:**
> Here, it is essential to ensure network connectivity among the integration resource group, MySQL instance, and EMR cluster. Tencent Cloud resources in the same region must be purchased.



集成资源组：北京集成资源组-zflb2yoy

配置数据源
- 类型：MySQL
- 数据来源：qirentest
- 库：wedataDB
- 表：categories
- 切割键：category_id
- 筛选条件：选填，请填写where条件

任务属性

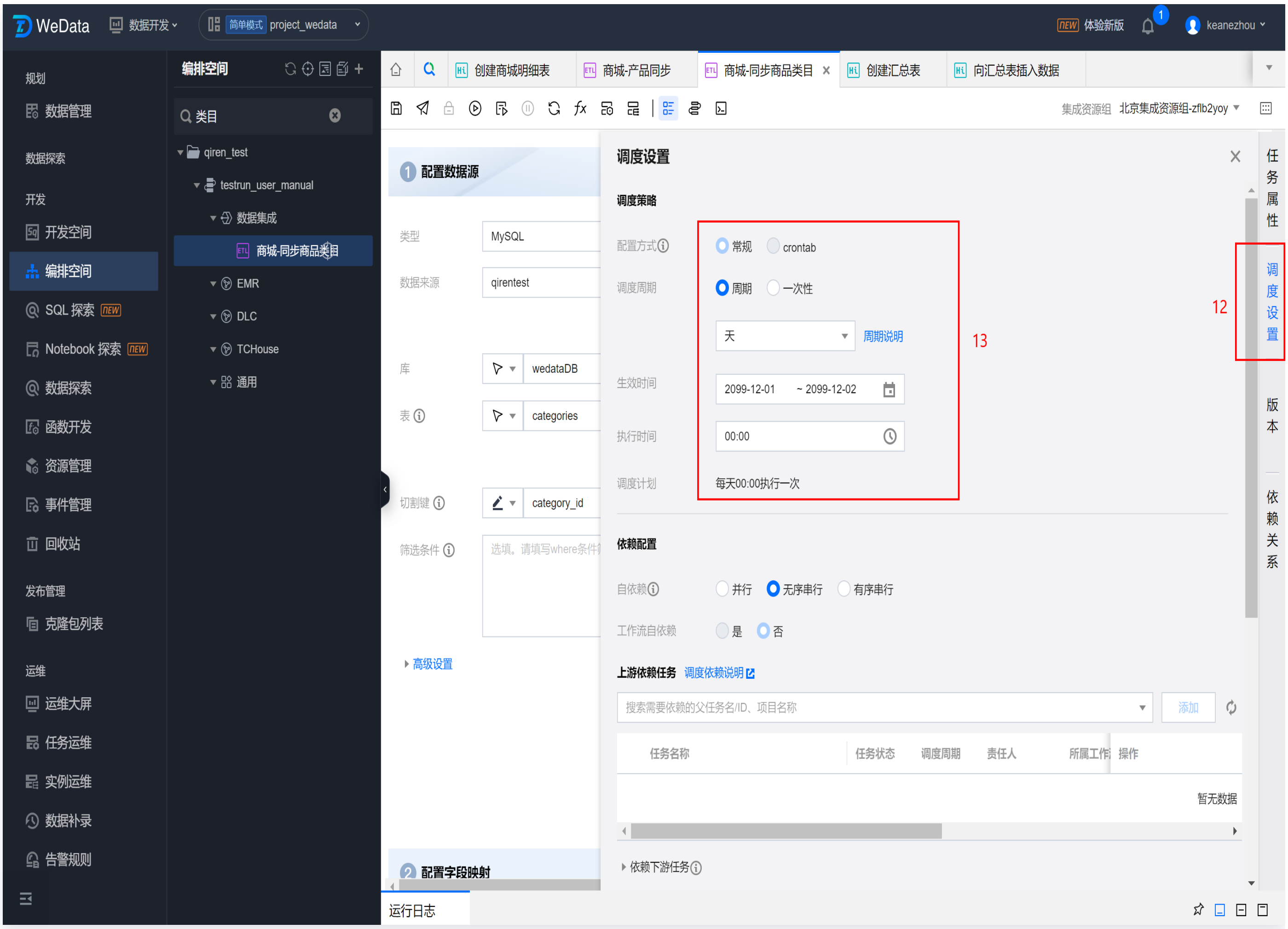
基础信息
- 任务名称：商城-同步商品类目
- 任务责任人：allenbguo
- 描述（选填）：选填，请输入描述内容
- 调度参数（选填）：变量名=参数值/表达式,多个参数以英文分号(;)分隔。例如vaule1="123";vaule2=${yyyyMMdd}

通道设置
- 脏数据阈值：0.0% 百分
- 并发数：1
- 同步速率限制：-1 条/s
- 高级参数：请输入参数名称及值（格式为：parameter=value），多个参数使用换行符分割
- 搜索参数、参数说明
- jvm.option.xmx=内存大小 添加

5. Set task scheduling. Here, we need to set the running strategy of the task. Since the frequency of changes in the category table is small, we set a daily sync once in the wee hours.

- Scheduling method: Periodic scheduling.
- Effective date: Default

- Scheduling cycle: Day
- Execution time: 00:00



类型 MySQL

数据来源 qirentest

库 wedataDB

表 categories

切割键 category_id

筛选条件 选填，请填写where条件

调度设置

调度策略

配置方式 ● 常规 ○ crontab

调度周期 ● 周期 ○ 一次性

天 周期说明

生效时间 2099-12-01 ~ 2099-12-02

执行时间 00:00

调度计划 每天00:00执行一次

依赖配置

自依赖 ○ 并行 ● 无序串行 ○ 有序串行

工作流自依赖 ○ 是 ● 否

上游依赖任务 调度依赖说明

搜索需要依赖的父任务名/ID、项目名称 添加

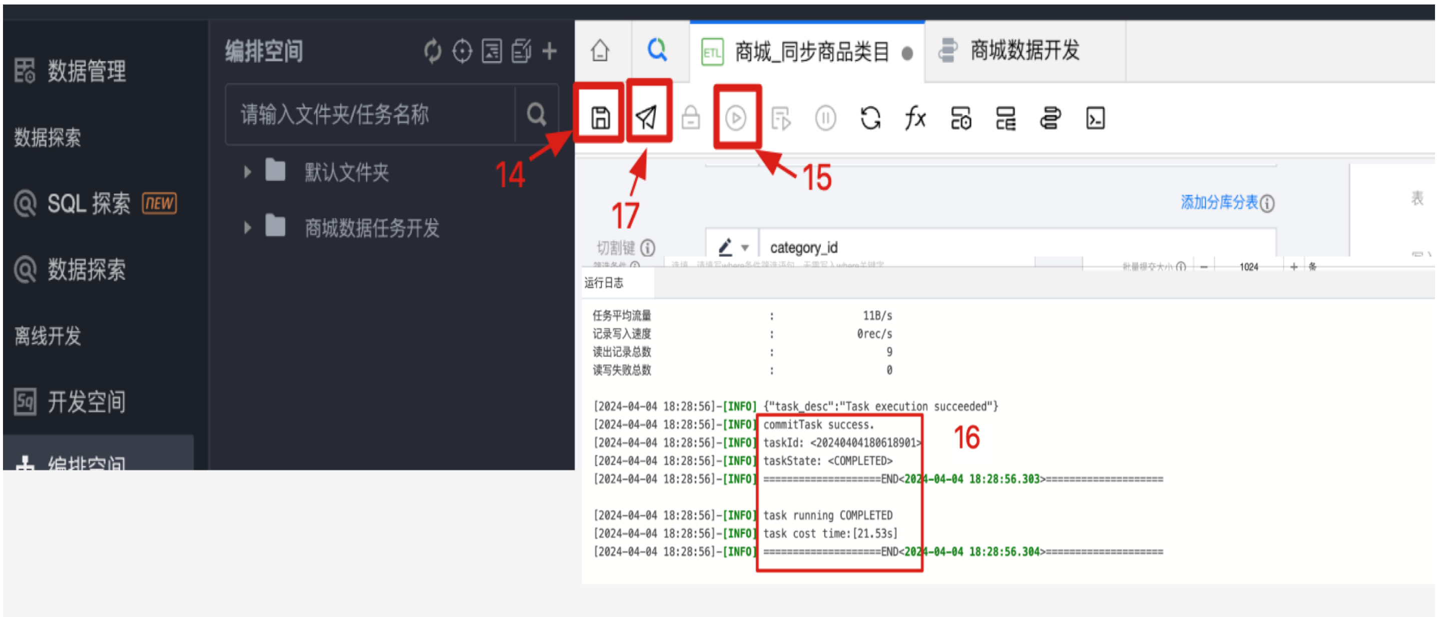
任务名称	任务状态	调度周期	责任人	所属工作	操作
暂无数据					

依赖下游任务

6. Set task attributes:
7. after the above steps are completed, please save data promptly.
8. You can perform a dry run once before the official submission. At this point, the system will also check the integrity of the configuration and network connectivity. Once the detection passes, the system will start running immediately. The running log will appear at the bottom of the page.
9. The logs and progress of the trial run can be seen at the bottom of the page. When Success or Completed appears, it indicates that the trial run is successful.
10. Submit the task to the scheduling resource server. Once the set running time is reached, the task will automatically start running. At this point, the configuration of the data integration task is completed.

> **Notes:**
>
> Upon submission, the system will automatically detect the integrity and network connectivity of the configuration.

> If a notification about network connectivity issues appears, do not ignore it. Check immediately whether the networks of the integration resource group, MySQL instance, and EMR cluster are reachable.



```
任务平均流量                  :              11B/s
记录写入速度                  :             0rec/s
读出记录总数                  :                  9
读写失败总数                  :                  0

[2024-04-04 18:28:56]-[INFO] {"task_desc":"Task execution succeeded"}
[2024-04-04 18:28:56]-[INFO] commitTask success.
[2024-04-04 18:28:56]-[INFO] taskId: <20240404180618901>
[2024-04-04 18:28:56]-[INFO] taskState: <COMPLETED>
[2024-04-04 18:28:56]-[INFO] ====================END<2024-04-04 18:28:56.303>====================

[2024-04-04 18:28:56]-[INFO] task running COMPLETED
[2024-04-04 18:28:56]-[INFO] task cost time:[21.53s]
[2024-04-04 18:28:56]-[INFO] ====================END<2024-04-04 18:28:56.304>====================
```

By completing steps 1 – 17, you have completed synchronizing the category table from the MySQL database to the EMR cluster. And every midnight, WeData will automatically pull full data for overwrite update.

Synchronizing City Table

We will now create the second offline synchronization task to synchronize the City Table from MySQL to the Hive table.

Since you have completed a sync task of a data table, you should have a certain understanding of data synchronization. Here we add a step: develop a synchronization strategy.

The synchronization strategy generally includes:

- **Offline sync or real-time synchronization?**

  When the business has small requirements for access latency and real-timeness, we generally select offline sync.
- **Additional Notes**

Difference between real-time synchronization and offline sync:

Method	Description
Real-time Sync	**Definition:** Real-time synchronization refers to data being transmitted to the target system almost immediately after a change occurs in the source system.
	Use cases: Applications requiring high real-timeness of data, such as financial transactions, online collaboration tools.

	Strengths: • Real-timeness: Data changes can be immediately reflected in the target system, reducing the time window of data inconsistency. • Data consistency: Due to the fast synchronization speed, problems caused by data inconsistency can be reduced.
	Drawbacks: • High system resource consumption: A continuous network connection and relatively high system resources are required to maintain real-timeness. • High cost: Real-time synchronization may need more complex technical support and higher O&M costs. • Complexity: Implementing real-time synchronization requires more complex logic to handle data conflicts and synchronization status.
Offline sync	**Definition:** Offline sync refers to data not being transmitted immediately after a change occurs in the source system, but rather batch-transmitted at a specific time point or under certain conditions.
	Use cases: Environments where data has low real-time requirement or network conditions are unstable, such as data backup on mobile devices, regular synchronization of specific enterprise data, etc.
	Strengths: • Low system resource consumption: Synchronization can be arranged according to network and system resource conditions to reduce the need for real-time resources. • Low cost: Compared with real-time synchronization, offline sync has lower O&M costs and technical support requirements. • Flexibility: The time and frequency of synchronization can be arranged according to your actual needs.
	Drawbacks: • Delay: There may be a delay in data sync, and the latest data changes may not be reflected immediately. • Data consistency risk: If the sync interval is relatively long, it may result in data inconsistency.
Summary: The choice of sync method depends on business requirements, the importance of data, network conditions, and cost budget. Real-time synchronization is suitable for scenarios with extremely high requirements for data real-timeness, while offline sync is applicable to environments that can tolerate a certain amount of data delay. In some cases, these two strategies can also be used in conjunction with each other. For example, use offline sync when network conditions are	

> poor and real-time synchronization when network conditions are good to balance real-timeness and cost.

- **Should incremental or full synchronization policy be selected?**

  In real business scenarios, incremental synchronization is generally selected. Full synchronization is only selected during data table initialization.
  In this tutorial, slice fields such as date are not set in city, category, and Product Table. We depend on selecting full synchronization.
  In following steps, when synchronizing the order table, we will introduce how to set up incremental synchronization.
- **If it is offline sync, then how often do you select for synchronization, per day or per hour?**

  Synchronization frequency needs to be determined according to business needs. The smaller the frequency, the larger the resource consumption.
  In this tutorial, we all select to sync once every day at wee hours.
  The synchronization strategy of the City Table is identical to that of the category table. Please refer to steps 1 - 17 in the category table and repeat the operation.

> ⚠ **Notes:**
> - **The following steps are all marked with the step numbers in the diagram.**
> - Step 3: Task name: Mall_Synchronize city information
> - Step 4: Selection table: cities;
> - Step 6, 7: The table name needs to be modified to: ods_order_city;
> - Steps 10 - 13: These four procedures are easy to ignore;
> - Step 15: Regardless of how familiar you are with the operation, remember to run it once before submission to ensure the task runs accurately.

Synchronize Product Table

Create the third offline synchronization task below, synchronizing the Product Table from MySQL to the Hive table.
Still think before reoperating.

No.	Issue	Conclusion
1	Offline sync or real-time synchronization?	Offline
2	Should incremental or full synchronization policy be selected?	Full ⓘ **Note** Actually, it should be incremental. Here, we select full for teaching purposes.

3	If it is offline sync, then how often do you select for synchronization, per day or per hour?	Sync at midnight every day

Based on the above thinking, the synchronization strategies of the Product Table, category table and City Table are also identical. Please continue to refer to steps 1 – 17 in the category table and repeat the operation.

> ⚠ **Notes:**
> - **The following steps are all marked with the step numbers in the diagram.**
> - Step 3: Task name: Mall_Synchronize product information.
> - Step 4: Selection table: products.
> - Step 6, 7: The table name needs to be modified to: ods_product_product.
> - Steps 10 – 13: These four procedures are easy to ignore.
> - Step 15: Regardless of how familiar you are with the operation, remember to run it once before submission to ensure the task runs accurately.

Synchronize Order Table

Create the fourth offline synchronization task below, synchronizing the order table from MySQL to the Hive table.
Still think before reoperating.

No.	Issue	Conclusion
1	Offline sync or real-time synchronization?	Offline
2	Should incremental or full synchronization policy be selected?	Incremental **Note** Since the order table often has a large data volume, it is more suitable for incremental synchronization, that is, synchronizing the order data of the previous day every day.
3	If it is offline sync, then how often do you select for synchronization, per day or per hour?	Sync at midnight every day

Through the above thinking, the synchronization strategy of the order table is different from that of the above three tables. Here we have selected incremental synchronization, which we will focus on introducing in this step.

Incremental sync logic:

There is a special field in the original data order table: order_time, which will change as time goes by. Therefore, we can use the order creation time as a partition and ensure the pulling of real-time incremental data daily based on order_time.

To dynamically compare **task running time** (represented as ${yyyy-MM-dd}) with **order_time** during scheduling

For example: When the running time is 2024-04-01 00:10, then ${yyyy-MM-dd} = 2024-04-01, and meanwhile ${yyyy-MM-dd**-1d**} = 2024-03-31

Therefore, we can use date(order_time) = '${yyyy-MM-dd**-1d**}' to represent the data of yesterday.

Create offline synchronization task

First, please continue to refer to steps 1 – 13 in the category table and repeat the operation.

> ⚠ **Notes:**
>
> - **The following serial numbers are all marked with the step numbers in the diagram.**
> - Step 4: Selection table: orders;
> - Step 6, 7: The table name needs to be modified to: ods_order_order;
> - Do not perform operation steps 14 – 17 (do not submit). We need to modify the following configuration.

Operation Demonstration Screenshot:

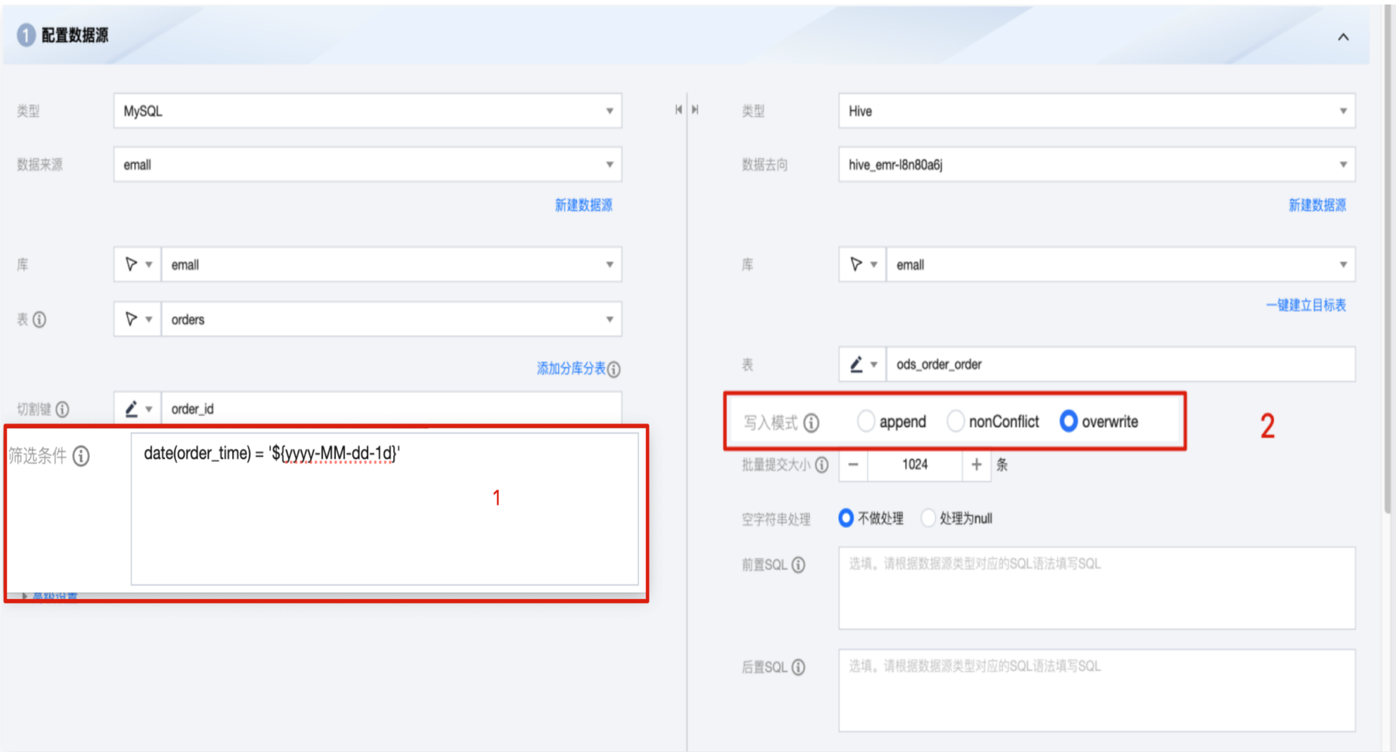
1. Open the Configure Data Source page. Fill in the **filter conditions** on the left. Select **write mode** on the right as **overwrite.**

- Filter conditions: Fill in the corresponding filter statement according to the data type. This statement will serve as the filter conditions for the data to be synchronized.
- Write mode

  1.1 Append: Retain original data and append new rows.

  1.2 nonConflict: Error reported on data conflict.

  1.3 Overwrite: Delete the original data and rewrite it.

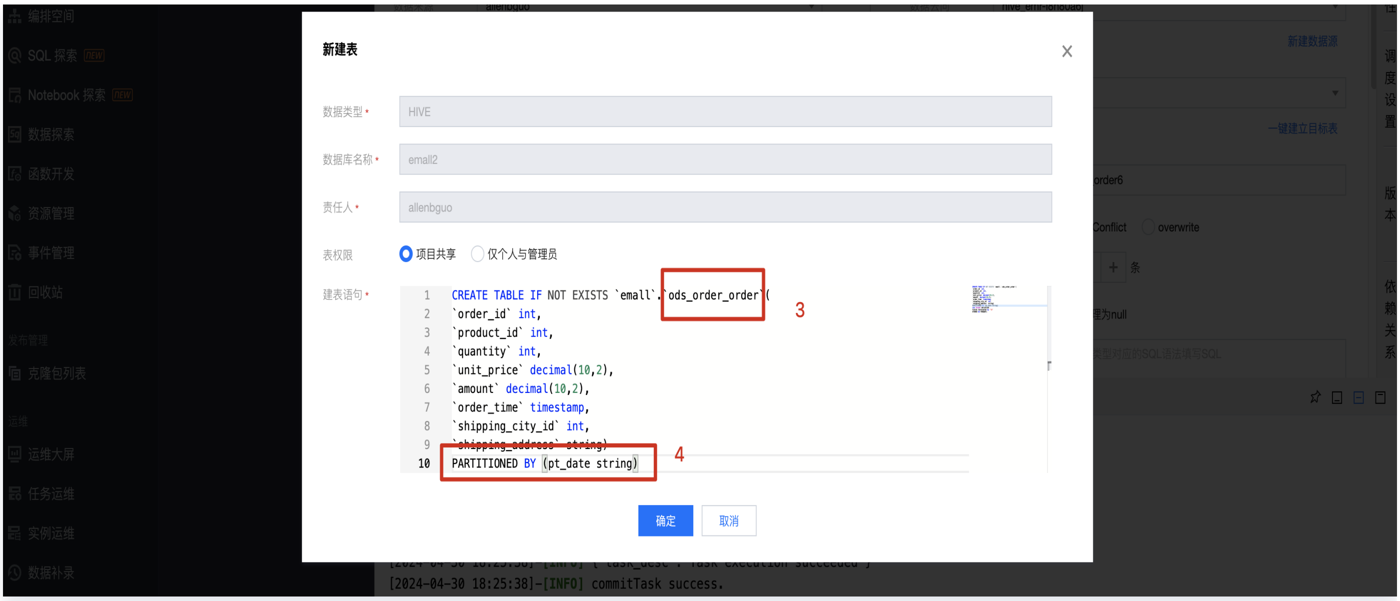


配置数据源

类型: MySQL
数据来源: email
新建数据源
库: email
表: orders
添加分库分表
切割键: order_id
筛选条件: date(order_time) = '${yyyy-MM-dd-1d}'

类型: Hive
数据去向: hive_emr-l8n80a6j
新建数据源
库: email
一键建立目标表
表: ods_order_order
写入模式: ○ append ○ nonConflict ● overwrite
批量提交大小: 1024 条
空字符串处理: ● 不做处理 ○ 处理为null
前置SQL: 选填。请根据数据源类型对应的SQL语法填写SQL
后置SQL: 选填。请根据数据源类型对应的SQL语法填写SQL

2. Open the **Create Table** page, modify the table name to ods_order_order, and add the partition field PARTITIONED BY (pt_date date) at the same time.



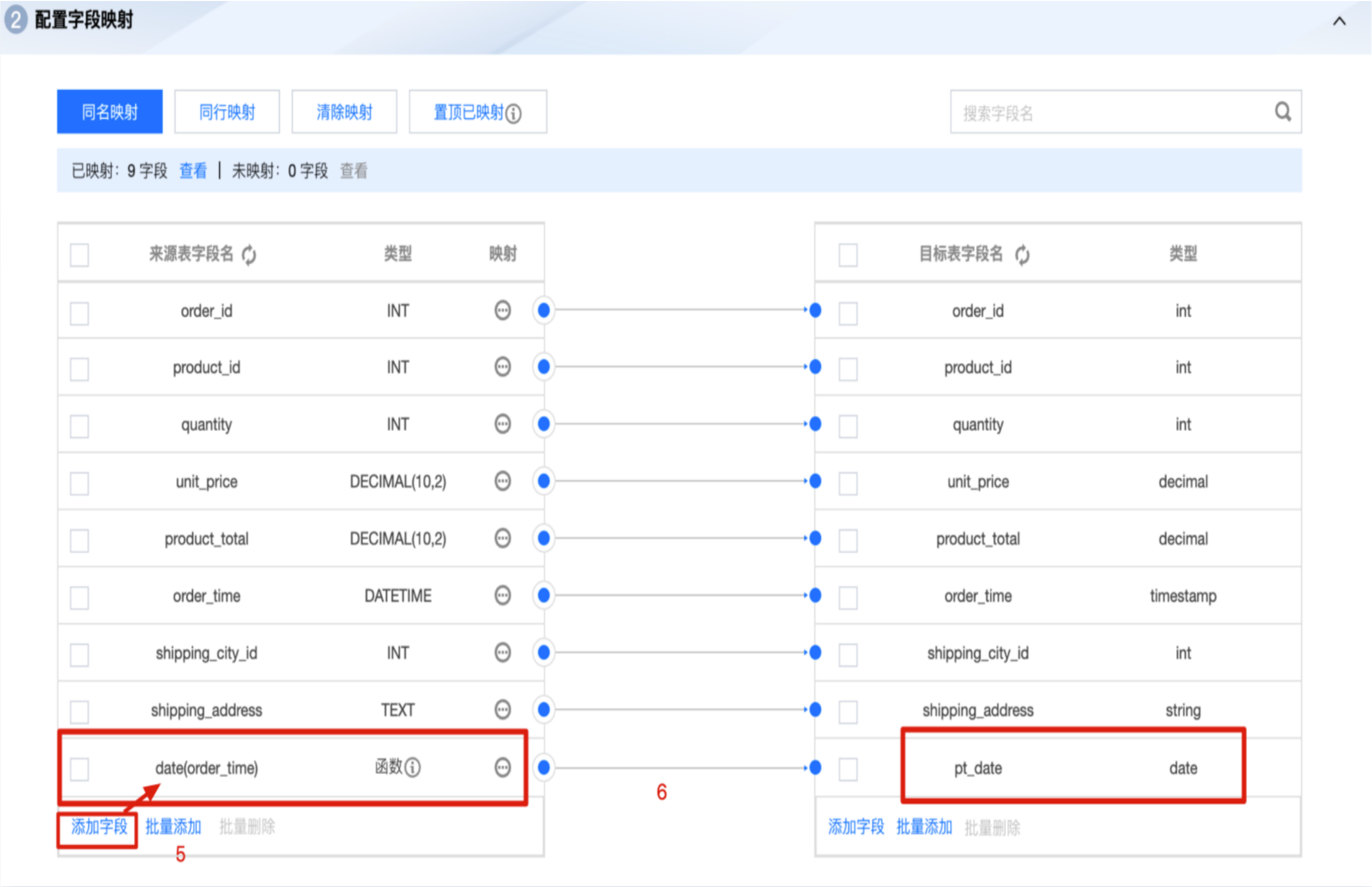
新建表

数据类型 *: HIVE
数据库名称 *: emall2
责任人 *: allenbguo
表权限: ● 项目共享 ○ 仅个人与管理员
建表语句 *:

```
CREATE TABLE IF NOT EXISTS `emall`.`ods_order_order`(
`order_id` int,
`product_id` int,
`quantity` int,
`unit_price` decimal(10,2),
`amount` decimal(10,2),
`order_time` timestamp,
`shipping_city_id` int,
`shipping_address` string)
PARTITIONED BY (pt_date string)
```

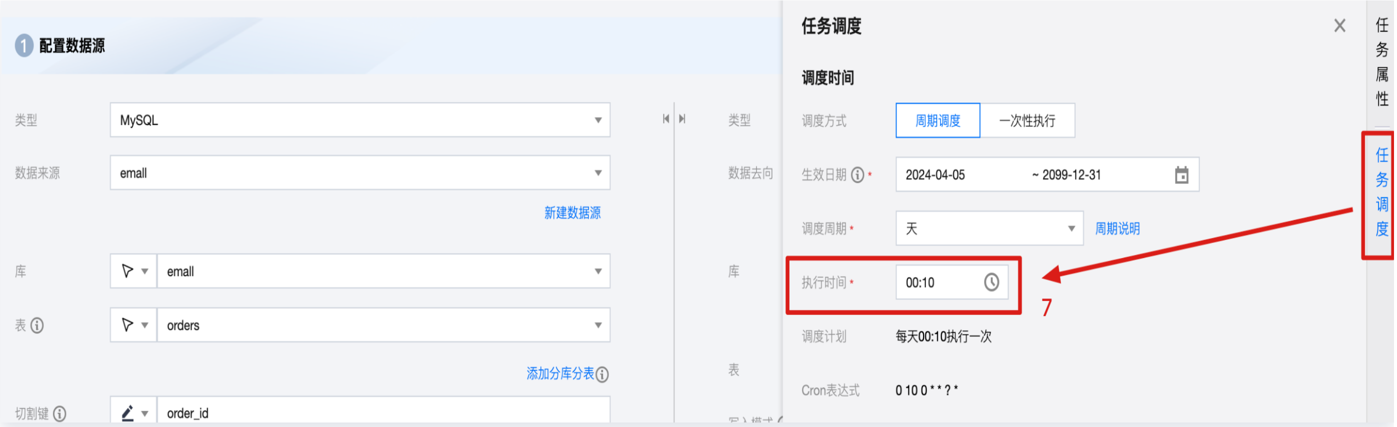
```
[2024-04-30 18:25:38]-[INFO] commitTask success.
```

3. On the **Configure Field Mappings** page, click **Add Fields.** The field name is date(order_time) and the type is function. Click on the small circle on the right. Drag and drop with the mouse to the target field pt_date on the right to **establish mapping relationship**.



来源表字段名	类型	映射	目标表字段名	类型
order_id	INT		order_id	int
product_id	INT		product_id	int
quantity	INT		quantity	int
unit_price	DECIMAL(10,2)		unit_price	decimal
product_total	DECIMAL(10,2)		product_total	decimal
order_time	DATETIME		order_time	timestamp
shipping_city_id	INT		shipping_city_id	int
shipping_address	TEXT		shipping_address	string
date(order_time)	函数		pt_date	date

4. Click on the **Task Scheduling** button on the right task bar. Find **Execution Time** on the Task Scheduling page and modify the execution time to: 00:10.



类型 MySQL

数据来源 email

库 email

表 orders

切割键 order_id

调度时间

调度方式 周期调度

生效日期 2024-04-05 ~ 2099-12-31

调度周期 天

执行时间 00:10

调度计划 每天00:10执行一次

Cron表达式 0 10 0 * * ? *

Create an order table SQL:

```
--One-click creation of order table
CREATE TABLE IF NOT EXISTS `email`.`ods_order_order`(
`order_id` int,
`product_id` int,
`quantity` int,
`unit_price` decimal(10,2),
`amount` decimal(10,2),
`order_time` timestamp,
`shipping_city_id` int,
`shipping_address` string)
PARTITIONED BY (pt_date date)
row format delimited
fields terminated by '\t'
STORED AS PARQUET;
```

So far, we have completed all offline synchronization tasks from raw data tables to Hive tables. And every midnight, WeData will automatically conduct full/incremental data synchronization.

Summary

Now you have completed the study of the Data Integration part. Now summarize:

No.	Step Name
1	**Confirm the original data table and the data target table** Base Table: Read: Data Source Target Table: Write: Data Destination
2	**Confirm offline sync or real-time synchronization** According to business needs, if not necessary, offline sync is available to reduce resource consumption.
3	**Confirm incremental synchronization or full synchronization** Generally, full synchronization is required during data initialization, and incremental synchronization is used for periodic synchronization. During incremental synchronization, it is required to set filter conditions to ensure no overlap when pulling data.
4	**Confirm network environment interoperability** Three environments are involved in the sync process: 1. original database instance 2. Integration Resource Group 3. EMR cluster ⚠ **Notes:** Must ensure that the integration resource group can access the original database instance and EMR cluster.

Below we will learn part of offline development, that is, performing data processing in the Hive table of the EMR cluster.

Offline development

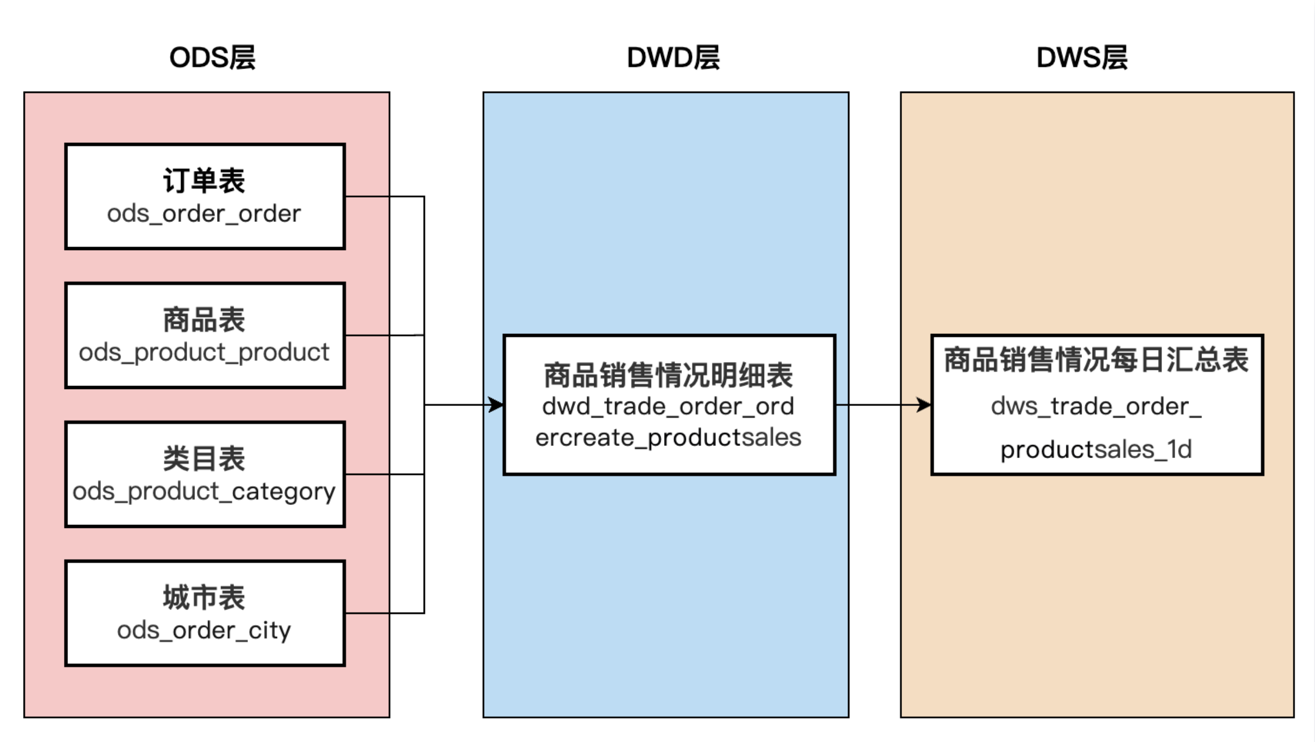
Last updated：2025-04-18 15:49:07

Through the previous steps, we have synchronized all the raw data to the Hive tables in the EMR cluster.

- However, these data are all in their raw structures and cannot be directly used for business purposes.
- Combining the content from the **Data table structure design** step, we have analyzed the business requirements and divided the data warehouse hierarchy.

Next, we will complete the generation and data processing of the detail and summary tables through data development.

Offline development task design



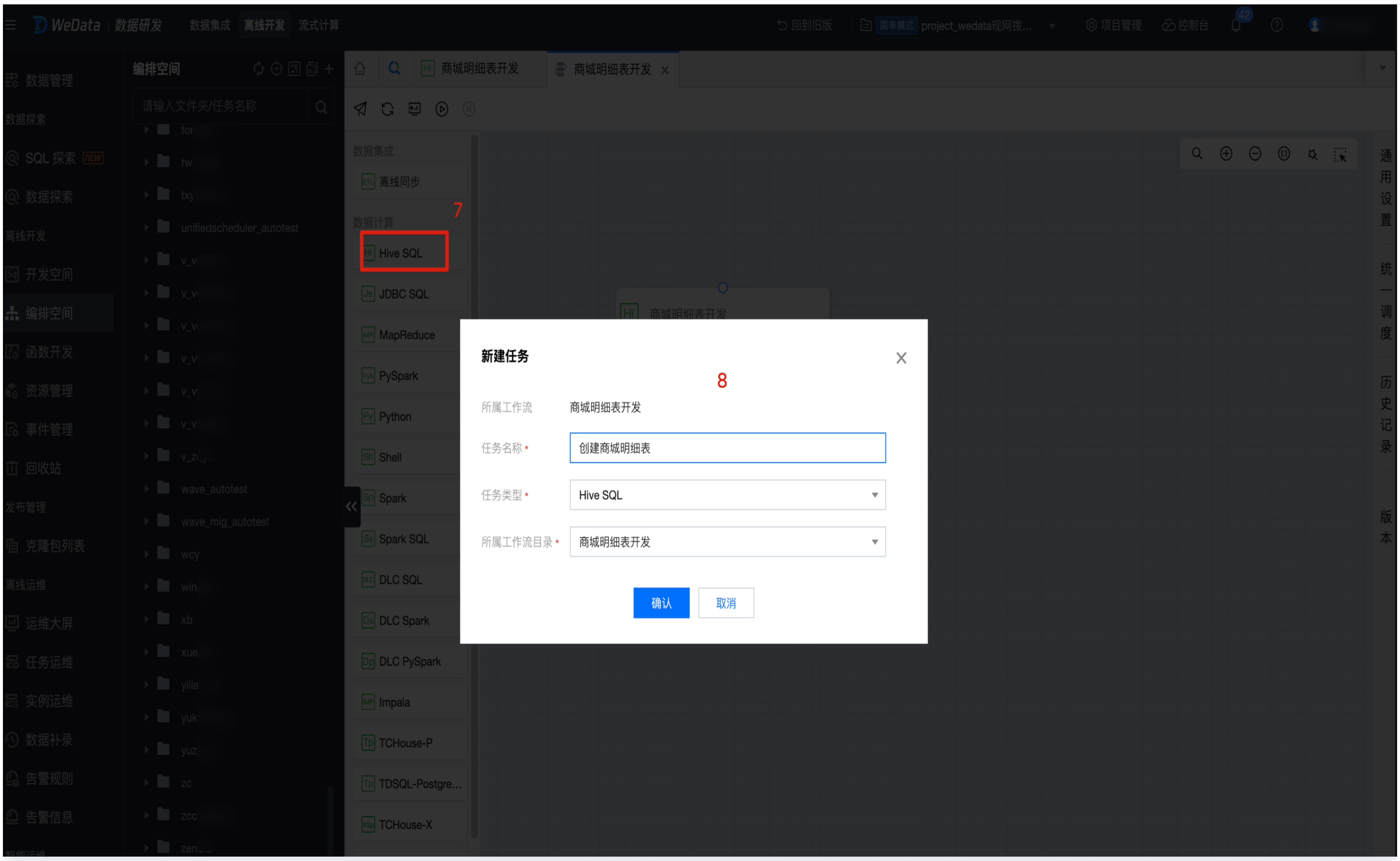
Offline development task development

Detail Table Development

Completing the detail table development mainly includes the following 4 steps:

Create a shopping mall detail table

1. In the **Orchestration Space** module, under any folder in the shopping mall detail table development workflow directory, click the **Hive SQL icon** under Data Calculation, and create a new **Hive SQL** task. Name the task 'Create a shopping mall detail table', task type is Hive SQL, then click **Confirm**.



新建任务

所属工作流 商城明细表开发

任务名称 * 创建商城明细表

任务类型 * Hive SQL

所属工作流目录 * 商城明细表开发

确认 取消

2. After confirmation, the Hive SQL development script page will pop up, follow these steps to complete the Hive SQL development task.

- Data Source: Select hive_emr-XXX.
- Scheduling Resource Group: Select the resource group we purchased, it is recommended to be in the same domain.
- Development Script: Write the table creation SQL script for the shopping mall detail table on the script page.
- Scheduling Settings: In the right taskbar, click **Scheduling Settings**, select the scheduling cycle as **One-time**, and default the execution event.
  - One-time: This Hive SQL executes only once according to the execution time.
  - Cycle: This Hive SQL executes periodically according to the execution time.
- **Click Save** button: Save this Hive SQL task.
- **Click Run** button: Execute the task once to verify the script task.
- **Click Submit** button: Officially submit this task to the scheduling resource server. The task will be executed according to the scheduling cycle at the specified time.

3. Create Shopping Mall Detail Table HiveQL Statement:

```
--Create Detail Table HiveQL Statement
CREATE TABLE
  IF NOT EXISTS emall.dwd_trade_order_ordercreate_productsales (
    order_id INT COMMENT 'Order ID, primary key',
    product_id INT COMMENT 'Product ID',
    product_name STRING COMMENT 'Product Name',
    category_id INT COMMENT 'Product Category ID',
    category_name STRING COMMENT 'Product Category Name',
    quantity INT COMMENT 'Quantity, positive integer',
    unit_price DECIMAL(10, 2) COMMENT 'Unit Price, with two decimal places',
    amount DECIMAL(10, 2) COMMENT 'Subtotal Amount, i.e., quantity times unit
price',
    order_time TIMESTAMP COMMENT 'Order Time, accurate to the minute',
    shipping_city_id INT COMMENT 'Shipping City ID, foreign key',
    shipping_city_name STRING COMMENT 'City Name',
    shipping_address STRING COMMENT 'Shipping Address, including province, city,
district, and detailed address'
  )
  COMMENT 'Product Sales Detail Table, records the sales details of each order'
  PARTITIONED BY (pt_date STRING)
```

```
    row format delimited fields terminated by '\t'
    STORED AS PARQUET;
```

By completing steps 7 – 15, you have created a Hive table in the Hive data source within the EMR cluster.

Write data to the Detail Table

Next, we will start writing data into the detail table:
Please repeat steps 7 – 15, paying attention to the following:

> ⚠ **Note:**
>
> - **The following steps correspond to the step numbers indicated in the illustrations**
> - In the same workflow, create a new HiveQL node
> - Step 8: Name it: Insert data into the Detail Table;
> - Steps 9 and 10: These two steps are often overlooked;
> - Step 11: The HiveQL statement is as follows;
> - Step 12:
>
>   Change the schedule cycle to: **Cycle**
>   Set the execution time to: 01:00
>   Description: This task needs to run once daily
> - Step 14: No matter how familiar you are with the operation, remember to test run before and after submission to ensure the task runs correctly.

HiveQL statement to insert data into the Detail Table:

```
--Insert data into the Detail Table HiveQL statement
SET hive.exec.dynamic.partition.mode=nonstrict;
INSERT INTO TABLE email.dwd_trade_order_ordercreate_productsales PARTITION
(pt_date)
SELECT
    o.order_id,
    o.product_id,
    p.product_name,
    p.category_id,
    ca.category_name,
    o.quantity,
    o.unit_price,
    o.amount,
    o.order_time,
    o.shipping_city_id,
    ci.city_name,
    o.shipping_address,
    o.pt_date
FROM
```

```
    emall.ods_order_order o
JOIN
    emall.ods_product_product p ON o.product_id = p.product_id
JOIN
    emall.ods_product_category ca ON p.category_id = ca.category_id
JOIN
    emall.ods_order_city ci ON o.shipping_city_id = ci.city_id
WHERE o.pt_date = '${yyyy-MM-dd-1d}';
```

At this point, we have completed the development task for the Detail Table. After the data is synchronized from the original table to the Hive cluster at midnight, the system will automatically link the four tables and compile the data into the Detail Table.

Note
There are some redundant fields in the Detail Table at this point, which is intended to minimize joins when processing the Summary Table to improve computational efficiency.

Summary Table Development

Create the Mall Summary Table

Next, we will start developing the Summary Table
First, please repeat steps 5 – 6 to create a new workflow: Mall Summary Table Development.
Next, repeat steps 7 – 15 to create the summary table. Pay attention to the following points:

Note:
- **The following steps correspond to the step numbers indicated in the illustrations**
- Create a new HiveQL node in the same workflow;
- Step 8: Name it: Create Mall Summary Table;
- Steps 9 and 10: These two steps are often overlooked;
- Step 11: The table creation SQL statement is as follows;
- Step 12:
  Set the scheduling cycle to:**Once**
  Set the execution time to: Default
  Description: This task only needs to run once
- Step 14: No matter how familiar you are with the operation, remember to test run before and after submission to ensure the task runs correctly.

Create the emall summary table HiveQL statement:

```
--Create summary table HiveQL statement
CREATE TABLE IF NOT EXISTS emall.dws_trade_order_productsales_1d (
    order_date DATE COMMENT 'Date of statistics, primary key',
    city_id INT COMMENT 'City ID',
    city_name STRING COMMENT 'City name',
    category_id INT COMMENT 'Product category ID',
    category_name STRING COMMENT 'Product category name',
    quantity INT COMMENT 'Total product sales, positive integer',
    amount DECIMAL(10, 2) COMMENT 'Total product sales in monetary terms, two 
decimal places kept'
)
COMMENT 'Daily summary table of product sales situations'
PARTITIONED BY (pt_date STRING)
row format delimited fields terminated by '\t'
STORED AS PARQUET;
```

Write data into the summary table

Next, repeat steps 7 – 15 to insert data into the summary table. Pay attention to the following points:

> ⚠ **Note:**
>
> - **The following steps correspond to the step numbers indicated in the illustrations**
> - Create a new HiveQL node in the same workflow;
> - Step 8: Name it: Insert Data into the Summary Table;
> - Steps 9 and 10: These two steps are often overlooked;
> - Step 11: The HiveQL statement is as follows;
> - Step 12:
>
>   Change the schedule cycle to: **Cycle**
>   Set the execution time to: 01:00
>   Description: This task needs to run once daily
> - Step 14: No matter how familiar you are with the operation, remember to test run before and after submission to ensure the task runs correctly.

Insert data into the Summary Table HiveQL statement:

```
--Insert data into the Summary Table HiveQL statement
SET hive.exec.dynamic.partition.mode=nonstrict;
INSERT INTO TABLE emall.dws_trade_order_productsales_1d PARTITION (pt_date)
SELECT
    p.pt_date AS order_date,
    p.shipping_city_id,
    p.shipping_city_name,
    p.category_id,
    p.category_name,
    SUM(p.quantity) AS quantity,
    SUM(p.amount) AS amount,
    p.pt_date
FROM
```

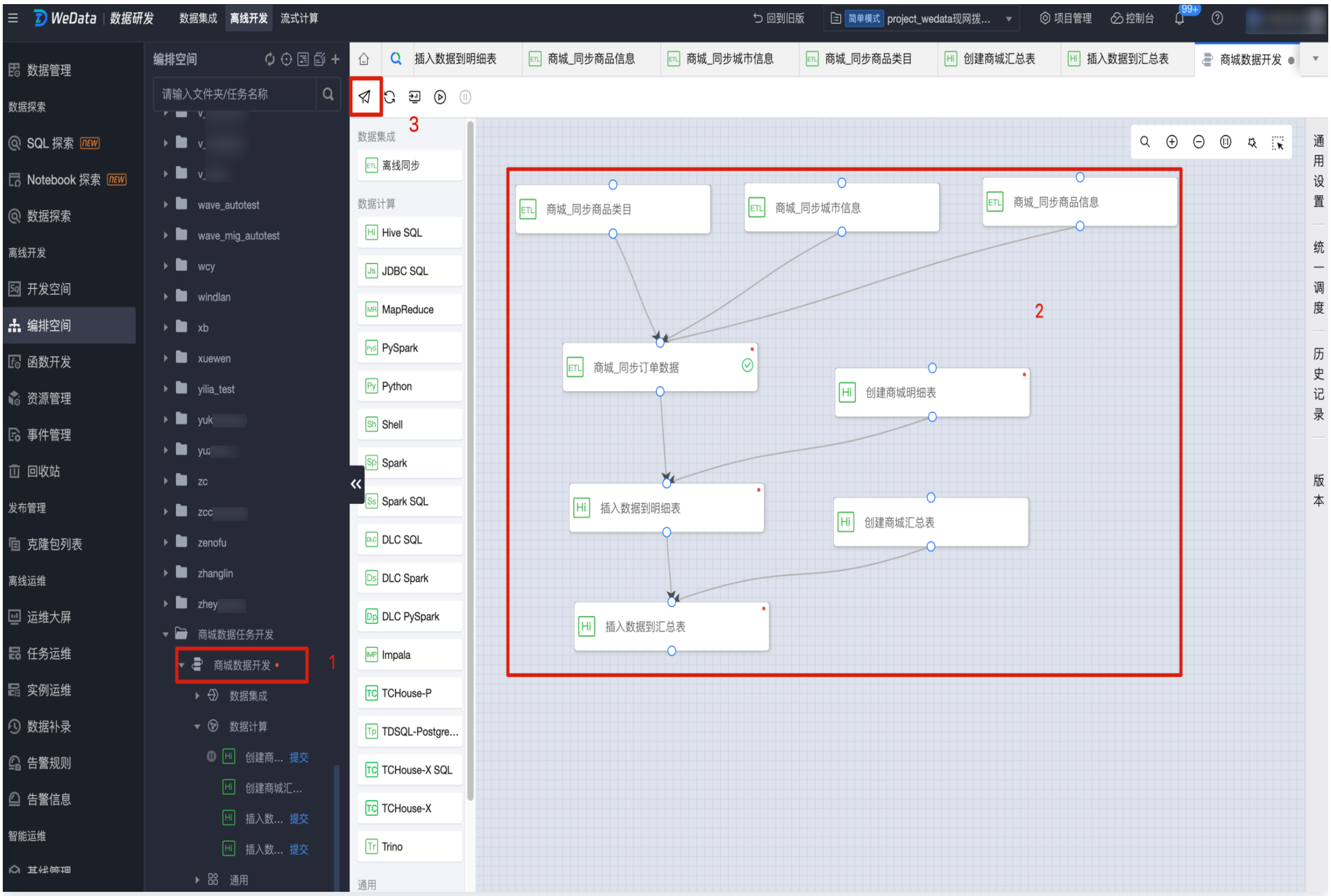
```
    emall.dwd_trade_order_ordercreate_productsales p
WHERE p.pt_date = '${yyyy-MM-dd-1d}'
GROUP BY
    p.pt_date,
    p.shipping_city_id,
    p.shipping_city_name,
    p.category_id,
    p.category_name;
```

We have completed the offline development tasks for the detail table and summary table. Every early morning, WeData will automatically perform the calculation tasks for the detail and summary tables.

Establish dependencies and submit

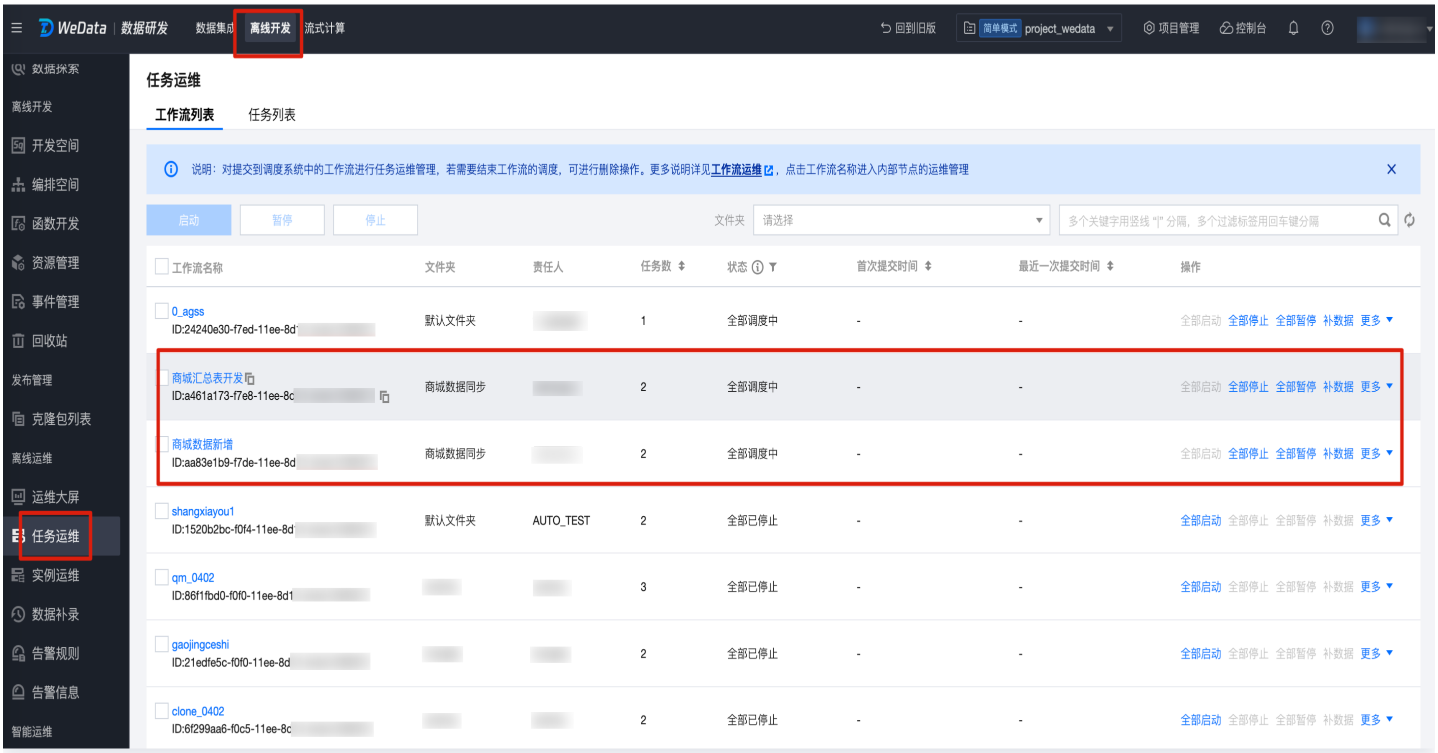
1. **Double-click** the mall data development workflow. A workflow canvas will pop up, displaying all the tasks under the workflow. Establish the dependencies among the workflows in sequence. Once completed, click the submit button.

- Establish dependencies in sequence:
    - Mall_Synchronize_Product_Category → Mall_Synchronize_Order_Data.
    - Mall_Synchronize_City_Information → Mall_Synchronize_Order_Data.
    - Mall_Synchronize_Product_Information → Mall_Synchronize_Order_Data.
    - Mall_Synchronize_Order_Data → Insert Data into the Detail Table.
    - Create Mall Detail Table → Insert Data into the Detail Table.
    - Insert Data into the Detail Table → Insert Data into the Summary Table.
    - Create Mall Summary Table → Insert Data into the Summary Table.
- Finally, submit the workflow. The tasks will run in an orderly manner according to the set dependencies.



Offline Development Task Ops

You can view the running status of workflows or offline tasks in Task Ops.



任务运维

工作流列表 任务列表

说明：对提交到调度系统中的工作流进行任务运维管理，若需要结束工作流的调度，可进行删除操作。更多说明详见工作流运维，点击工作流名称进入内部节点的运维管理

□ 工作流名称	文件夹	责任人	任务数	状态	首次提交时间	最近一次提交时间	操作
□ 0_agss ID:24240e30-f7ed-11ee-8d[redacted]	默认文件夹	[redacted]	1	全部调度中	-	-	全部启动 全部停止 全部暂停 补数据 更多 ▾
□ 商城汇总表开发 ID:a461a173-f7e8-11ee-8c[redacted]	商城数据同步	[redacted]	2	全部调度中	-	-	全部启动 全部停止 全部暂停 补数据 更多 ▾
□ 商城数据新增 ID:aa83e1b9-f7de-11ee-8d[redacted]	商城数据同步	[redacted]	2	全部调度中	-	-	全部启动 全部停止 全部暂停 补数据 更多 ▾
□ shangxiayou1 ID:1520b2bc-f0f4-11ee-8d[redacted]	默认文件夹	AUTO_TEST	2	全部已停止	-	-	全部启动 全部停止 全部暂停 补数据 更多 ▾
□ qm_0402 ID:86f1fbd0-f0f0-11ee-8d1[redacted]	[redacted]	[redacted]	3	全部已停止	-	-	全部启动 全部停止 全部暂停 补数据 更多 ▾
□ gaojingceshi ID:21edfe5c-f0f0-11ee-8d[redacted]	[redacted]	[redacted]	2	全部已停止	-	-	全部启动 全部停止 全部暂停 补数据 更多 ▾
□ clone_0402 ID:6f299aa6-f0c5-11ee-8c[redacted]	[redacted]	[redacted]	2	全部已停止	-	-	全部启动 全部停止 全部暂停 补数据 更多 ▾

Data Quality

Last updated：2025-04-18 15:51:17

In this step, we will complete quality monitoring of the data tables in the data warehouse to prevent dirty data from being transmitted downstream.

Quality Monitoring Task Design

When the following fields in the detailed table are empty, it will cause a serious impact on the summary table:

- Monitoring table: dwd_trade_order_ordercreate_productsales.
- Monitoring fields: amount, order_date.
- Monitoring logic: Depend on the completion of the detailed task and then automatically detect whether there are null values.

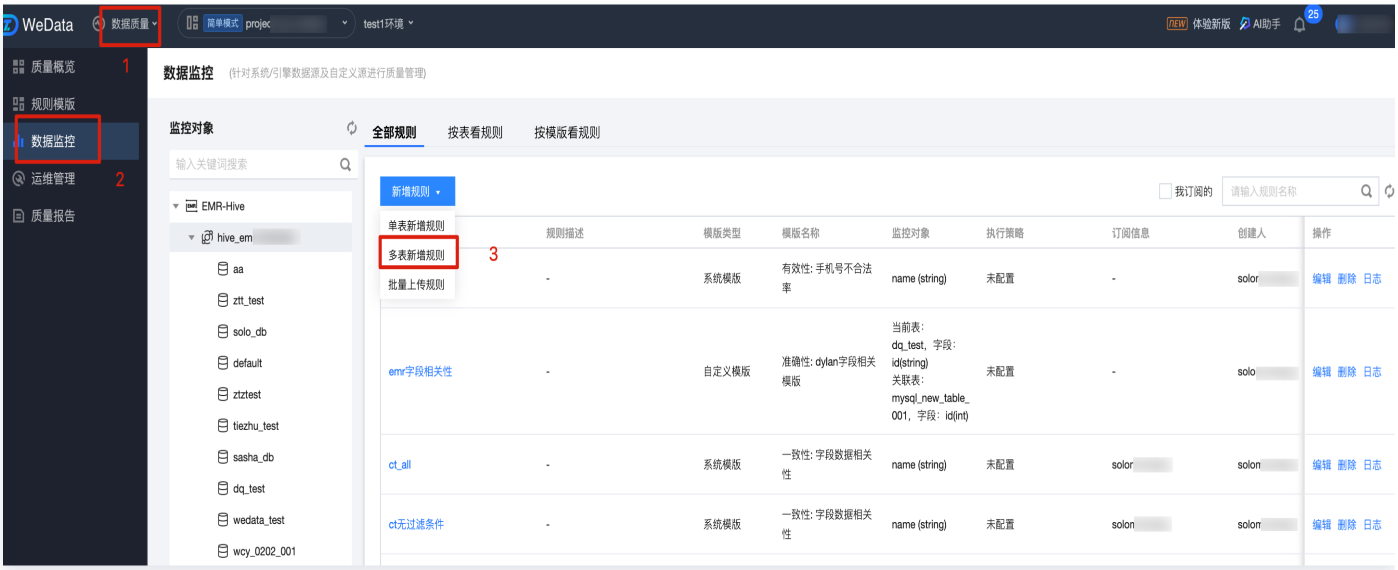
Quality Monitoring Task Development

Quality task development mainly includes the following 7 steps:

Null Value Detection Task

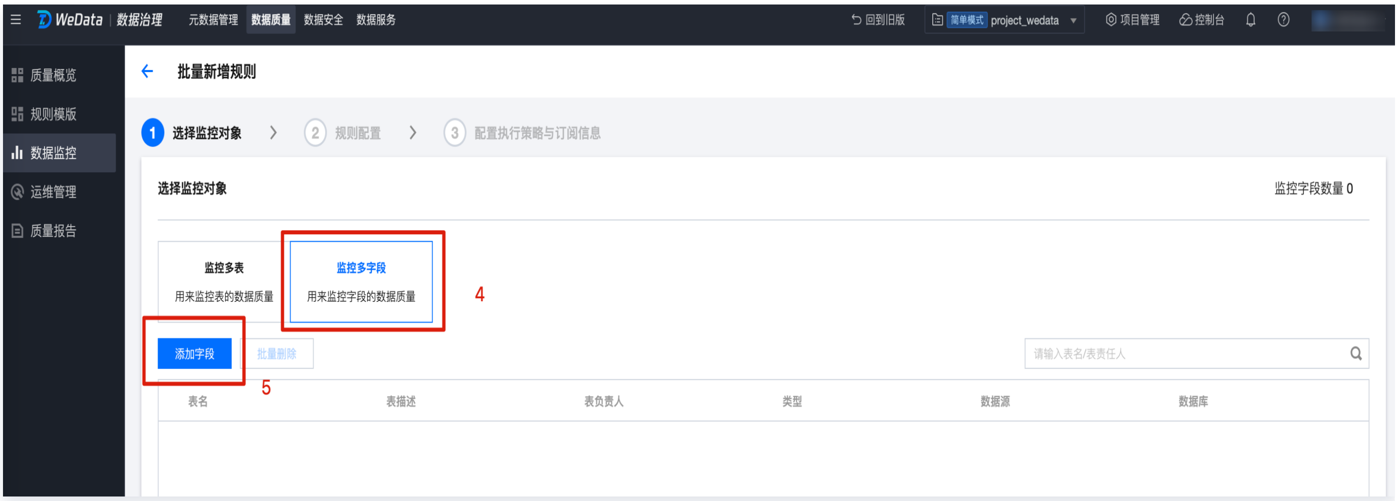
Step 1: Select Monitoring Fields

1. Click the **Data Quality** module, enter the Data Monitoring page, and click the **Add Rules to Multiple Tables** button again.
   - Add rules to multiple tables: Support setting monitoring rules for multiple tables or more fields at one time.



	规则描述	模版类型	模版名称	监控对象	执行策略	订阅信息	创建人	操作
	-	系统模版	有效性: 手机号不合法率	name (string)	未配置	-	solor[redacted]	编辑 删除 日志
emr字段相关性	-	自定义模版	准确性: dylan字段相关模版	当前表： dq_test，字段：id(string) 关联表： mysql_new_table_001，字段：id(int)	未配置	-	solo[redacted]	编辑 删除 日志
ct_all	-	系统模版	一致性: 字段数据相关性	name (string)	未配置	solor[redacted]	solon[redacted]	编辑 删除 日志
ct无过滤条件	-	系统模版	一致性: 字段数据相关性	name (string)	未配置	solon[redacted]	solon[redacted]	编辑 删除 日志

2. Click **Monitor Multiple Fields**, then click the **Add Fields** button to start adding the fields that need to be monitored.



选择监控对象

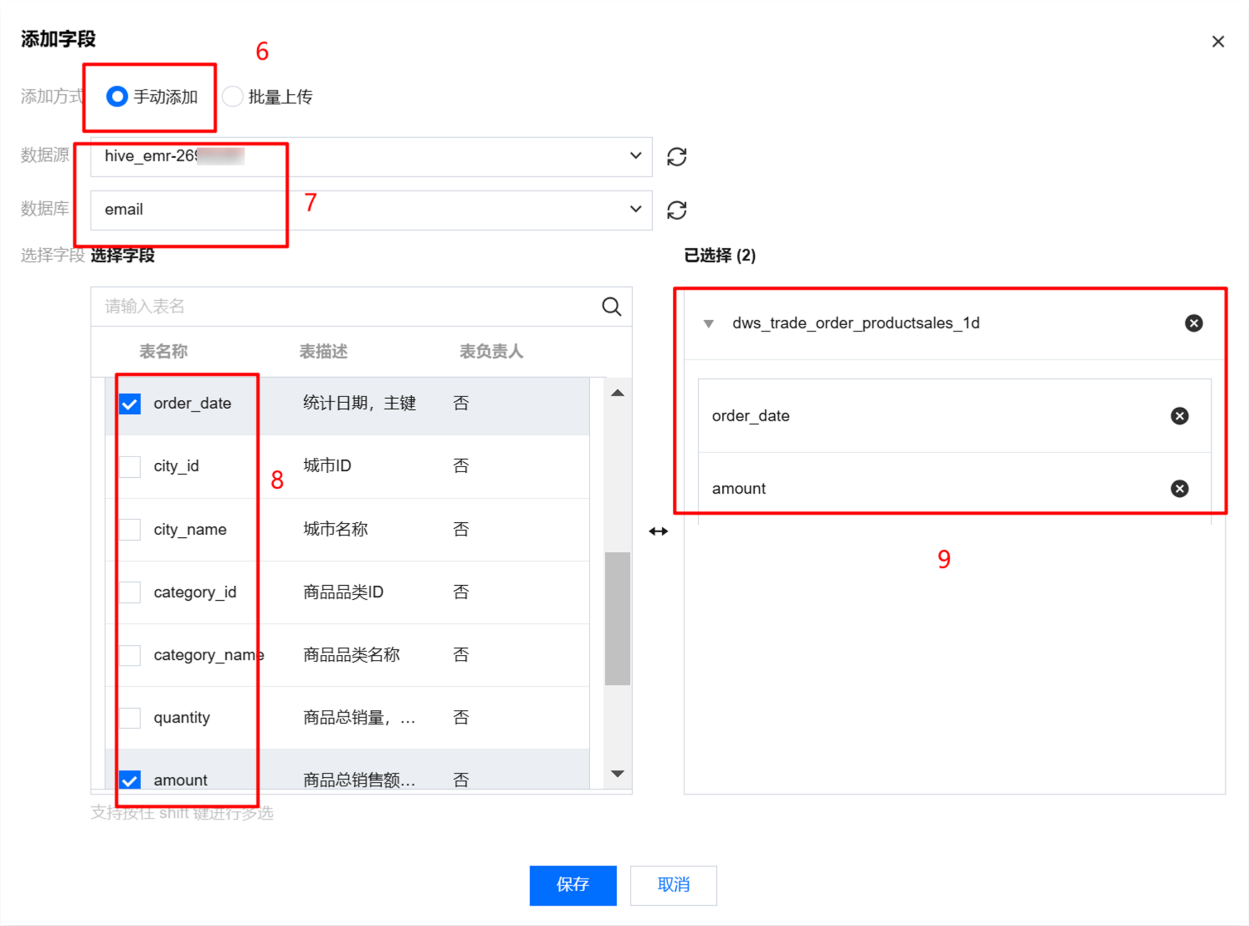
监控字段数量 0

○ 监控多表 用来监控表的数据质量

● 监控多字段 用来监控字段的数据质量

表名	表描述	表负责人	类型	数据源	数据库

3. On the add field page, select the add method as **manually add**, select **data source** as hive_emr-XXX, select **database** as hive database emall. After completion of the selection, tables and fields below will be updated. Select table dwd_trade_order_ordercreate_productsales and corresponding fields amount, order_date, and click **save**.



添加方式 ● 手动添加 ○ 批量上传

数据源 hive_emr-26[redacted]

数据库 email

选择字段 选择字段

请输入表名

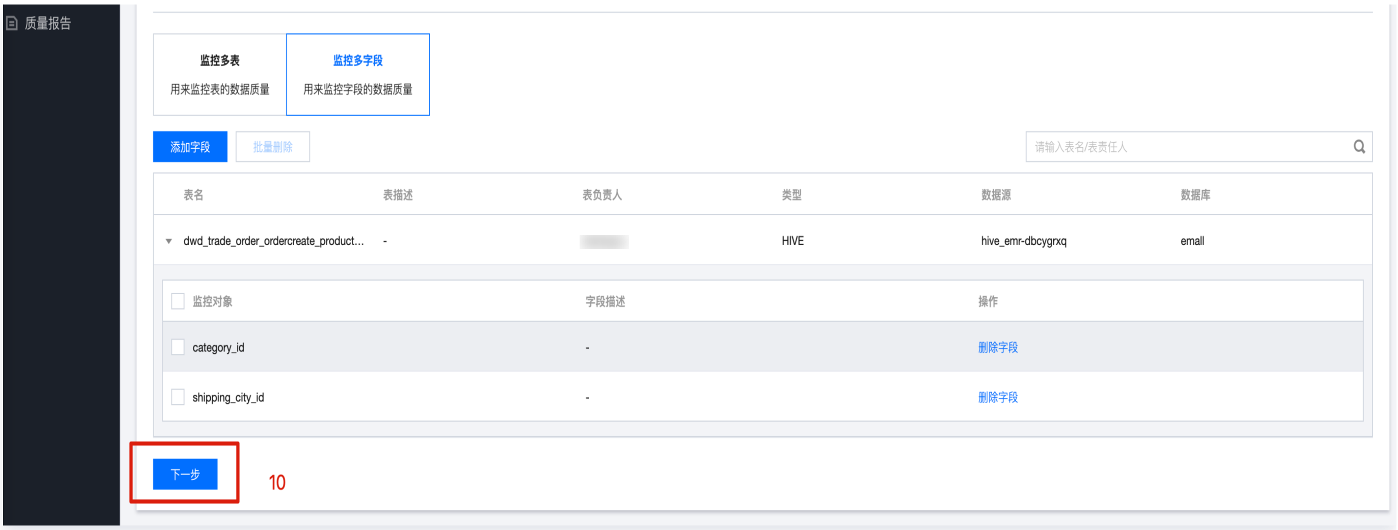
表名称	表描述	表负责人
☑ order_date	统计日期，主键	否
□ city_id	城市ID	否
□ city_name	城市名称	否
□ category_id	商品品类ID	否
□ category_name	商品品类名称	否
□ quantity	商品总销量，...	否
☑ amount	商品总销售额...	否

支持按住 shift 键进行多选

已选择 (2)

- dws_trade_order_productsales_1d
  - order_date
  - amount

4. After successful saving, the page will refresh to display the selected tables and fields. Click the **Next** button to start configuring monitoring rules.

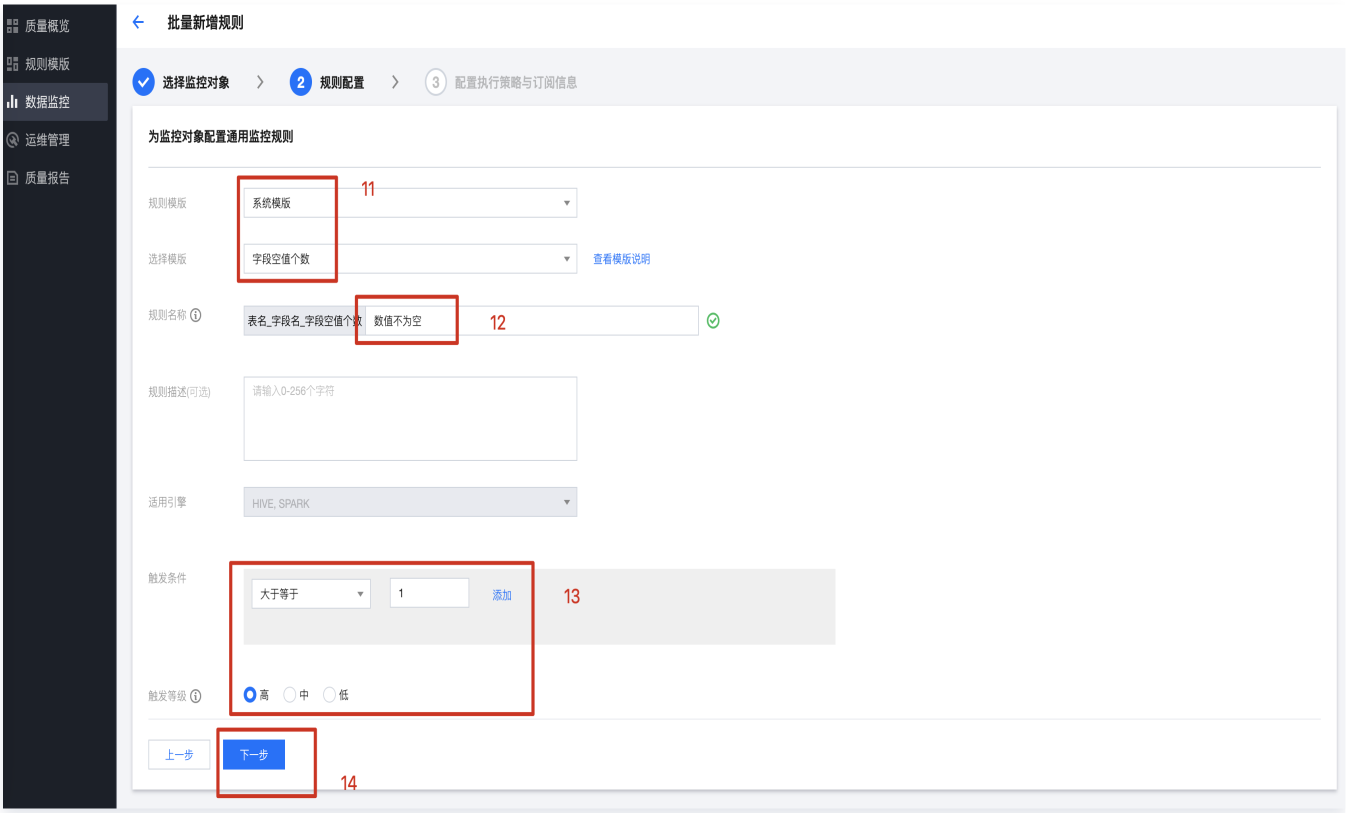


表名	表描述	表负责人	类型	数据源	数据库
dwd_trade_order_ordercreate_product...	-	[redacted]	HIVE	hive_emr-dbcygrxq	email

监控对象	字段描述	操作
category_id	-	删除字段
shipping_city_id	-	删除字段

Step 2: Configure Monitoring Rules

1. Configure monitoring rules for the previously selected tables and fields. Select **rule template** as system template, **select template** as number of empty fields, fill in **rule name** as value not empty, set **trigger condition** as equal to or greater than 1, set **trigger level** as high. After the settings are completed, click **next** button to start configuring execution policies and subscription information.
   - Rule template: WeData has already built-in more than 50 system templates, which can be used directly here.
   - Select a template: On the right, you can **view template descriptions.**
   - Trigger condition: means that when the count of empty values is equal to or greater than 1, immediately interrupt downstream tasks and send an alarm.



为监控对象配置通用监控规则

规则模版：系统模版

选择模版：字段空值个数　查看模版说明

规则名称：表名_字段名_字段空值个数_数值不为空

规则描述(可选)：请输入0-256个字符

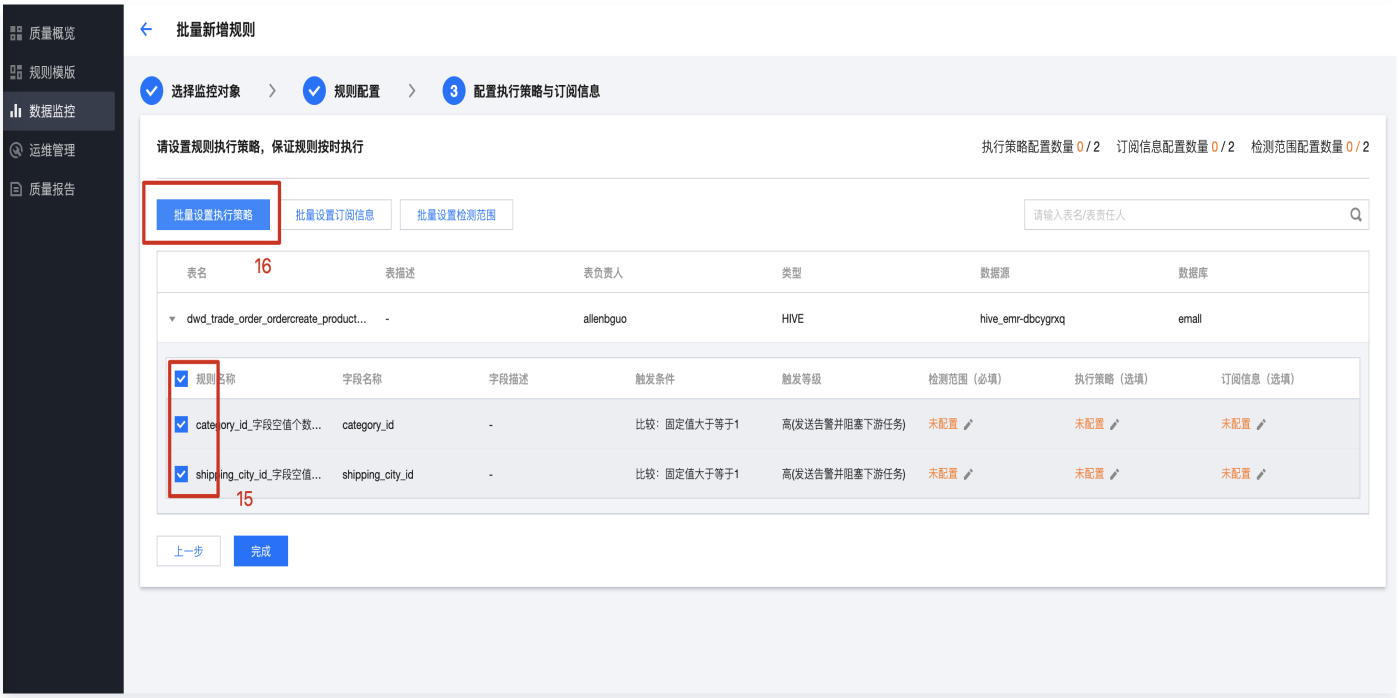
适用引擎：HIVE, SPARK

触发条件：大于等于 1　添加

触发等级：● 高　○ 中　○ 低

Procedure 3: Set Execution Policy

1. Click **Rule Name**, batch–select all rules, and then click **Batch Set Execution Strategy** button to configure execution policies.



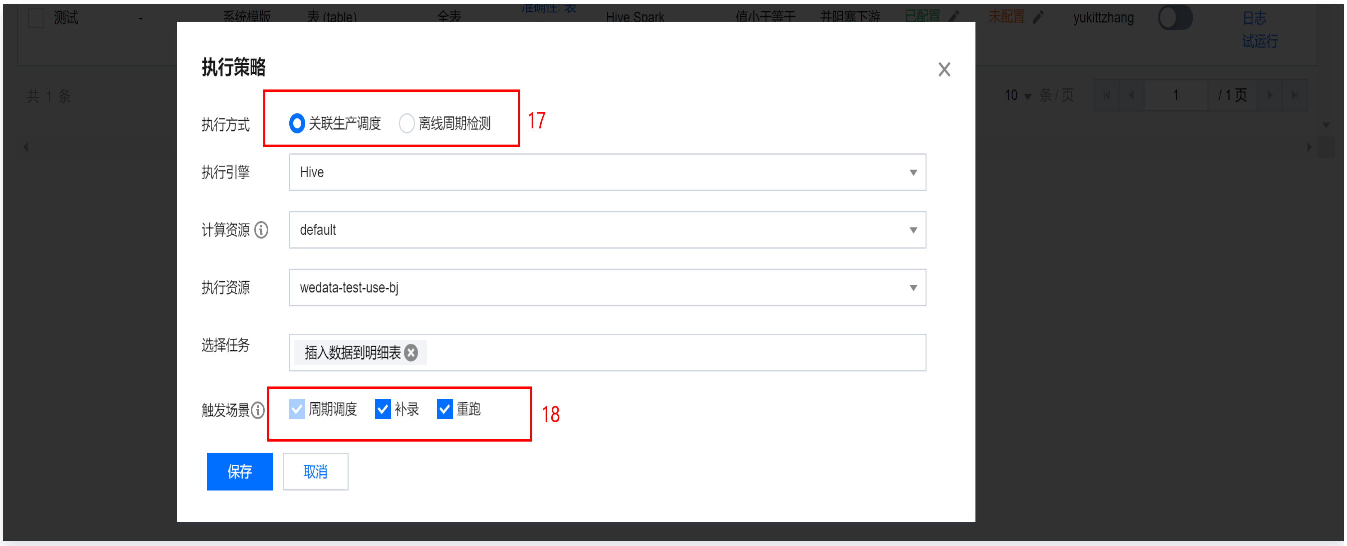
请设置规则执行策略，保证规则按时执行

执行策略配置数量 0 / 2 订阅信息配置数量 0 / 2 检测范围配置数量 0 / 2

表名	表描述	表负责人	类型	数据源	数据库
dwd_trade_order_ordercreate_product...	-	allenbguo	HIVE	hive_emr-dbcygrxq	email

☑ 规则名称	字段名称	字段描述	触发条件	触发等级	检测范围（必填）	执行策略（选填）	订阅信息（选填）
☑ category_id_字段空值个数...	category_id	-	比较：固定值大于等于1	高(发送告警并阻塞下游任务)	未配置	未配置	未配置
☑ shipping_city_id_字段空值...	shipping_city_id	-	比较：固定值大于等于1	高(发送告警并阻塞下游任务)	未配置	未配置	未配置

2. **Select Execution Mode** as Associated Production Scheduling, **Select Task** as Insert Data into Detail Table, and click **Save** button after the settings are completed.
   - Associate with production scheduling: means associating quality tasks with data development tasks. Only after the associated tasks are completed will this quality monitoring task be executed. Since we select **Insert data into detail table** here, data integrity will be detected immediately after data is inserted into the detail table.
   - The execution engine, computational resource, and execution resource are consistent with the selection in the above text.
   - Select task: the data development task that needs to be associated.



执行策略

执行方式 ● 关联生产调度 ○ 离线周期检测

执行引擎 Hive

计算资源 default

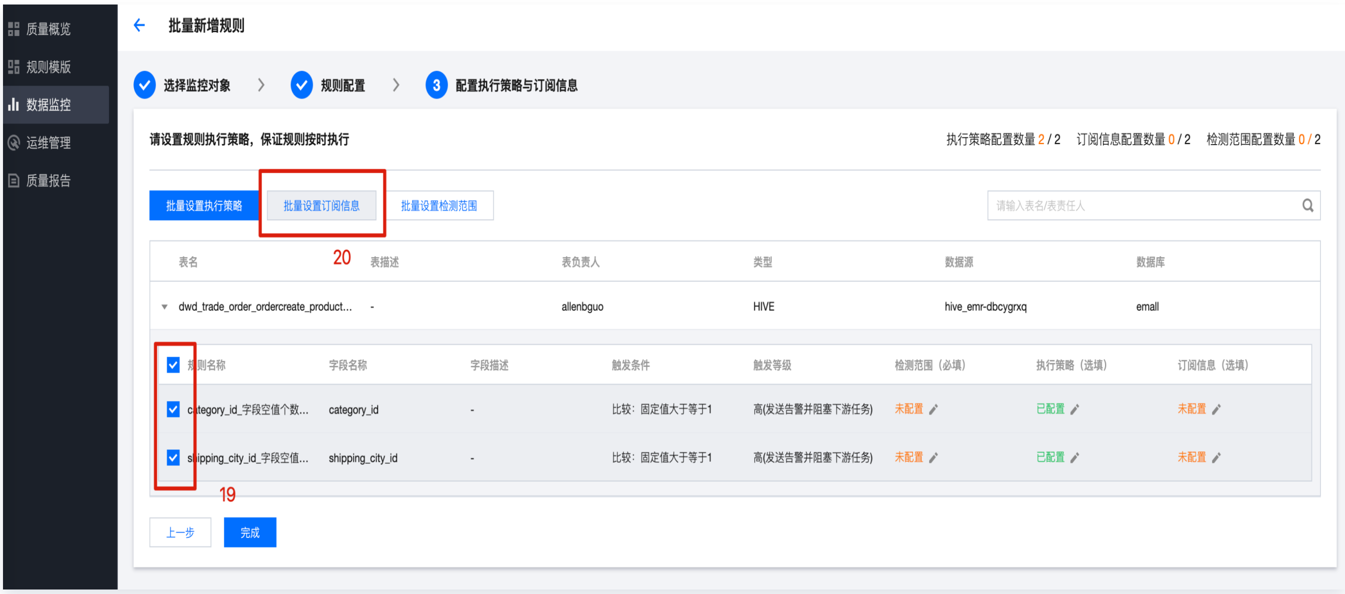
执行资源 wedata-test-use-bj

选择任务 插入数据到明细表

触发场景 ☑ 周期调度 ☑ 补录 ☑ 重跑

Procedure 4: Set Subscription Notifications

1. Click **rule name**, batch-select all rules, and click **Batch Set Subscription Information** button to configure subscription information.
   - Set up subscription notifications: Decide what method will be used to send a message reminder when an exception is detected.



批量新增规则

选择监控对象 > 规则配置 > 3 配置执行策略与订阅信息

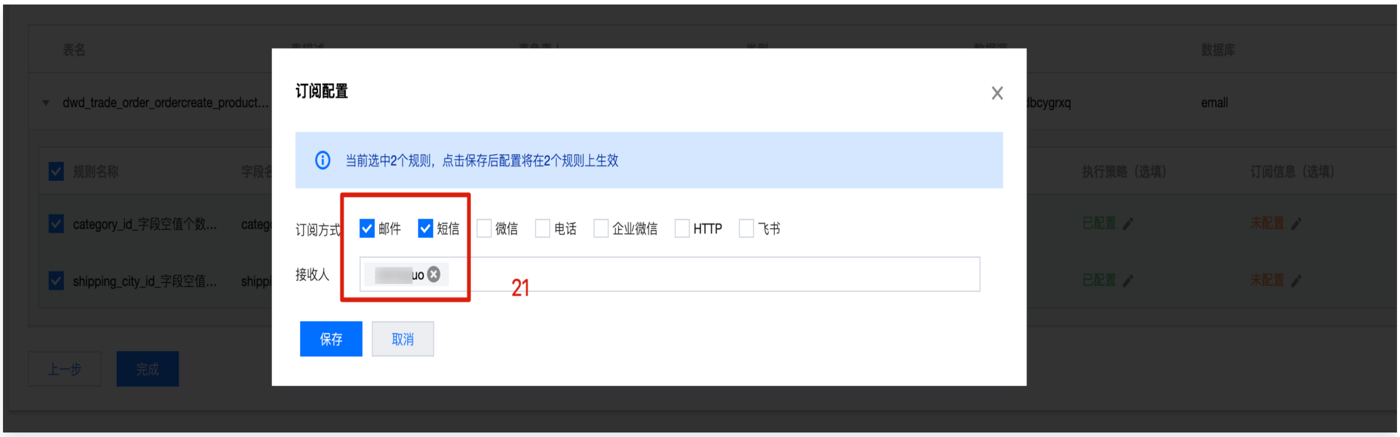
请设置规则执行策略，保证规则按时执行

执行策略配置数量 2 / 2 订阅信息配置数量 0 / 2 检测范围配置数量 0 / 2

表名	表描述	表负责人	类型	数据源	数据库
dwd_trade_order_ordercreate_product...	-	allenbguo	HIVE	hive_emr-dbcygrxq	emall

☑ 规则名称	字段名称	字段描述	触发条件	触发等级	检测范围（必填）	执行策略（选填）	订阅信息（选填）
☑ category_id_字段空值个数...	category_id	-	比较：固定值大于等于1	高(发送告警并阻塞下游任务)	未配置	已配置	未配置
☑ shipping_city_id_字段空值...	shipping_city_id	-	比较：固定值大于等于1	高(发送告警并阻塞下游任务)	未配置	已配置	未配置

2. **Subscription Configuration:** Select **email** and **Short Message Service**, and select the recipient as XXX.



订阅配置

当前选中2个规则，点击保存后配置将在2个规则上生效

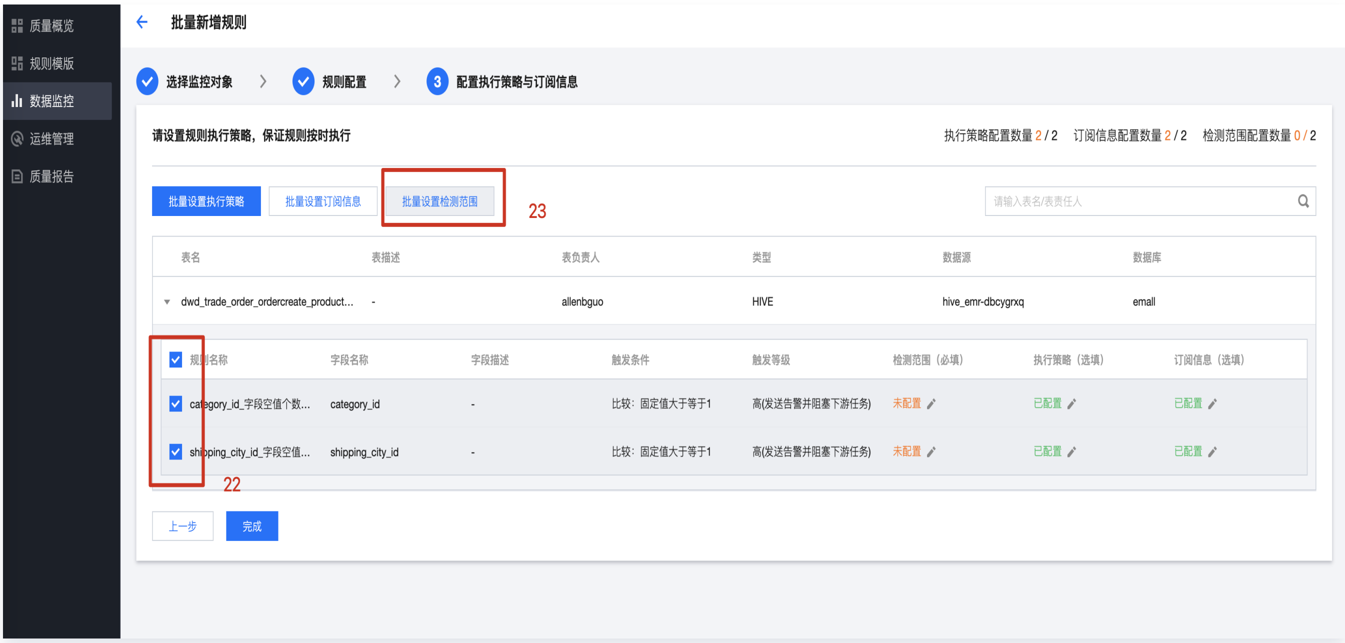
订阅方式 ☑ 邮件 ☑ 短信 □ 微信 □ 电话 □ 企业微信 □ HTTP □ 飞书

接收人 [redacted]uo

表名
dwd_trade_order_ordercreate_product...
dbcygrxq
数据库
email
☑ 规则名称
category_id_字段空值个数...
shipping_city_id_字段空值...
执行策略（选填）
订阅信息（选填）
已配置
未配置
上一步
完成
保存
取消
21

Procedure 5: Set Detection Scope

1. Click **Rule Name**, batch-select all rules, and then click **Batch Set Detection Scope** button to configure the detection scope.
   - ○ Set detection scope: Set which data scope to detect.



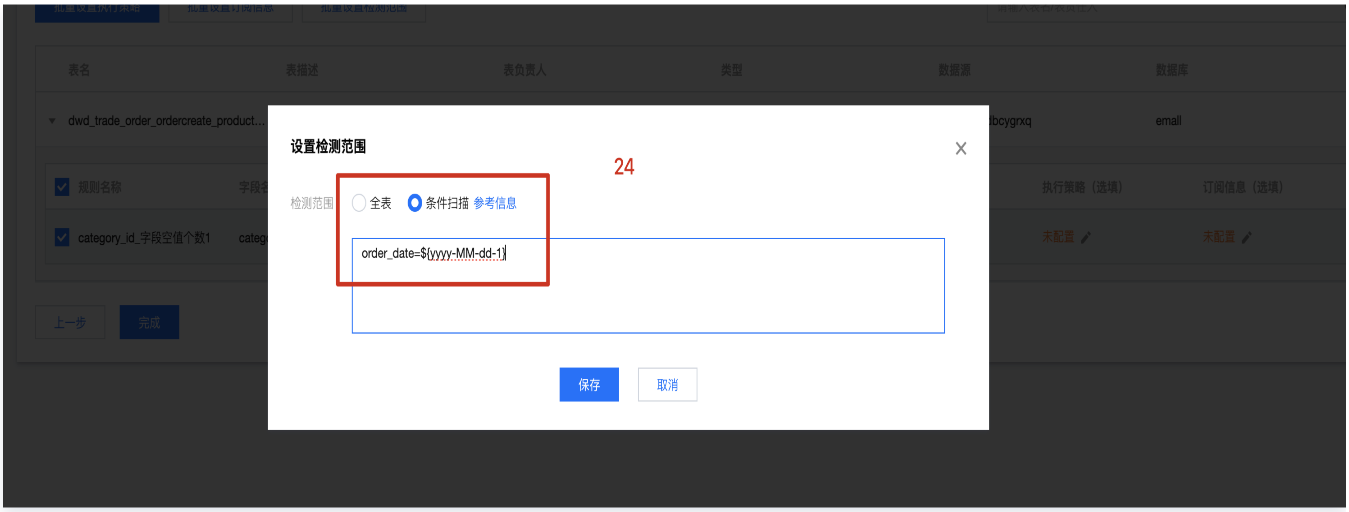
请设置规则执行策略，保证规则按时执行

执行策略配置数量 2 / 2　订阅信息配置数量 2 / 2　检测范围配置数量 0 / 2

表名	表描述	表负责人	类型	数据源	数据库
dwd_trade_order_ordercreate_product...	-	allenbguo	HIVE	hive_emr-dbcygrxq	email

☑ 规则名称	字段名称	字段描述	触发条件	触发等级	检测范围（必填）	执行策略（选填）	订阅信息（选填）
☑ category_id_字段空值个数...	category_id	-	比较：固定值大于等于1	高(发送告警并阻塞下游任务)	未配置	已配置	已配置
☑ shipping_city_id_字段空值...	shipping_city_id	-	比较：固定值大于等于1	高(发送告警并阻塞下游任务)	未配置	已配置	已配置

2. Set **Detection Scope** as conditional scanning, fill in content: order_date = ${yyyy-MM-dd-1}, and click **Save** button.
   - ○ Conditional scanning: Only check the newly produced data of each day according to the filled-in conditions, rather than performing a full check every day. Because the larger the amount of monitoring data, the larger the resource consumption.
   - ○ Related reference information can be viewed on the right.

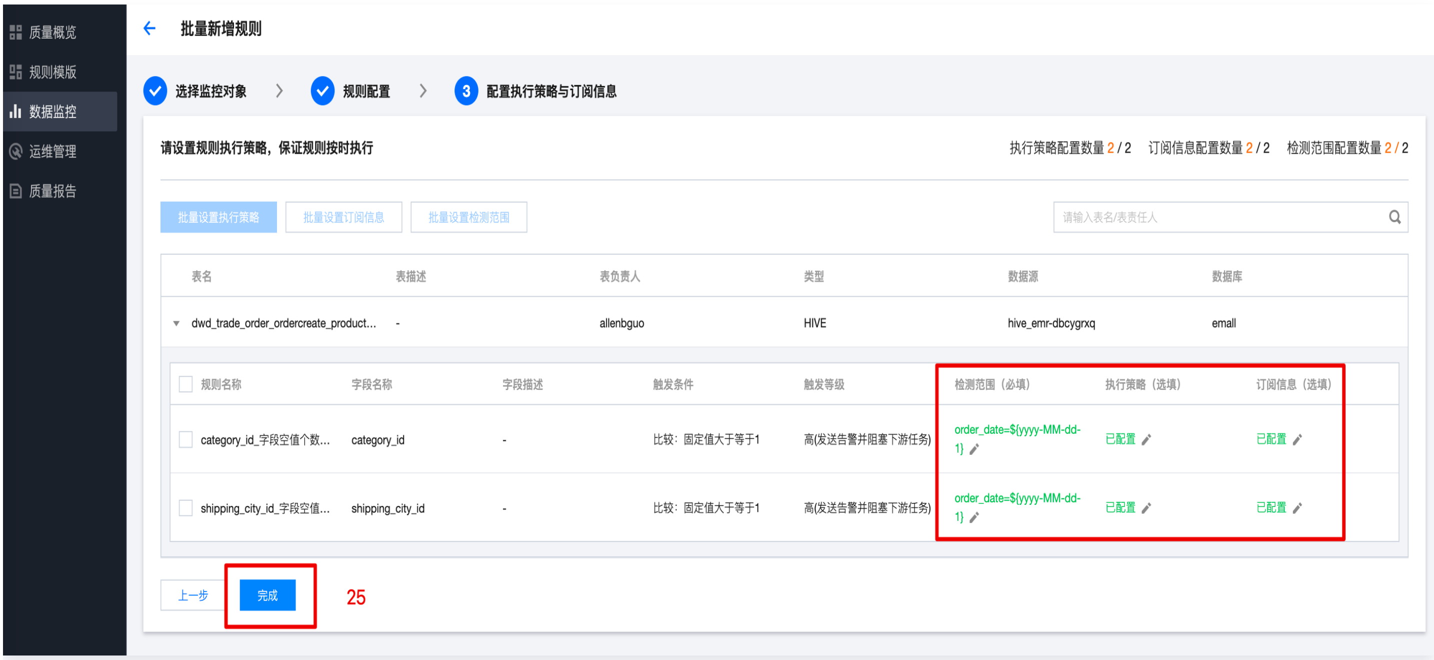


设置检测范围

检测范围 ○ 全表 ● 条件扫描 参考信息

order_date=${yyyy-MM-dd-1}

3. The above indicates that the configuration is all completed. Click **Complete** button.



批量新增规则

✓ 选择监控对象 > ✓ 规则配置 > 3 配置执行策略与订阅信息

请设置规则执行策略，保证规则按时执行

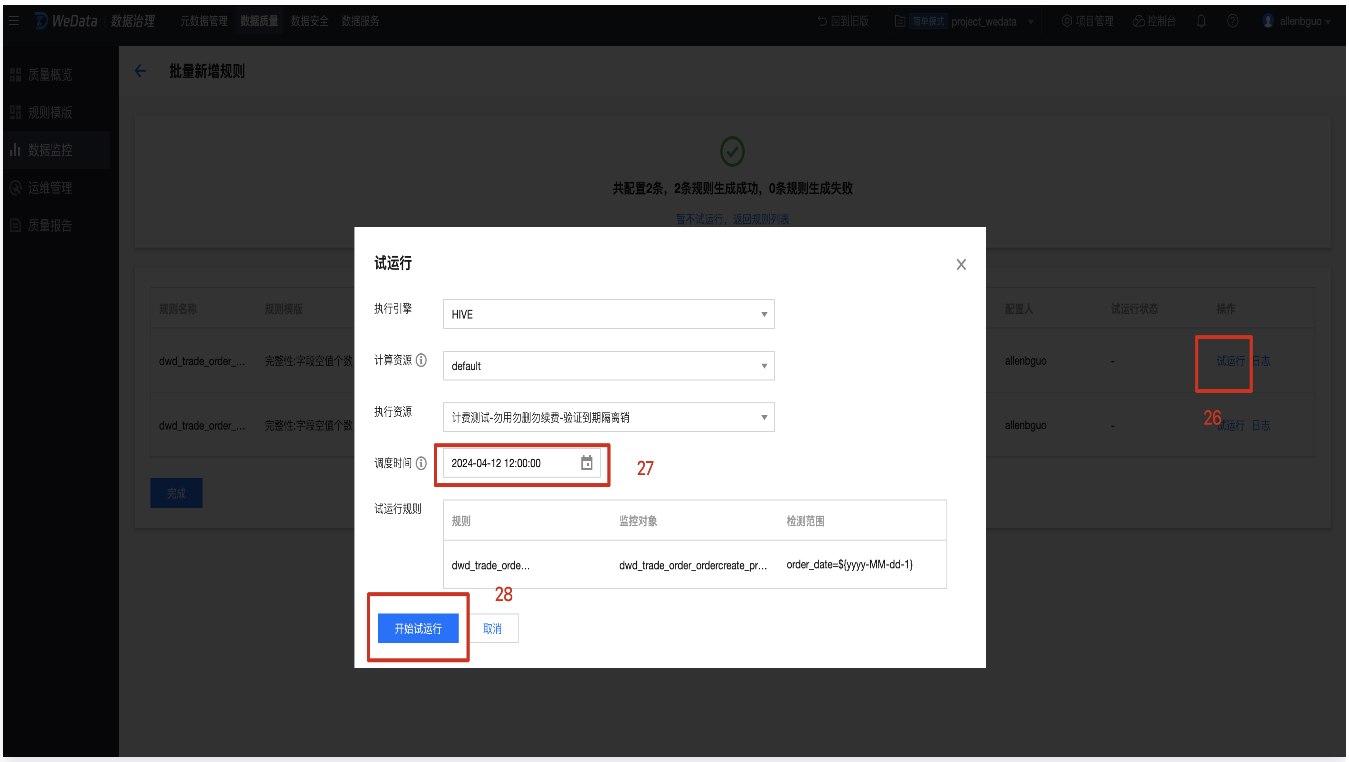
执行策略配置数量 2 / 2　订阅信息配置数量 2 / 2　检测范围配置数量 2 / 2

表名	表描述	表负责人	类型	数据源	数据库
dwd_trade_order_ordercreate_product...	-	allenbguo	HIVE	hive_emr-dbcygrxq	email

□ 规则名称	字段名称	字段描述	触发条件	触发等级	检测范围（必填）	执行策略（选填）	订阅信息（选填）
□ category_id_字段空值个数...	category_id	-	比较：固定值大于等于1	高(发送告警并阻塞下游任务)	order_date=${yyyy-MM-dd-1}	已配置	已配置
□ shipping_city_id_字段空值...	shipping_city_id	-	比较：固定值大于等于1	高(发送告警并阻塞下游任务)	order_date=${yyyy-MM-dd-1}	已配置	已配置

Procedure 6: Task Trial Run

1. Click **Trial Run** button on the right side of the page to configure, select the scheduling time as the trial run time, and click **Start Trial Run** button again.
    - execution engine, computational resource and execution resource should be consistent with the above context. **Note network connectivity**.
    - Trial run: Before releasing the task, you can run it once to detect whether there are errors in the task.
    - Start trial run: Just wait for pending execution and monitoring.
        - During the trial run, it should be ensured that there is data in the detailed table. If the trial run has been performed during both data integration and data development processes, there should be data in the detailed table.



共配置2条，2条规则生成成功，0条规则生成失败

规则名称	规则模版	配置人	试运行状态	操作
dwd_trade_order_...	完整性:字段空值个数	allenbguo	-	试运行 日志
dwd_trade_order_...	完整性:字段空值个数	allenbguo	-	试运行 日志

试运行

执行引擎: HIVE

计算资源: default

执行资源: 计费测试-勿用勿删勿续费-验证到期隔离销

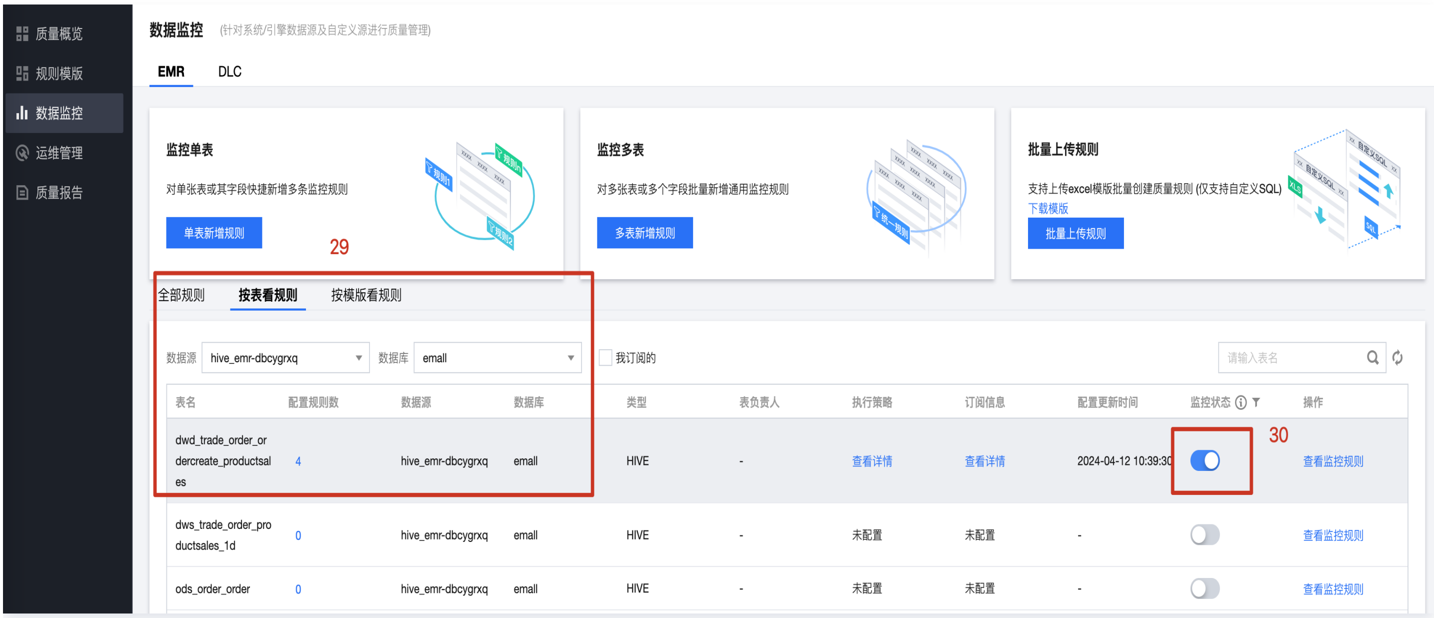
调度时间: 2024-04-12 12:00:00

试运行规则:

规则	监控对象	检测范围
dwd_trade_orde...	dwd_trade_order_ordercreate_pr...	order_date=${yyyy-MM-dd-1}

Procedure 7: Task Release

1. Return to the Data Monitoring page, select the **View Rules by Table** page, and click the **Monitoring Status** button to enable monitoring.
   - View Rules by Table: Filters out rules based on data source, database, and data table.



数据监控 (针对系统/引擎数据源及自定义源进行质量管理)

监控单表: 对单张表或其字段快捷新增多条监控规则

监控多表: 对多张表或多个字段批量新增通用监控规则

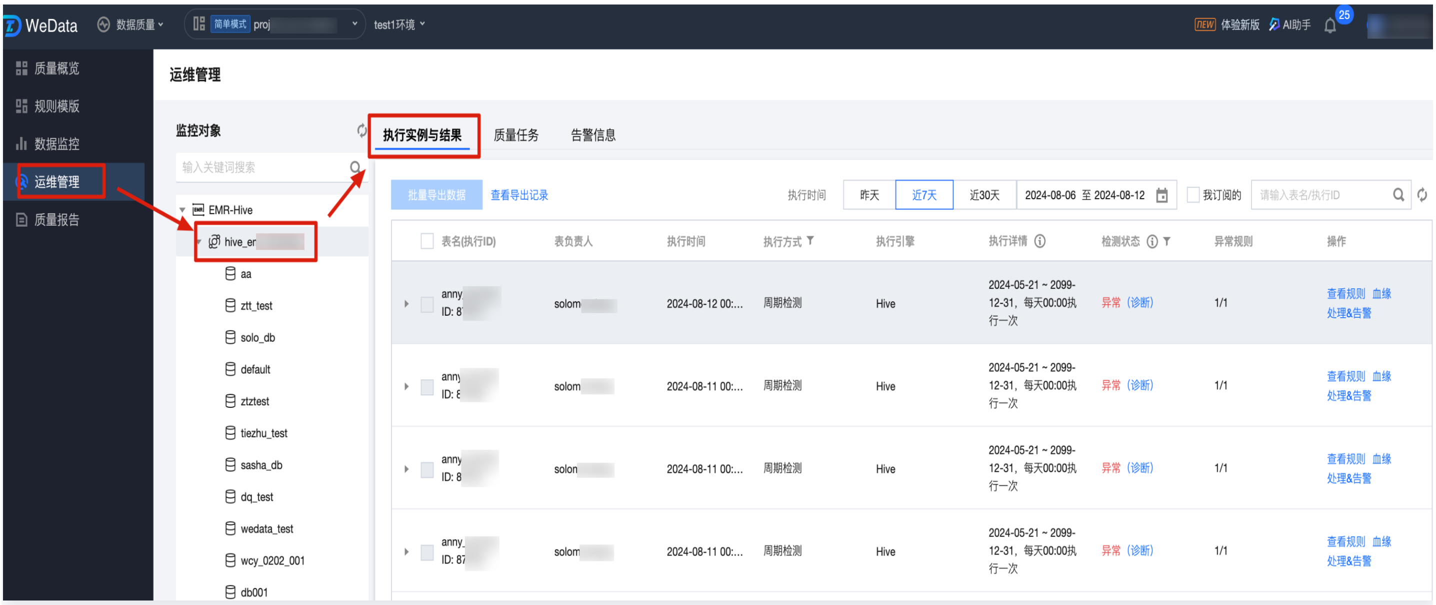
批量上传规则: 支持上传excel模版批量创建质量规则 (仅支持自定义SQL)

数据源: hive_emr-dbcygrxq 数据库: email

表名	配置规则数	数据源	数据库	类型	表负责人	执行策略	订阅信息	配置更新时间	监控状态	操作
dwd_trade_order_ordercreate_productsales	4	hive_emr-dbcygrxq	email	HIVE	-	查看详情	查看详情	2024-04-12 10:39:30	●	查看监控规则
dws_trade_order_productsales_1d	0	hive_emr-dbcygrxq	email	HIVE	-	未配置	未配置	-	○	查看监控规则
ods_order_order	0	hive_emr-dbcygrxq	email	HIVE	-	未配置	未配置	-	○	查看监控规则

Quality Monitoring Task Operation and Maintenance

You can view the task running result of the quality monitoring task in Ops management.



运维管理

执行时间：近7天　2024-08-06 至 2024-08-12

□ 我订阅的

表名(执行ID)	表负责人	执行时间	执行方式	执行引擎	执行详情	检测状态	异常规则	操作
anny[redacted] ID: 8[redacted]	solom[redacted]	2024-08-12 00:...	周期检测	Hive	2024-05-21 ~ 2099-12-31，每天00:00执行一次	异常（诊断）	1/1	查看规则 血缘 处理&告警
anny[redacted] ID: 8[redacted]	solom[redacted]	2024-08-11 00:...	周期检测	Hive	2024-05-21 ~ 2099-12-31，每天00:00执行一次	异常（诊断）	1/1	查看规则 血缘 处理&告警
anny[redacted] ID: 8[redacted]	solon[redacted]	2024-08-11 00:...	周期检测	Hive	2024-05-21 ~ 2099-12-31，每天00:00执行一次	异常（诊断）	1/1	查看规则 血缘 处理&告警
anny_[redacted] ID: 87[redacted]	solom[redacted]	2024-08-11 00:...	周期检测	Hive	2024-05-21 ~ 2099-12-31，每天00:00执行一次	异常（诊断）	1/1	查看规则 血缘 处理&告警

Data services

Last updated：2024-09-05 16:45:56

> **Note**
>
> The data service module is a **non-essential step** in the data development process. If your business does not involve this aspect, you **can skip** this module.
> If you need to experience this module's tutorial, please contact the enterprise administrator to complete the environment preparation.

According to the initial scheme design of the tutorial, the business side expects to understand the sales performance of different categories in different regions every day.
In Chapter Five, we have created a Mall summary table, which records product sales quantity and total sales amount by day, city, and category.
Regarding the data application of the Shopping Mall Detail Table, there are multiple ways:

1. Using SQL to perform direct data extraction and view daily sales performance is quite simple and will not be covered in this tutorial.
2. Displaying data report views using BI software (e.g., BI). For relevant documentation, please refer to the BI official website. This tutorial will not cover this method.
3. Using WeData data service features to produce API services for external systems to call. This part will be the focus of this tutorial.

Next, we will introduce how to use WeData data service features to produce API services for external systems to call.
Before developing the data service interface, we need to understand some technical information.

> **Note**
>
> - **Hive**'s feature is big data processing. It is mainly used for handling large-scale structured data, especially in data warehouse and big data analysis scenarios. However, its query speed is relatively slow, making it unsuitable for APIs that require rapid response. Therefore, it is generally not used as backend data storage for APIs. The summarized data computed from big data must be imported into an engine with better query performance before providing API queries.
> - **MySQL** can be used for applications that require rapid response and transaction support. It has fast query speed and supports complex queries and transaction processing.
>
> Therefore, we use MySQL as the underlying data storage for the API.

Next, we need to transfer the computed data to the MySQL database.

Environment preparations

Role: Enterprise Administrator.

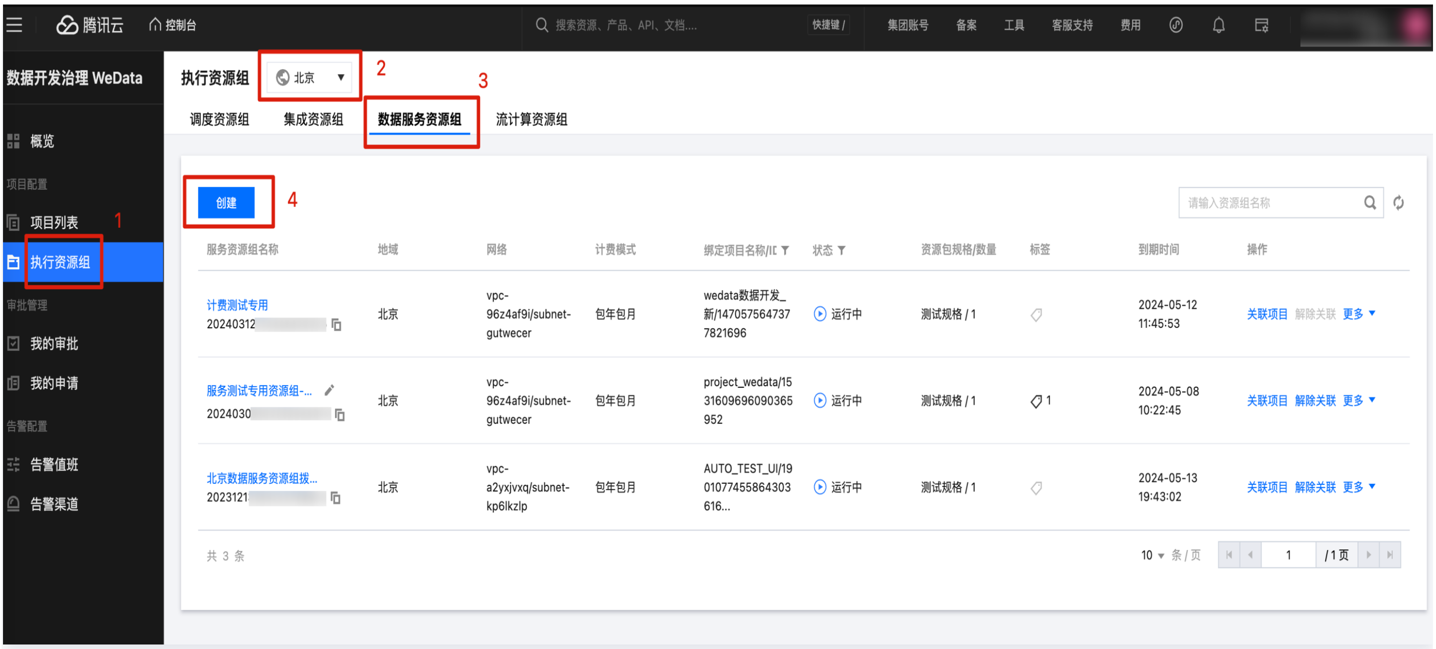
> **Note**

> The following operations (environment and resource preparation) involve resource purchase and paid content, which need to be performed by the Enterprise Administrator.

Purchase Data Service Resources

1. Go to the WeData Console and enter the Execution Resource Group page, select the region as **Beijing**, click **Data Service Resource Group** to enter the page, then click the **Create** button.

- Region: In this tutorial, select resources in the **Beijing** region.

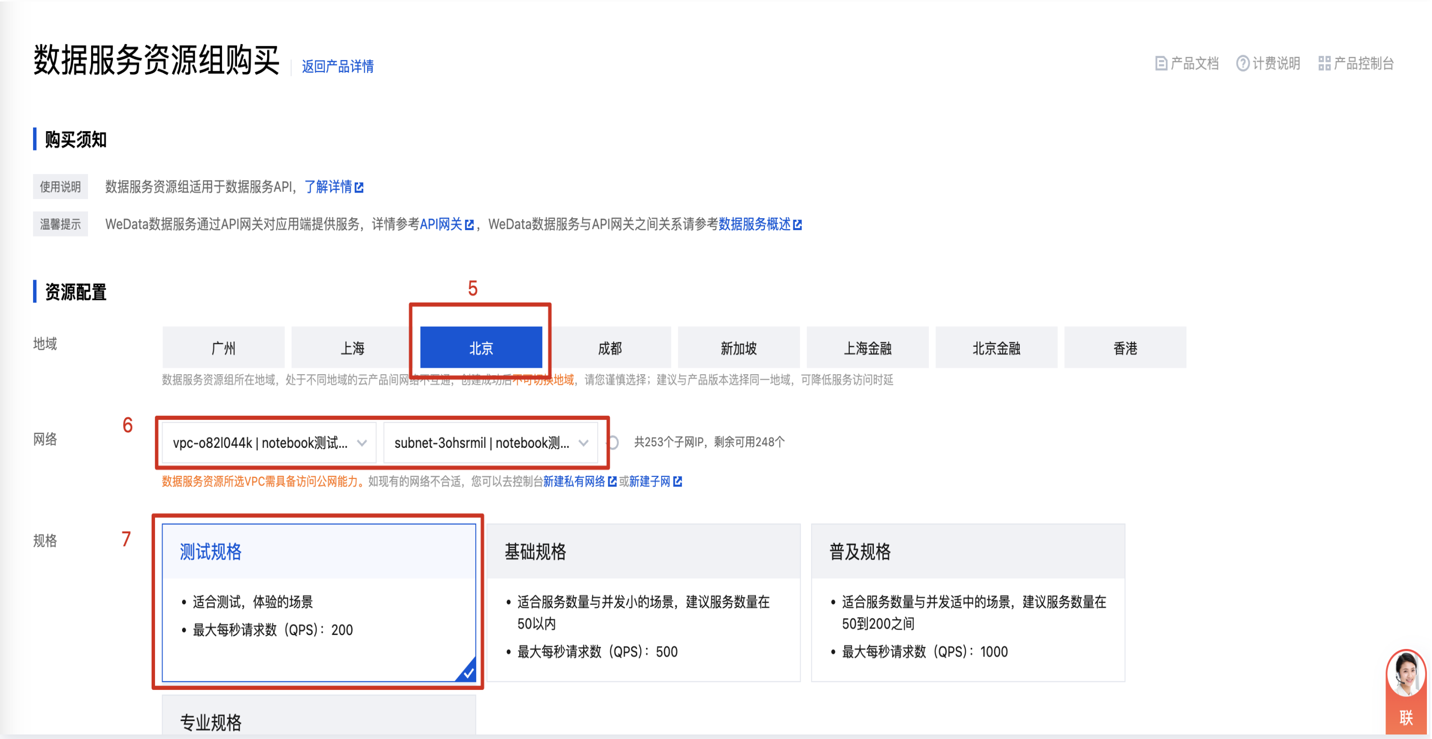


服务资源组名称	地域	网络	计费模式	绑定项目名称/ID	状态	资源包规格/数量	标签	到期时间	操作
计费测试专用 20240312[redacted]	北京	vpc-96z4af9i/subnet-gutwecer	包年包月	wedata数据开发_新/1470575647377821696	运行中	测试规格 / 1		2024-05-12 11:45:53	关联项目 解除关联 更多
服务测试专用资源组-... 2024030[redacted]	北京	vpc-96z4af9i/subnet-gutwecer	包年包月	project_wedata/1531609696090365952	运行中	测试规格 / 1	1	2024-05-08 10:22:45	关联项目 解除关联 更多
北京数据服务资源组拔... 2023121[redacted]	北京	vpc-a2yxjvxq/subnet-kp6lkzlp	包年包月	AUTO_TEST_UI/1901077455864303616...	运行中	测试规格 / 1		2024-05-13 19:43:02	关联项目 解除关联 更多

共 3 条　10 条 / 页　1 / 1 页

2. After clicking the **Create** button, you will enter the **Data Service Resource Group Purchase** page. In **Resource Configuration**, select the region as **Beijing**, select the **Network** as the VPC created in step Create a New VPC, and select the specifications as **Test Specifications**.

- Specifications: For detailed information about service resources, please see Service Resource Billing Explanation.



数据服务资源组购买 返回产品详情

购买须知

使用说明 数据服务资源组适用于数据服务API，了解详情

温馨提示 WeData数据服务通过API网关对应用端提供服务，详情参考API网关，WeData数据服务与API网关之间关系请参考数据服务概述

资源配置

地域 广州 上海 北京 成都 新加坡 上海金融 北京金融 香港

数据服务资源组所在地域，处于不同地域的云产品间网络不互通，创建成功后不可切换地域，请您谨慎选择；建议与产品版本选择同一地域，可降低服务访问时延

网络 vpc-o82l044k | notebook测试... subnet-3ohsrmil | notebook测... 共253个子网IP，剩余可用248个

数据服务资源所选VPC需具备访问公网能力。如现有的网络不合适，您可以去控制台新建私有网络或新建子网

规格

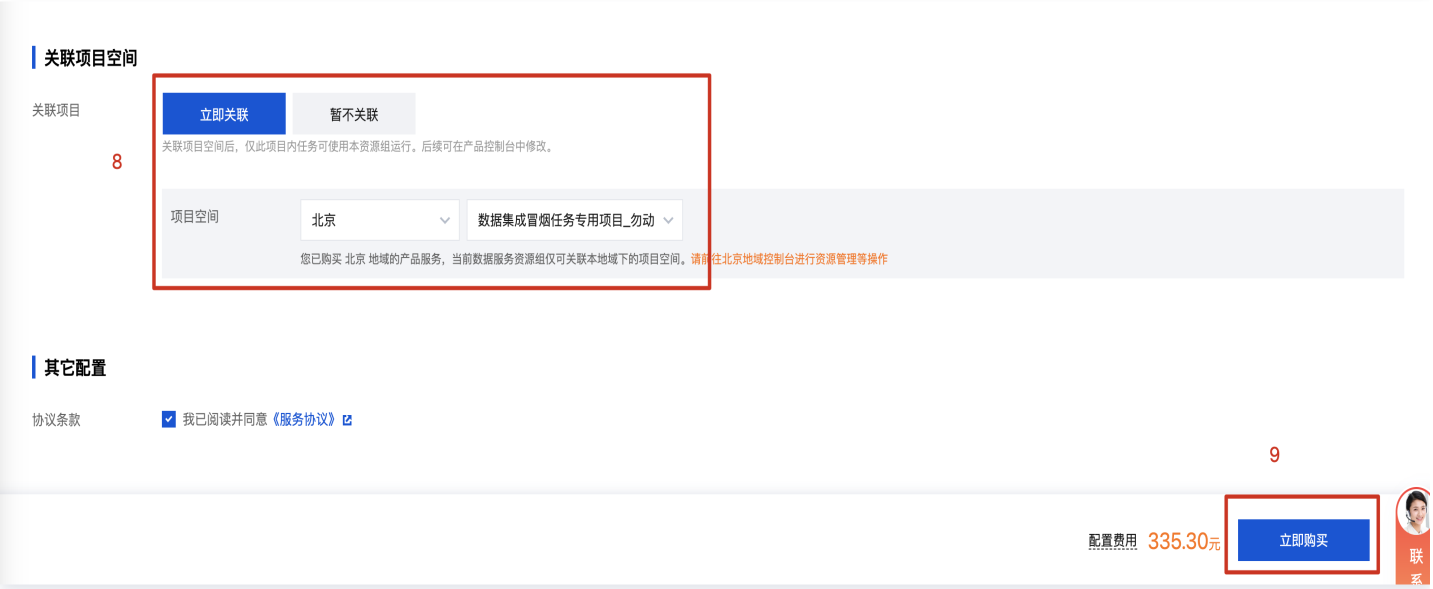
测试规格
- 适合测试，体验的场景
- 最大每秒请求数（QPS）：200

基础规格
- 适合服务数量与并发小的场景，建议服务数量在50以内
- 最大每秒请求数（QPS）：500

普及规格
- 适合服务数量与并发适中的场景，建议服务数量在50到200之间
- 最大每秒请求数（QPS）：1000

专业规格

3. In **Associated Project Space**, select **Associate Project** as Immediate Association, associate with the project created in step Create a Project in WeData, and click the **Purchase Now** button.



关联项目空间

关联项目 立即关联 暂不关联

关联项目空间后，仅此项目内任务可使用本资源组运行。后续可在产品控制台中修改。

项目空间 北京 数据集成冒烟任务专用项目_勿动

您已购买 北京 地域的产品服务，当前数据服务资源组仅可关联本地域下的项目空间。请前往北京地域控制台进行资源管理等操作

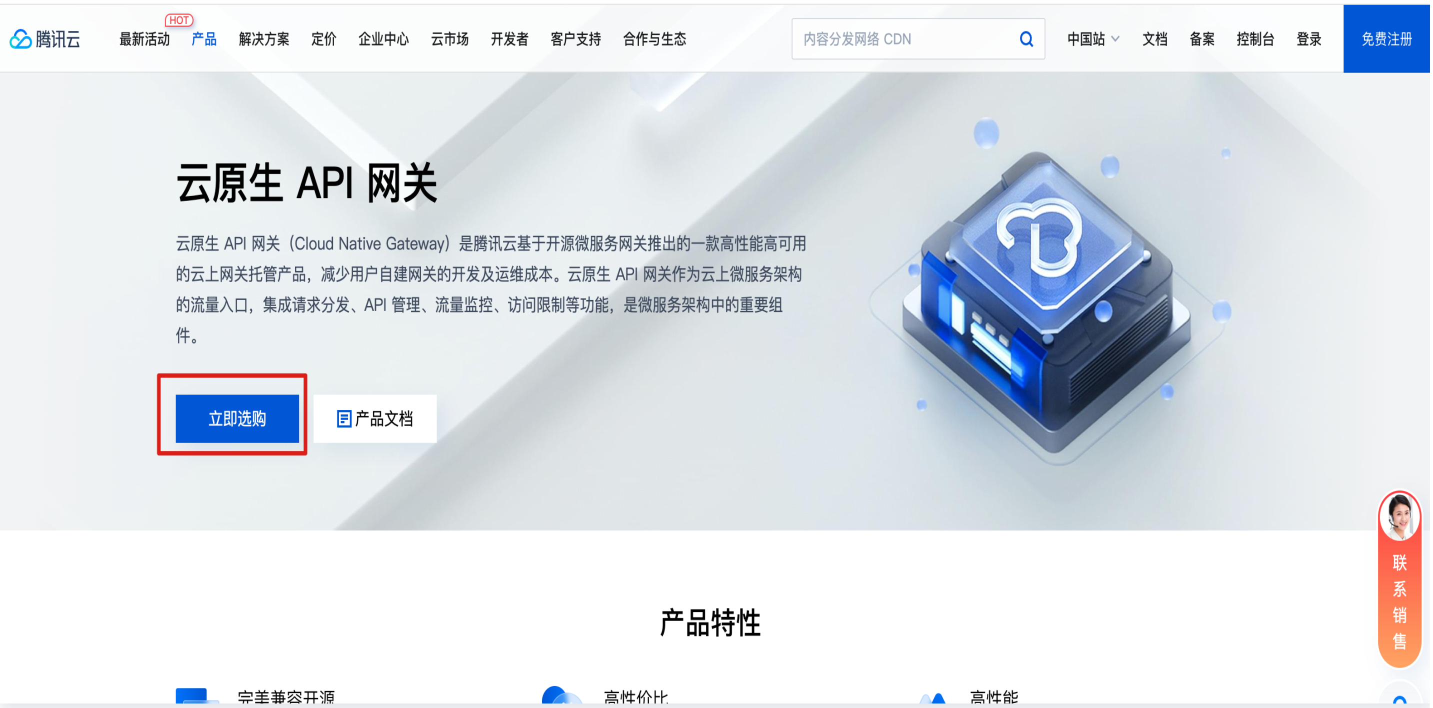
其它配置

协议条款 ☑ 我已阅读并同意《服务协议》

配置费用 335.30元 立即购买

Purchase API Gateway Resources

1. Go to the Tencent Cloud Official Website > Cloud Native Gateway. If you are using it for the first time, you will need to enable permissions by clicking the **Use Now** button.



云原生 API 网关

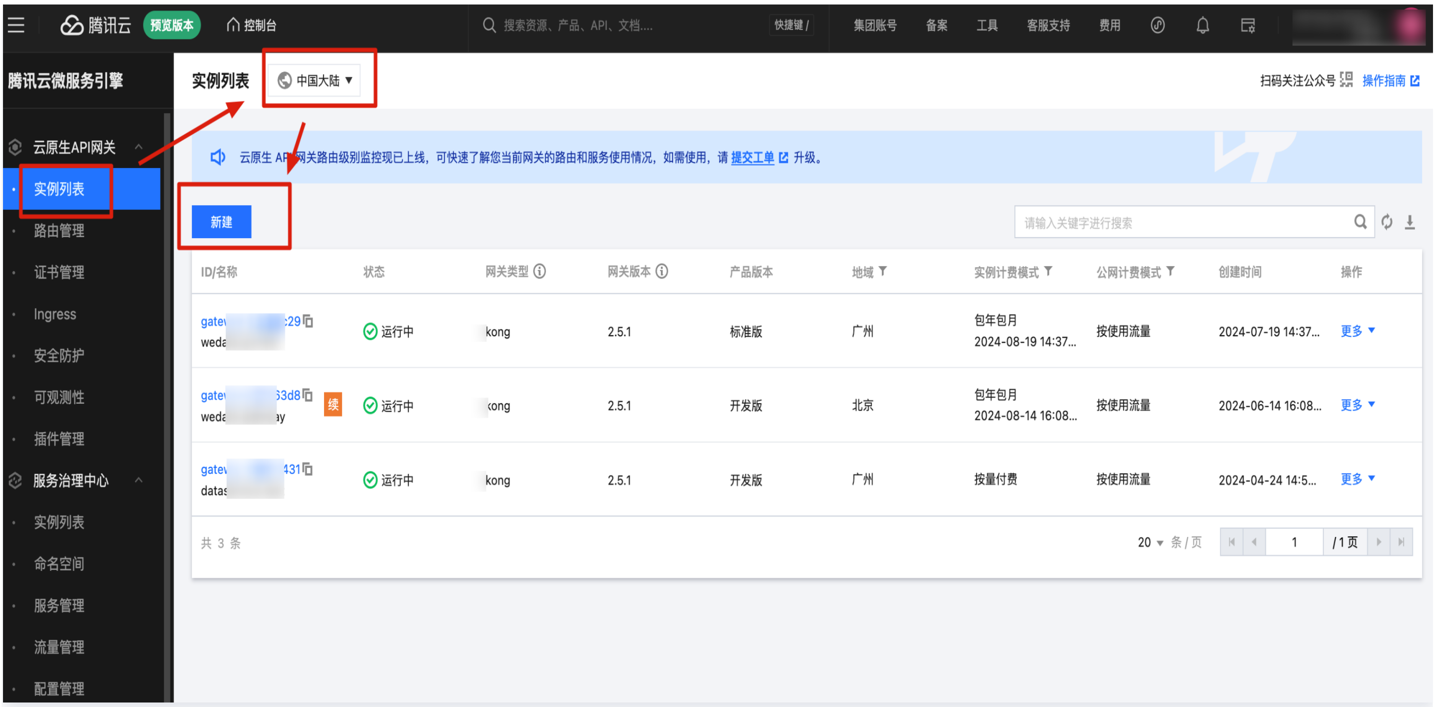
云原生 API 网关（Cloud Native Gateway）是腾讯云基于开源微服务网关推出的一款高性能高可用的云上网关托管产品，减少用户自建网关的开发及运维成本。云原生 API 网关作为云上微服务架构的流量入口，集成请求分发、API 管理、流量监控、访问限制等功能，是微服务架构中的重要组件。

立即选购　　产品文档

产品特性

完美兼容开源　　高性价比　　高性能

2. Log in to the Tencent Cloud Console > Instance List, go to the Gateway **Instance** list, select the region as **Beijing**, and click the **Create** button.



实例列表　中国大陆 ▾

云原生 API 网关路由级别监控现已上线，可快速了解您当前网关的路由和服务使用情况，如需使用，请 提交工单 升级。

新建

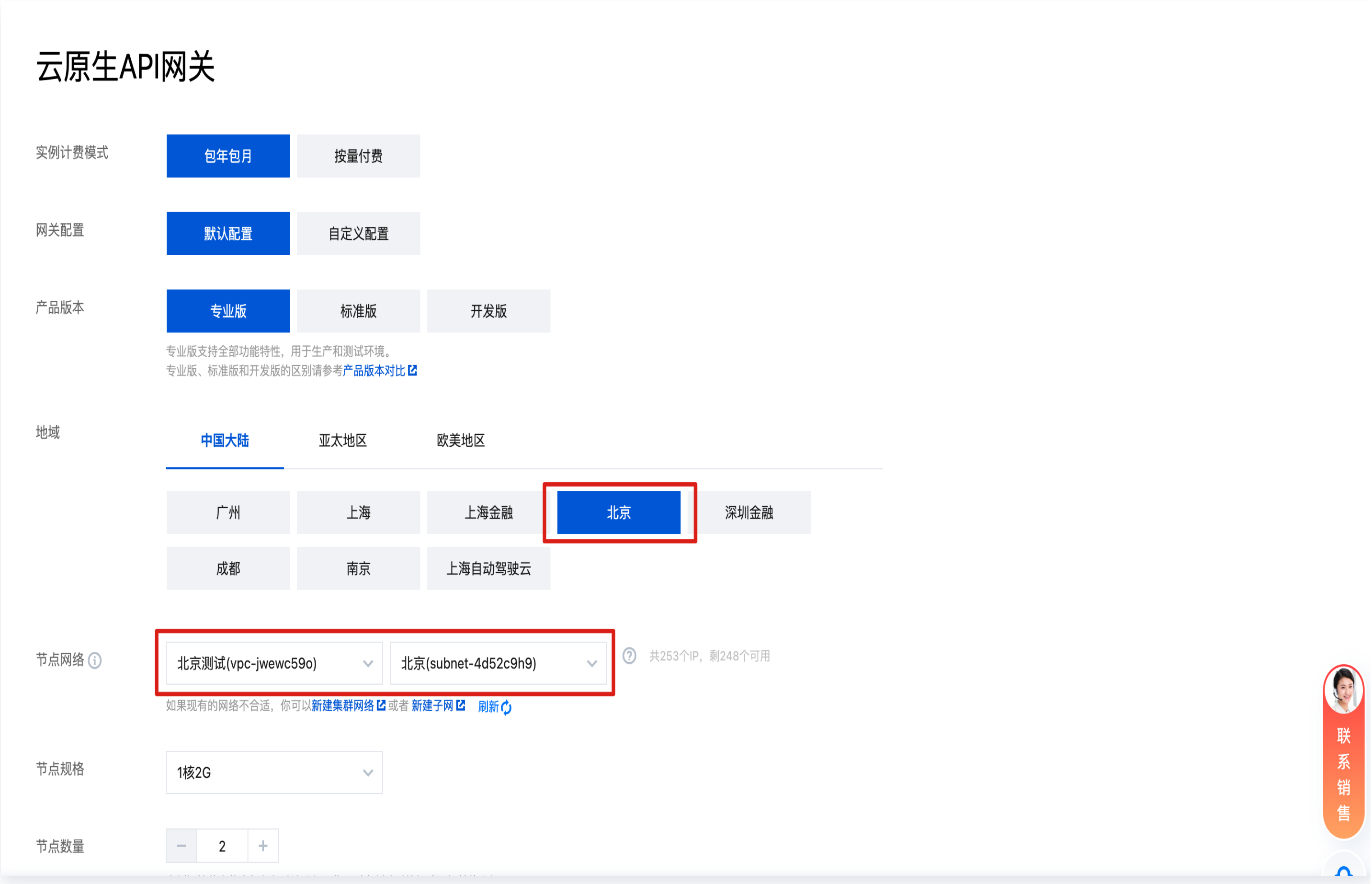
请输入关键字进行搜索

ID/名称	状态	网关类型	网关版本	产品版本	地域	实例计费模式	公网计费模式	创建时间	操作
gate[redacted]29 weda[redacted]	运行中	kong	2.5.1	标准版	广州	包年包月 2024-08-19 14:37...	按使用流量	2024-07-19 14:37...	更多 ▾
gate[redacted]3d8 续 weda[redacted]ay	运行中	kong	2.5.1	开发版	北京	包年包月 2024-08-14 16:08...	按使用流量	2024-06-14 16:08...	更多 ▾
gate[redacted]431 datas[redacted]	运行中	kong	2.5.1	开发版	广州	按量付费	按使用流量	2024-04-24 14:5...	更多 ▾

共 3 条　　20 ▾ 条 / 页　　1　/ 1 页

3. In the configuration selection, select the region as **Beijing**, select the instance specification as **Basic Edition**, and select the network as the VPC created in step Create a New VPC.

- Availability Zone: The availability zone where the subnet in the VPC is located.



云原生API网关

实例计费模式　包年包月　按量付费

网关配置　默认配置　自定义配置

产品版本　专业版　标准版　开发版

专业版支持全部功能特性，用于生产和测试环境。
专业版、标准版和开发版的区别请参考产品版本对比

地域　中国大陆　亚太地区　欧美地区

广州　上海　上海金融　北京　深圳金融
成都　南京　上海自动驾驶云

节点网络　北京测试(vpc-jwewc59o)　北京(subnet-4d52c9h9)　共253个IP，剩248个可用

如果现有的网络不合适，你可以新建集群网络或者 新建子网　刷新

节点规格　1核2G

节点数量　2

Create Summary Table

Create a data summary table in Tencent Cloud MySQL. For specific steps, see Summary Table Development.

The table creation SQL statement is as follows:

```
-- In MySQL, create a summary table
CREATE TABLE email.dws_trade_order_productsales_1d (
    order_date DATE NOT NULL,
    city_id INT NOT NULL,
    category_id INT NOT NULL,
    city_name VARCHAR(50) NOT NULL,
    category_name VARCHAR(50) NOT NULL,
    quantity INT NOT NULL,
    amount DECIMAL(10, 2) NOT NULL,
    pt_date VARCHAR(50) NOT NULL
);
```

Data Service Interface Design

Interface Input Parameters:

Field Name	Associated Fields	Mandatory or Not	Format
Date	order_date	Mandatory	Date
Region	city_id	Optional	City Code, INT
Category	category_id	Optional	Category Code, INT

Interface Output:

Here we output all fields of the data table.

Data Service Interface Development

Sync data to MySQL

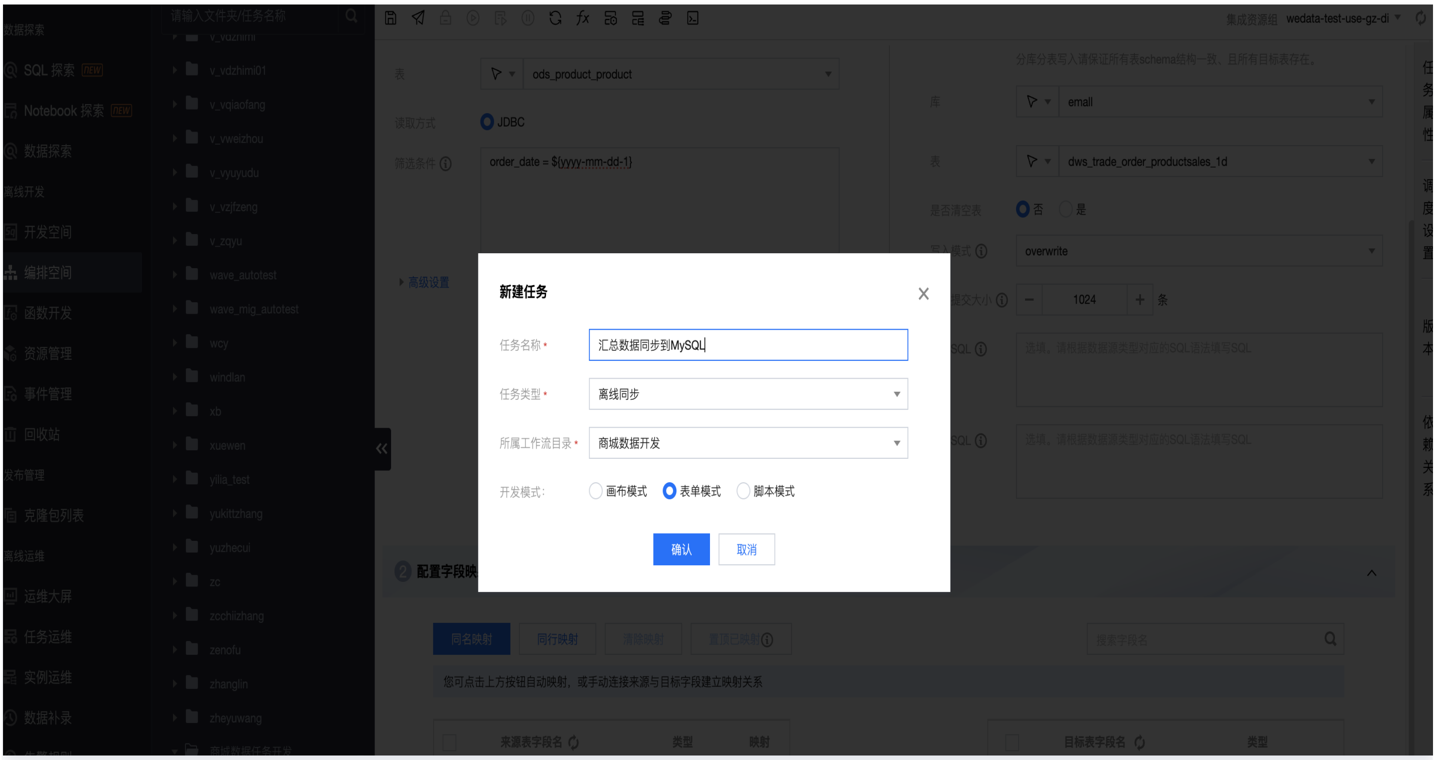
First, we need to sync the data from the Hive mall summary table to the MySQL summary table.

Step 1: Create a data synchronization task

> **Note**
> In the **WeData > Data Development > Mall Development** workflow, add an offline synchronization node. For detailed steps, refer to Offline Synchronization Task Development.

1. In the orchestration space, find the created **Mall Data Task Development** > **Mall Data Development** > click **Offline Synchronization**, select configuration mode, click **Confirm.**

- Task Name: Sync summary data to MySQL.
- Development Mode: Choose **Form Mode.**



新建任务

任务名称 *: 汇总数据同步到MySQL

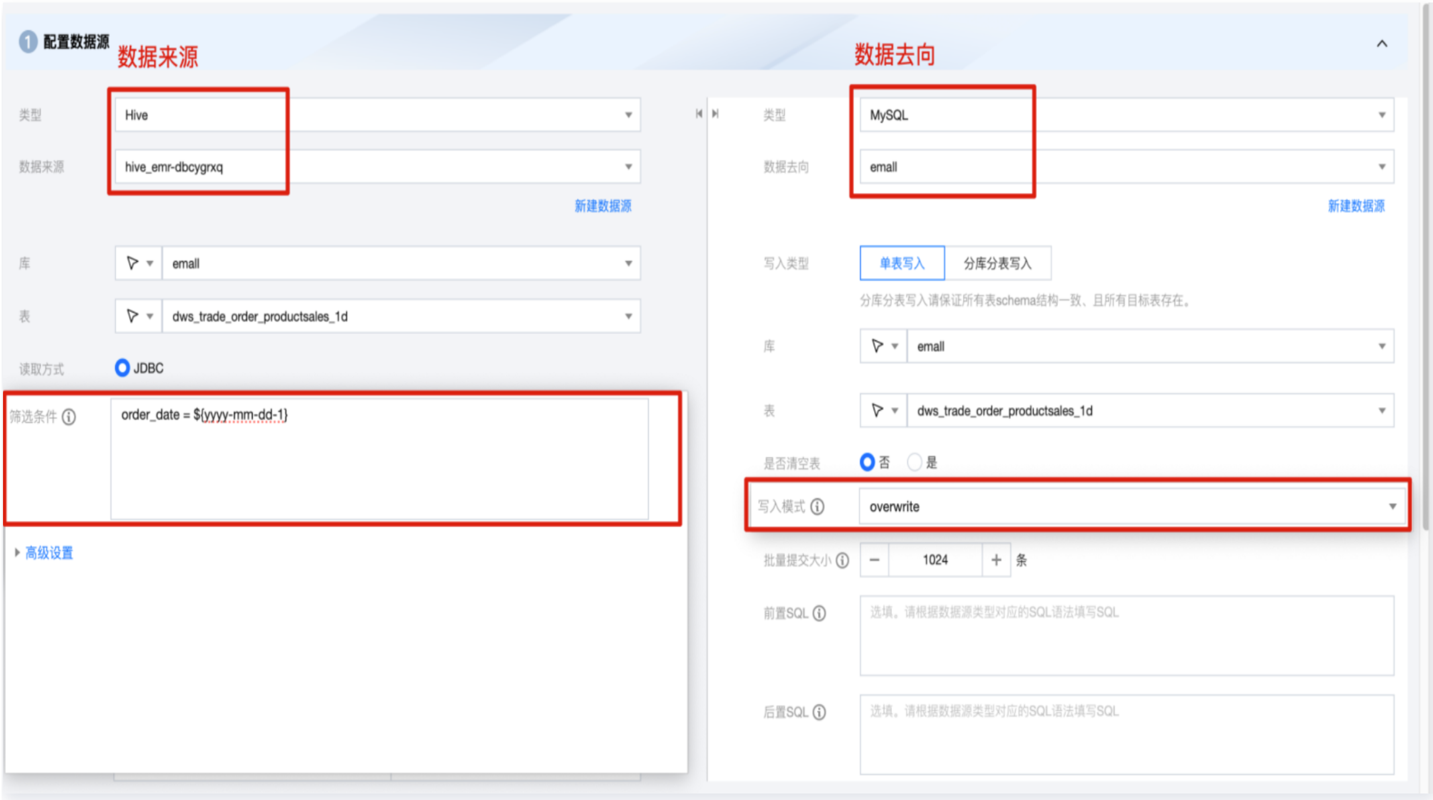
任务类型 *: 离线同步

所属工作流目录 *: 商城数据开发

开发模式: ○ 画布模式 ● 表单模式 ○ 脚本模式

确认 取消

2. Configure **Data Source** and **Data Destination** in order.



1 配置数据源

数据来源

类型: Hive

数据来源: hive_emr-dbcygrxq

新建数据源

库: email

表: dws_trade_order_productsales_1d

读取方式: ● JDBC

筛选条件: order_date = ${yyyy-mm-dd-1}

高级设置

数据去向

类型: MySQL

数据去向: email

新建数据源

写入类型: 单表写入 分库分表写入

分库分表写入请保证所有表schema结构一致、且所有目标表存在。

库: email

表: dws_trade_order_productsales_1d

是否清空表: ● 否 ○ 是

写入模式: overwrite

批量提交大小: − 1024 + 条

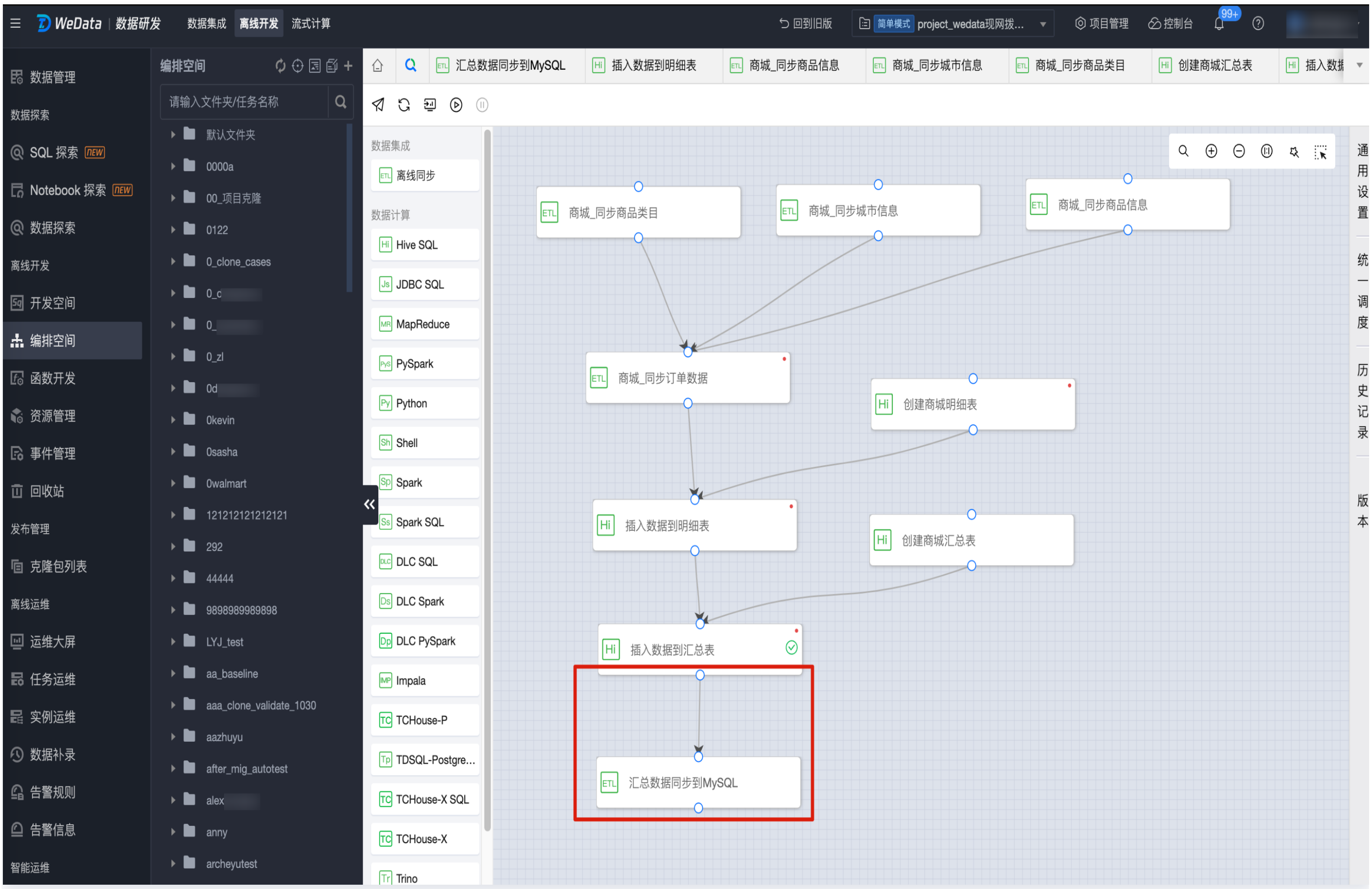
前置SQL: 选填，请根据数据源类型对应的SQL语法填写SQL

后置SQL: 选填，请根据数据源类型对应的SQL语法填写SQL

> **Note:**
> - Data Flow Direction: Hive → MySQL.
> - It is not possible to create MySQL tables in one click in WeData, so you need to create the data table in MySQL first.
> - Hive table filter condition: order_date = '${yyyy-MM-dd-1d}'.
> - MySQL write mode: overwrite.
> - Scheduling Settings:
>   - Scheduling Period: Choose a period.
>   - Set the execution time to: 01:00.

Step 2: Suggest dependencies and submit

Establish an association with the upstream task for the data synchronization task just created. For specific operations, please refer to Establish Dependencies and Submit.
The suggested task dependency relationship after association is as shown below:

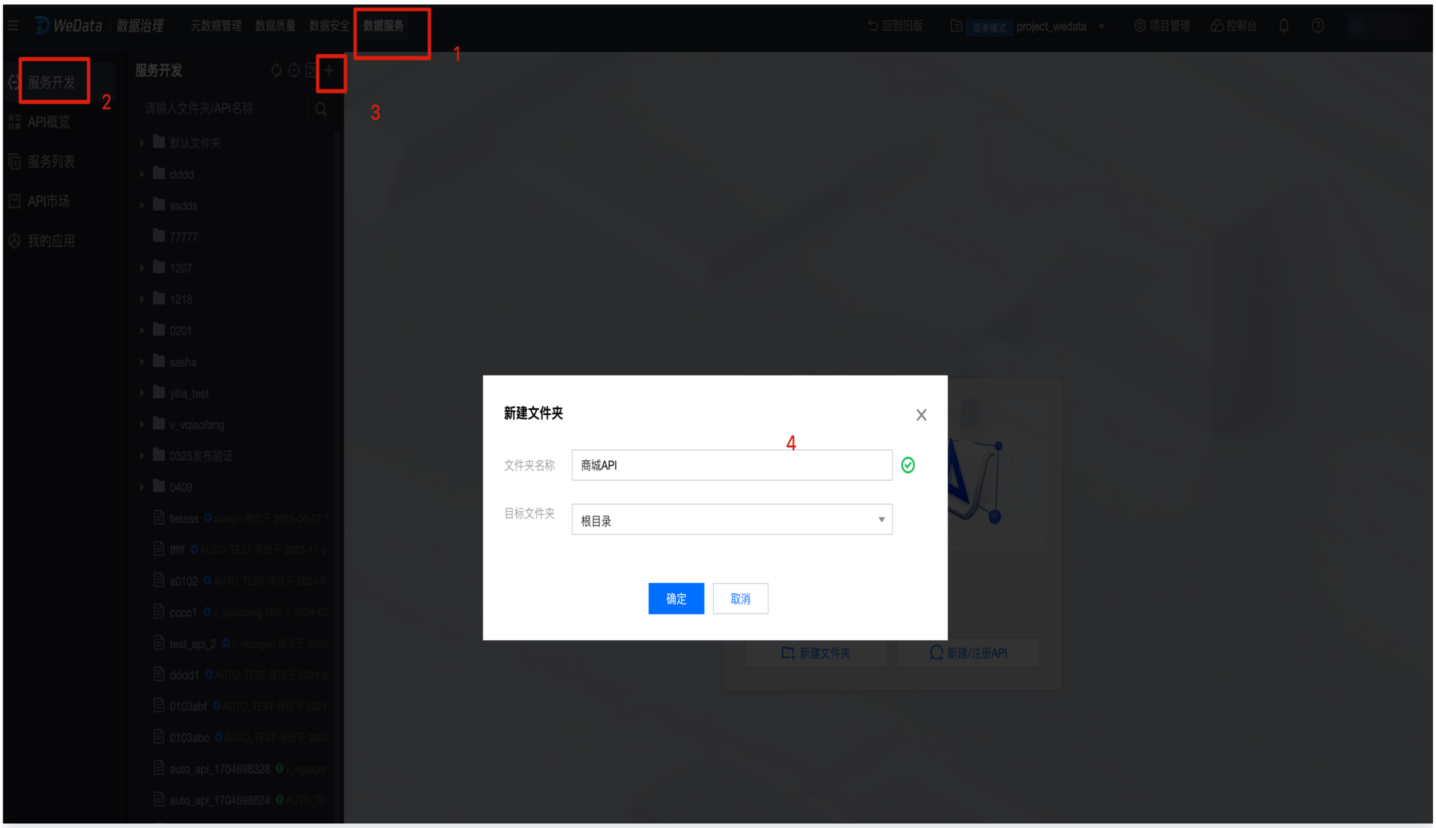


Upon completing the above steps, the data will automatically sync from the Hive table to MySQL.

Create Interface Service

Step 1: Create a Service

1. Enter the **Data Service > Service Development** page, click the **Create New Folder** button. Configure the folder name as **Shopping Mall API**. Select the target folder as the root directory. Click the **OK** button.



新建文件夹

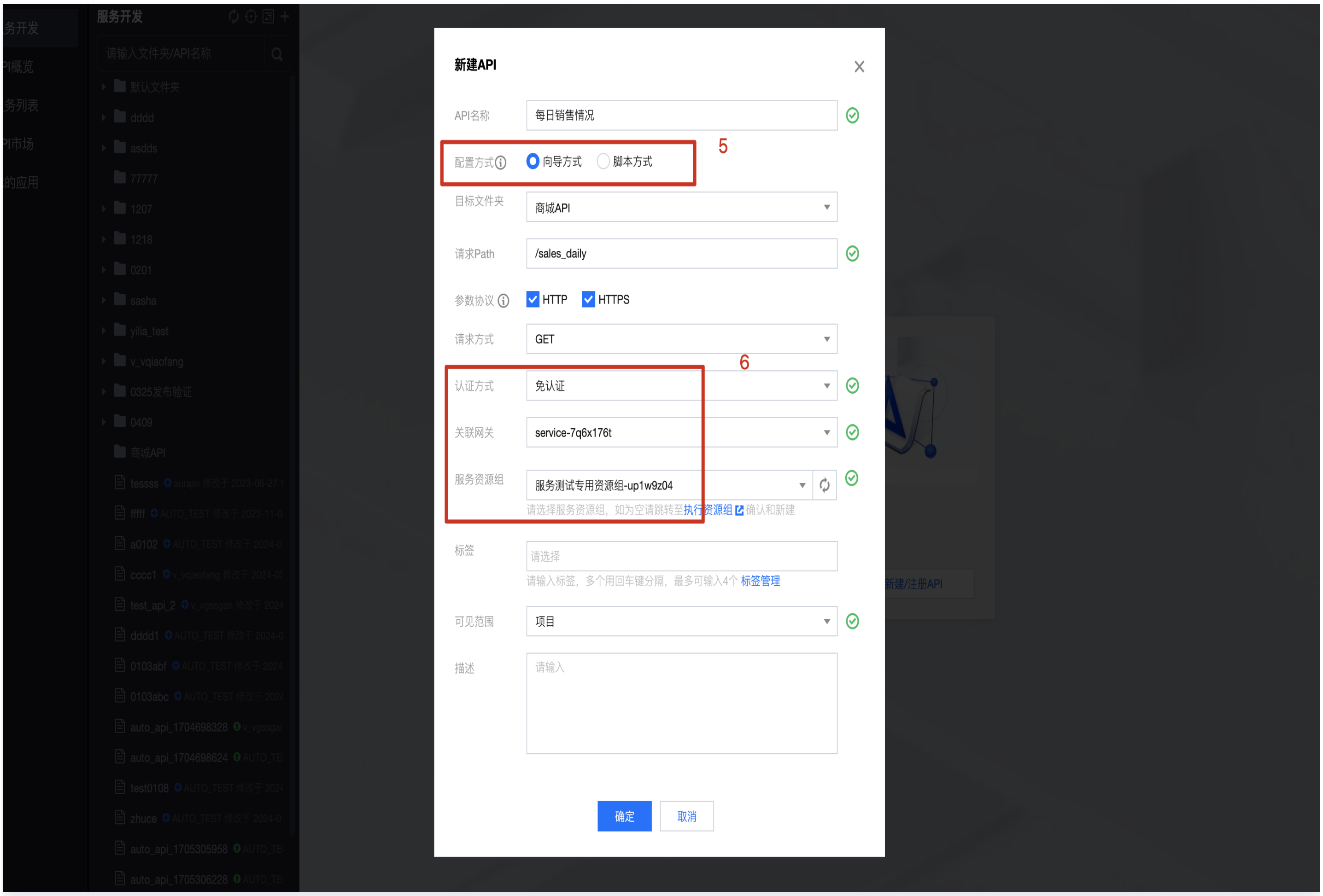
文件夹名称: 商城API

目标文件夹: 根目录

确定 取消

2. Then click **+ sign**, choose **Create New API** and click to enter the Create New API page. Select the configuration method as **Wizard Mode**, then configure **Authentication Method**, **Associated Gateway** and **Service Resource Group** sequentially. Click the **OK** button.

- Authentication Method: No Authentication.
- Associated Gateway: Select the API Gateway purchased in the step Purchasing API Gateway Resource.
- Service Resource Group: Select the data service purchased in the step Purchase Data Service Resources.



新建API

API名称：每日销售情况

配置方式：● 向导方式 ○ 脚本方式

目标文件夹：商城API

请求Path：/sales_daily

参数协议：☑ HTTP ☑ HTTPS

请求方式：GET

认证方式：免认证

关联网关：service-7q6x176t

服务资源组：服务测试专用资源组-up1w9z04

请选择服务资源组，如为空请跳转至执行资源组 确认和新建

标签：请选择

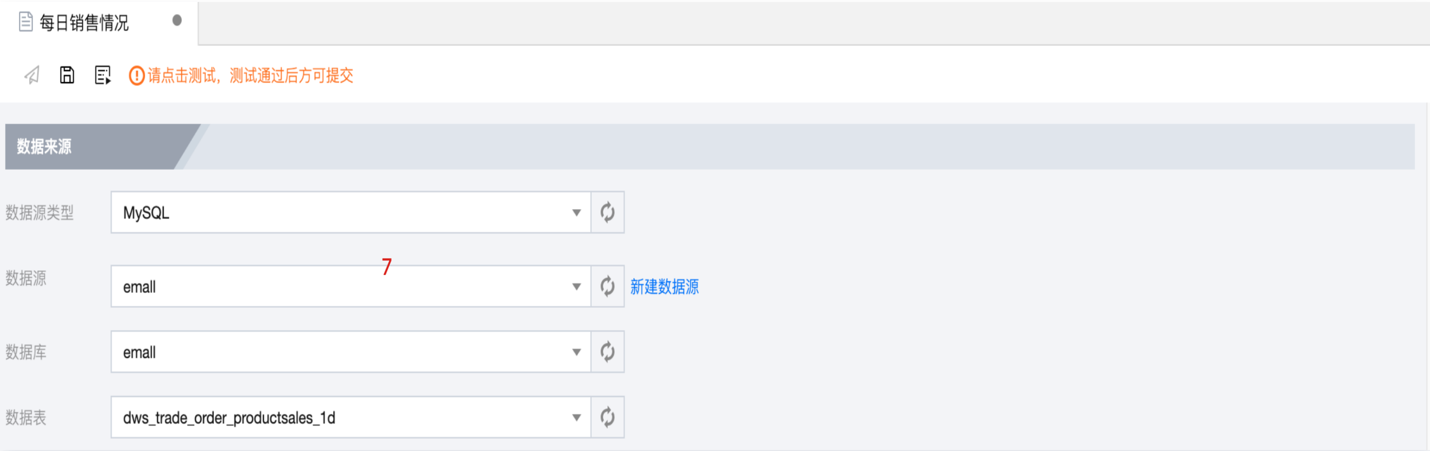
请输入标签，多个用回车键分隔，最多可输入4个 标签管理

可见范围：项目

描述：请输入

Step 2: Set Data Source

1. Select **MySQL** data source, and choose the **Summary Table** created in the previous step.



每日销售情况

请点击测试，测试通过后方可提交

数据来源

数据源类型：MySQL

数据源：email

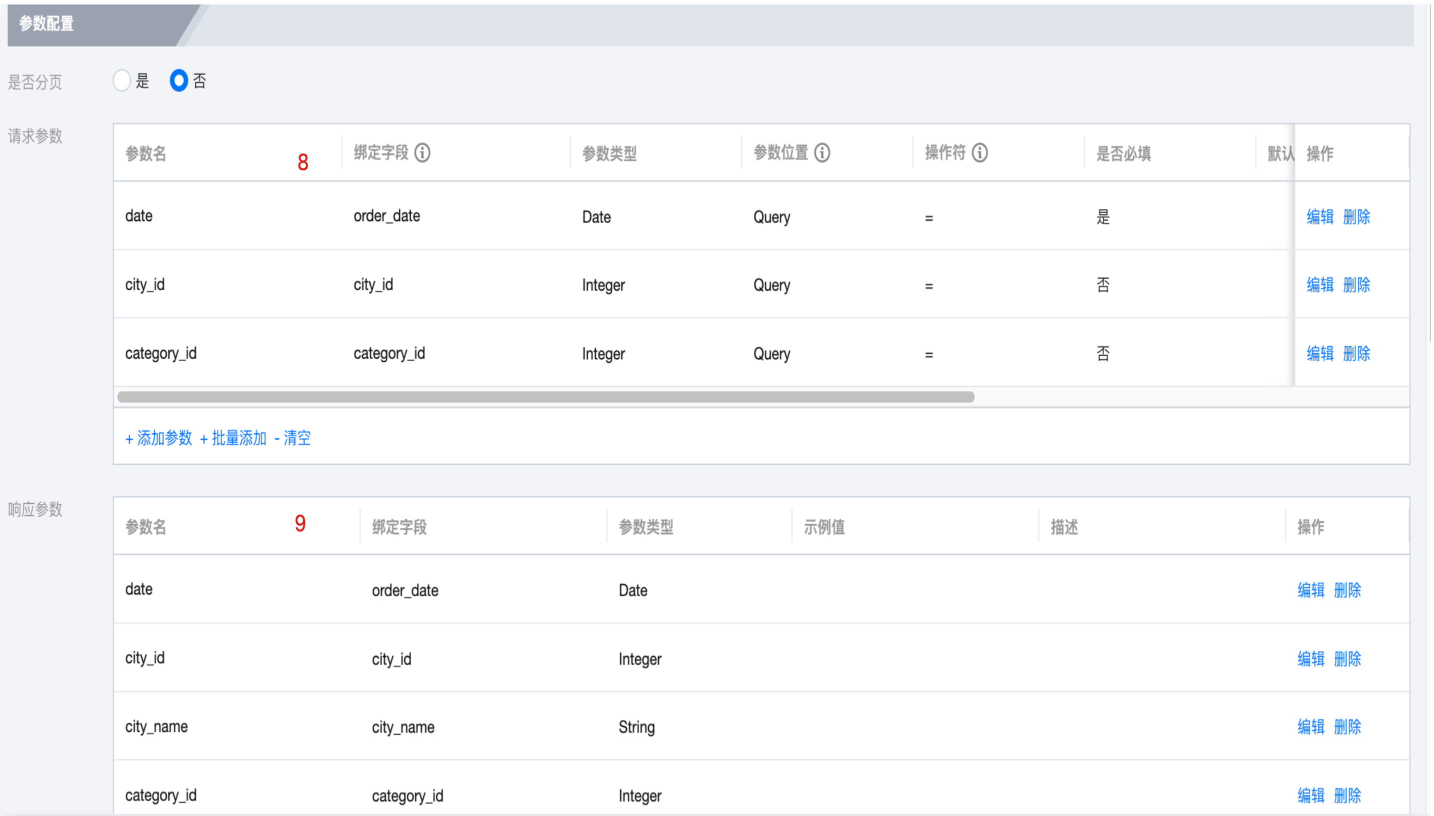
数据库：email

数据表：dws_trade_order_productsales_1d

Step 3: Set Input and Output Parameters

1. Configure sequentially **Request Parameters**, **Response Parameters**.

- Request parameters:
  - Set three parameters: date, city_id, category_id.
  - Bind the database field.
  - Adjust the parameter types.
  - Keep date as the required field, others as not required.
- Response parameters:
  - Set all fields in the table as output fields.
  - Quick operation: After selecting the bound field, the parameter name will be automatically brought out.
  - Adjust the parameter types.



参数配置

是否分页 ○ 是 ● 否

请求参数

参数名	绑定字段 ⓘ	参数类型	参数位置 ⓘ	操作符 ⓘ	是否必填	默认	操作
date	order_date	Date	Query	=	是		编辑 删除
city_id	city_id	Integer	Query	=	否		编辑 删除
category_id	category_id	Integer	Query	=	否		编辑 删除

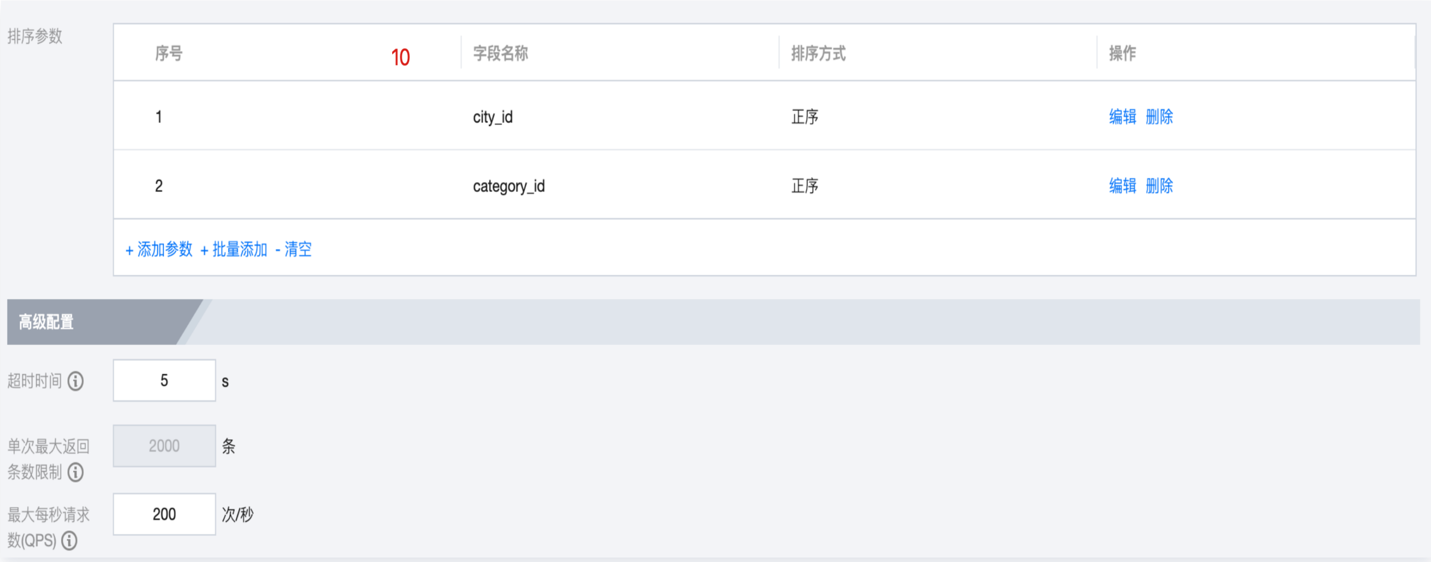
+ 添加参数 + 批量添加 - 清空

响应参数

参数名	绑定字段	参数类型	示例值	描述	操作
date	order_date	Date			编辑 删除
city_id	city_id	Integer			编辑 删除
city_name	city_name	String			编辑 删除
category_id	category_id	Integer			编辑 删除

2. Configure **Sorting Parameters**.

- Sorting parameters: Set city_id, category_id to ascending sort.



排序参数

序号	字段名称	排序方式	操作
1	city_id	正序	编辑 删除
2	category_id	正序	编辑 删除

+添加参数 +批量添加 -清空

高级配置

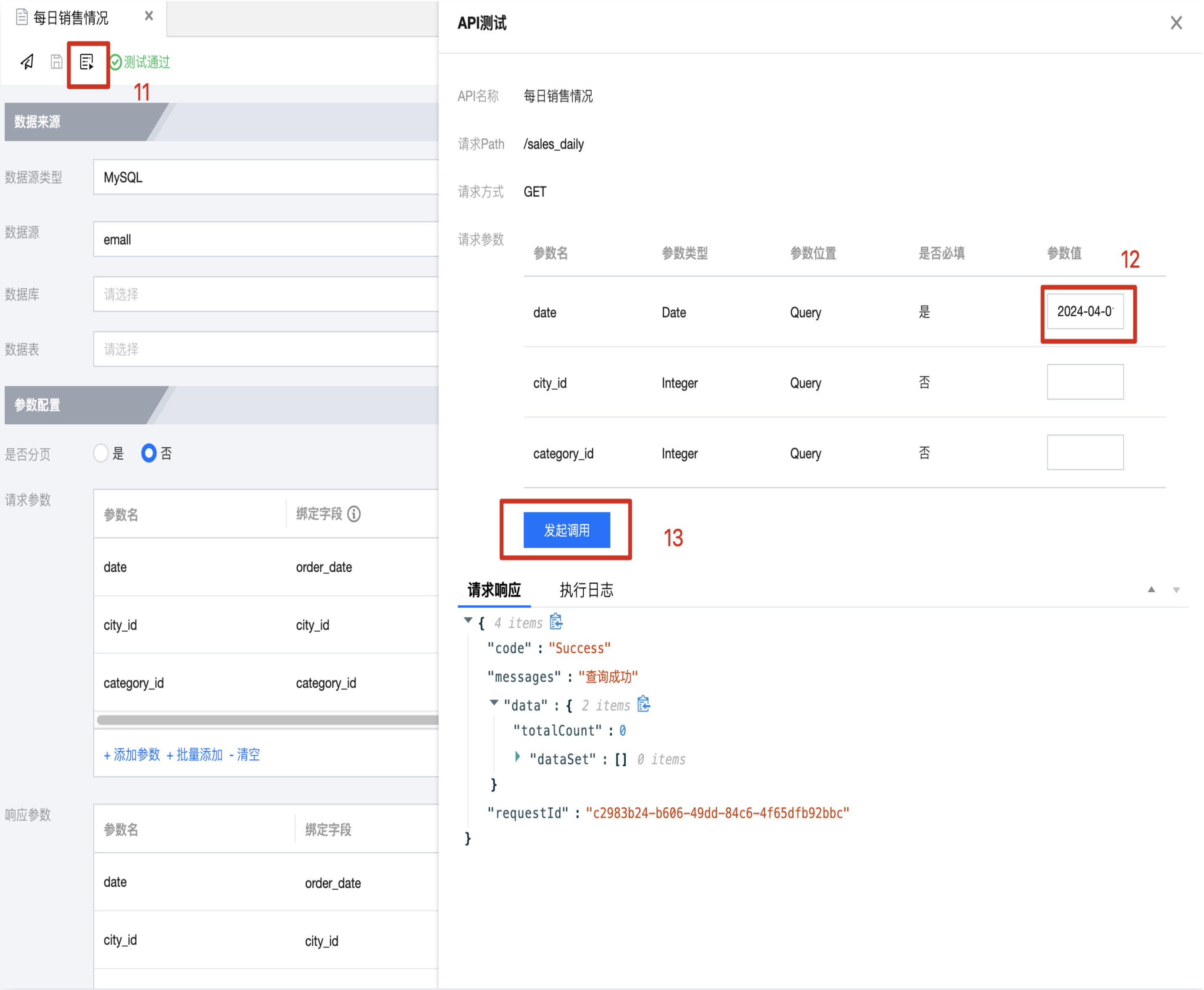
超时时间 ⓘ 5 s

单次最大返回条数限制 ⓘ 2000 条

最大每秒请求数(QPS) ⓘ 200 次/秒

Step 4: API Testing

1. Click the **Test** button, in the pop-up test window, fill in the **mandatory parameter** date, e.g., 2024-04-01, then click the **Initiate a call** button, wait for the response, and check the output result.



数据源类型	MySQL
数据源	email
数据库	请选择
数据表	请选择
是否分页	○ 是 ● 否

请求参数

参数名	绑定字段
date	order_date
city_id	city_id
category_id	category_id

+添加参数 +批量添加 -清空

响应参数

参数名	绑定字段
date	order_date
city_id	city_id

API名称 每日销售情况

请求Path /sales_daily

请求方式 GET

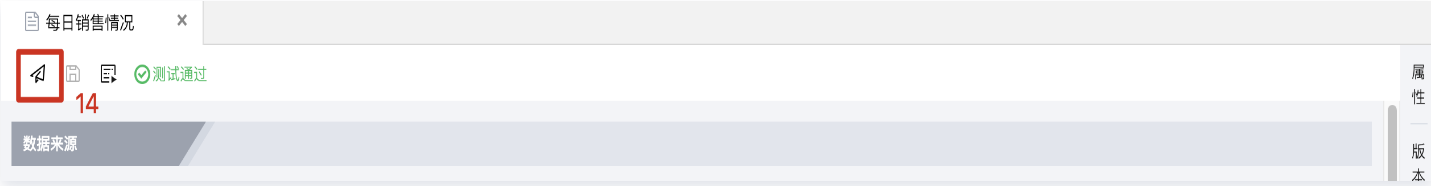
请求参数

参数名	参数类型	参数位置	是否必填	参数值
date	Date	Query	是	2024-04-0
city_id	Integer	Query	否	
category_id	Integer	Query	否	

```
{ 4 items
  "code" : "Success"
  "messages" : "查询成功"
  "data" : { 2 items
    "totalCount" : 0
    "dataSet" : [] 0 items
  }
  "requestId" : "c2983b24-b606-49dd-84c6-4f65dfb92bbc"
}
```

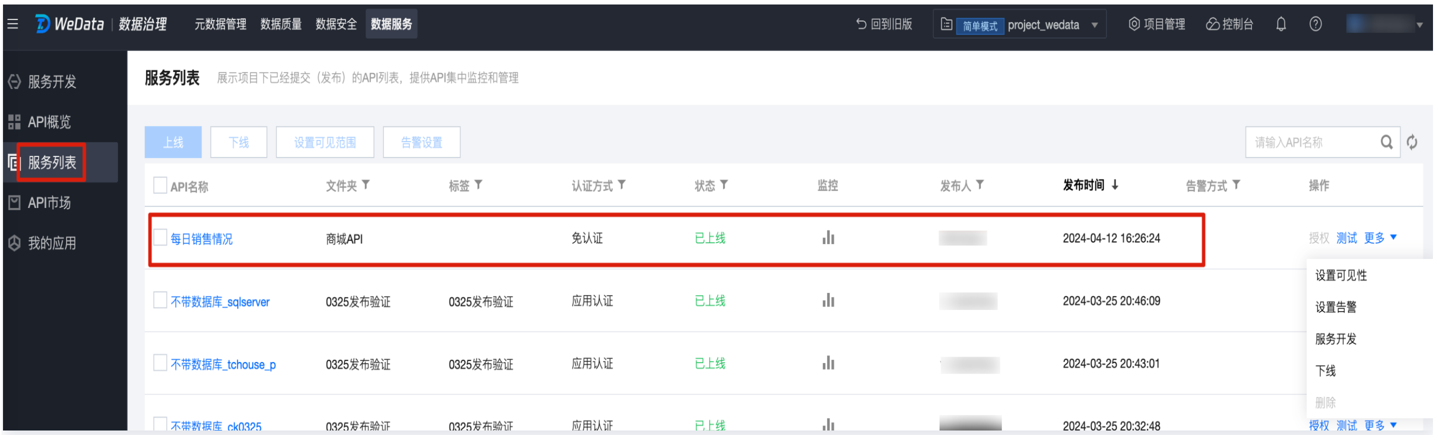
Step 5: Submit

1. After passing the test, click the **Submit** button. After submission, the interface is available.



View the Data Service Interface

After the Data Service API is submitted, you can view this service in the service list. In this list, you can manage the API, such as setting visibility, setting alarms, decommissioning, and other operations.



服务列表 展示项目下已经提交（发布）的API列表，提供API集中监控和管理

□ API名称	文件夹	标签	认证方式	状态	监控	发布人	发布时间 ↓	告警方式	操作
□ 每日销售情况	商城API		免认证	已上线	ılı	[redacted]	2024-04-12 16:26:24		授权 测试 更多 ▾
□ 不带数据库_sqlserver	0325发布验证	0325发布验证	应用认证	已上线	ılı	[redacted]	2024-03-25 20:46:09		
□ 不带数据库_tchouse_p	0325发布验证	0325发布验证	应用认证	已上线	ılı	[redacted]	2024-03-25 20:43:01		
□ 不带数据库_ck0325	0325发布验证	0325发布验证	应用认证	已上线	ılı	[redacted]	2024-03-25 20:32:48		授权 测试 更多 ▾

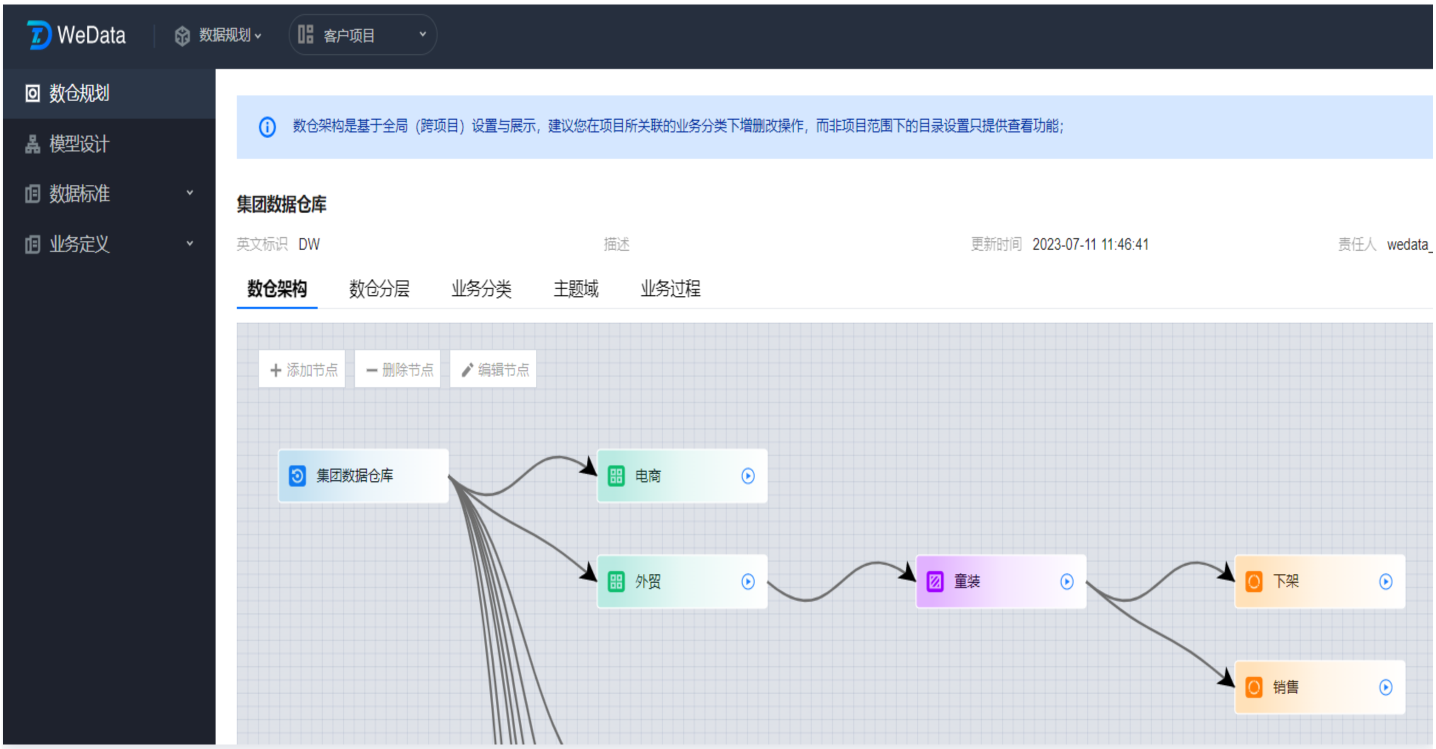
Business Research and Model Design (Optional)

Last updated：2024-08-24 17:36:18

Building a Data Warehouse

Business research will involve building the data warehouse framework based on business and data dimensions, using the concept of layering, categorization, and domain-based abstraction Definition:

- The Data Hierarchy Definition includes ODS, DWD, DWS, ADS, and DIM layers, which are mapped through logical layering and physical database associations.
- Business Type Definition includes business classification, subject domain, and business process, managing data objects with custom business catalog management.
- After Definition, the data warehouse architecture will be automatically generated. The subsequent process of defining model metrics dimensions will rely on the overall data warehouse architecture for Definition management.



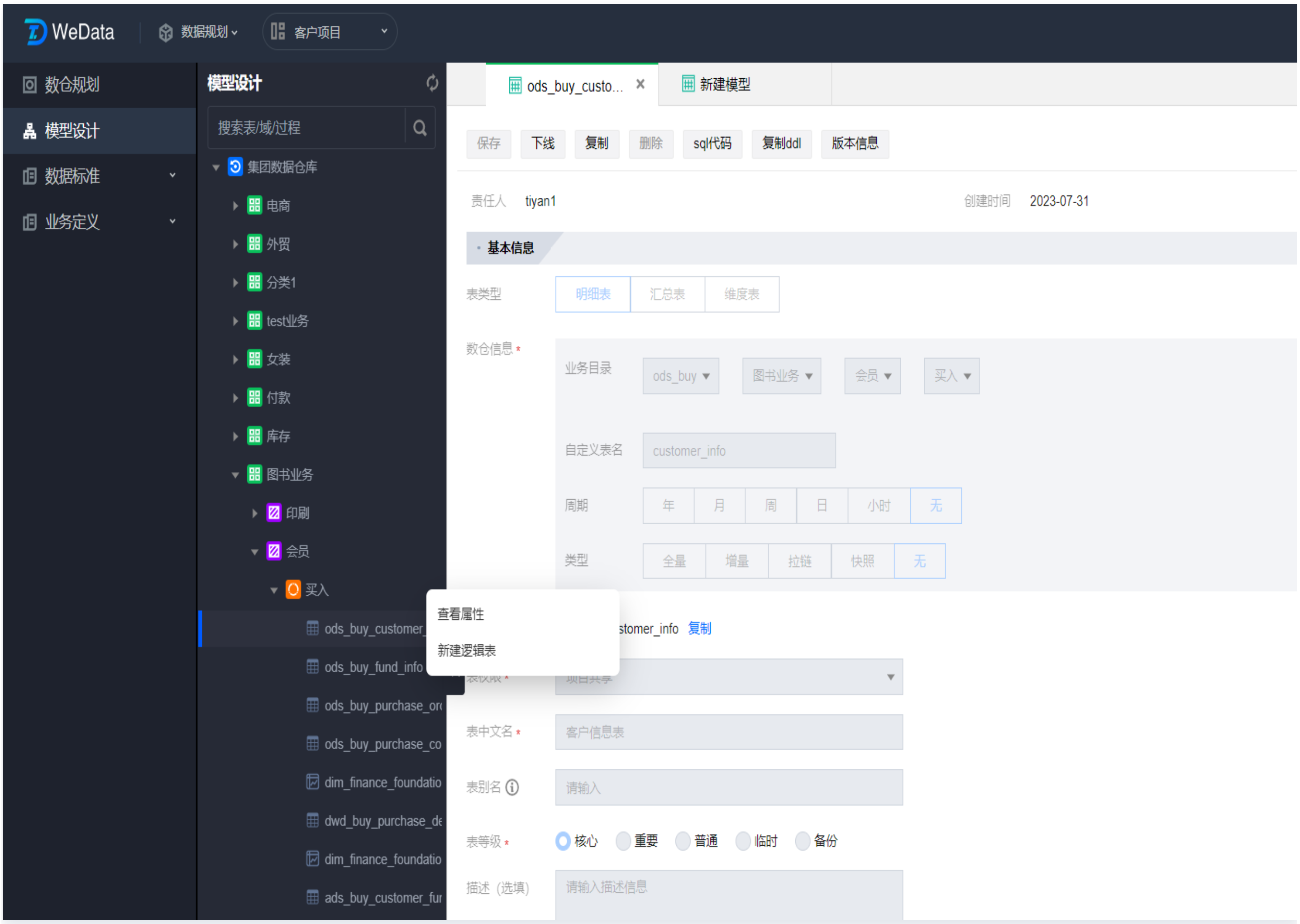
数仓架构是基于全局（跨项目）设置与展示，建议您在项目所关联的业务分类下增删改操作，而非项目范围下的目录设置只提供查看功能；

集团数据仓库

英文标识 DW　描述　更新时间 2023-07-11 11:46:41　责任人 wedata_

Model Design

After building the data warehouse, the designer will define the logical model and physical model based on data features and business scenarios. During the logical table definition, standardized naming is performed according to the data warehouse architecture. Additionally, metadata and value range standards can be bound during field configuration to complete the standardized definition process



责任人 tiyan1

创建时间 2023-07-31

表类型: 明细表 | 汇总表 | 维度表

数仓信息 *

业务目录: ods_buy ▾ 图书业务 ▾ 会员 ▾ 买入 ▾

自定义表名: customer_info

周期: 年 | 月 | 周 | 日 | 小时 | 无

类型: 全量 | 增量 | 拉链 | 快照 | 无

stomer_info 复制

表权限 *: 项目共享

表中文名 *: 客户信息表

表别名: 请输入

表等级 *: ● 核心 ○ 重要 ○ 普通 ○ 临时 ○ 备份

描述（选填）: 请输入描述信息

The model design process will consider both data-driven (Bottom-up) and business-driven (Top-down) approaches:

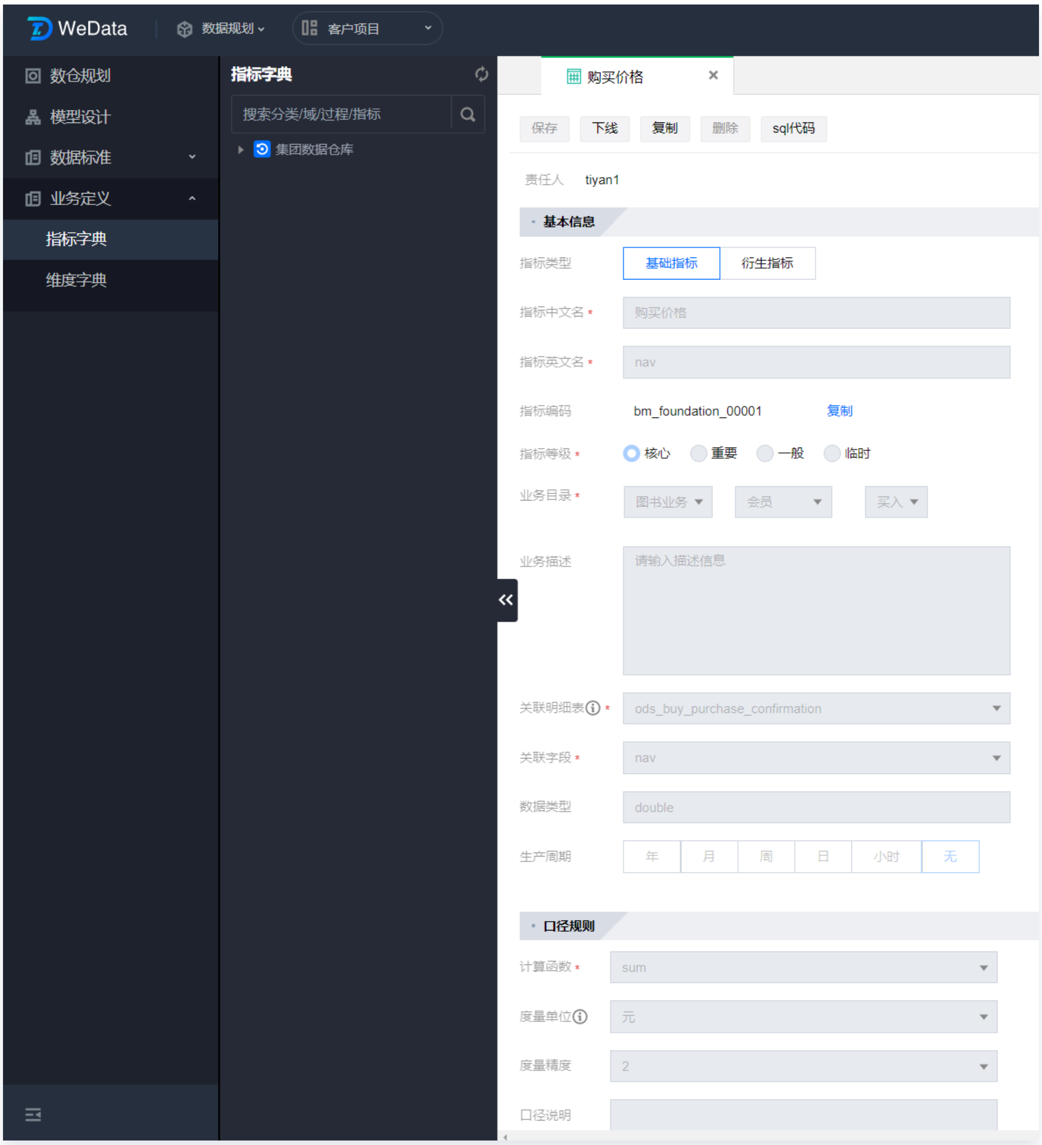
1. For the data-driven dimension, the designer first needs to synchronize raw data from the production source system to the interface layer. After cleansing and transforming, the raw data in the interface layer will form the detail table, also known as the fact table, which stores the finest granularity data. The detail table is the source for metric statistics, and its fields will be bound with basic metrics and dimension conditions.
2. From the business-driven perspective, based on business scenarios, designers need to define an aggregation layer and market layer. The summary table in the aggregation layer will store aggregated metric data under different dimension conditions and will be used as the target table for derived metrics, forming a one-to-one binding.
3. The defined analysis dimensions will create dimension tables that store attribute hierarchy data for dimensions. These tables will be one-to-one bound with common dimensions and can be auto-generated during dimension definition.

After completing the logical model design, the physical model can be generated through the publishing action, linking the design and development processes. If some physical models have already been created, they can be reverse-imported to generate logical models, completing the design phase.

Business Definition

During business research, the designer needs to abstractly define indicators and dimensions based on business scenarios:

1. Indicator Definition



责任人 tiyan1

基本信息

指标类型 [✓ 基础指标] 衍生指标

指标中文名 * 购买价格

指标英文名 * nav

指标编码 bm_foundation_00001 复制

指标等级 * ● 核心 ○ 重要 ○ 一般 ○ 临时

业务目录 * 图书业务 会员 买入

业务描述 请输入描述信息

关联明细表 * ods_buy_purchase_confirmation

关联字段 * nav

数据类型 double

生产周期 年 月 周 日 小时 [✓ 无]

口径规则

计算函数 * sum

度量单位 元

度量精度 2

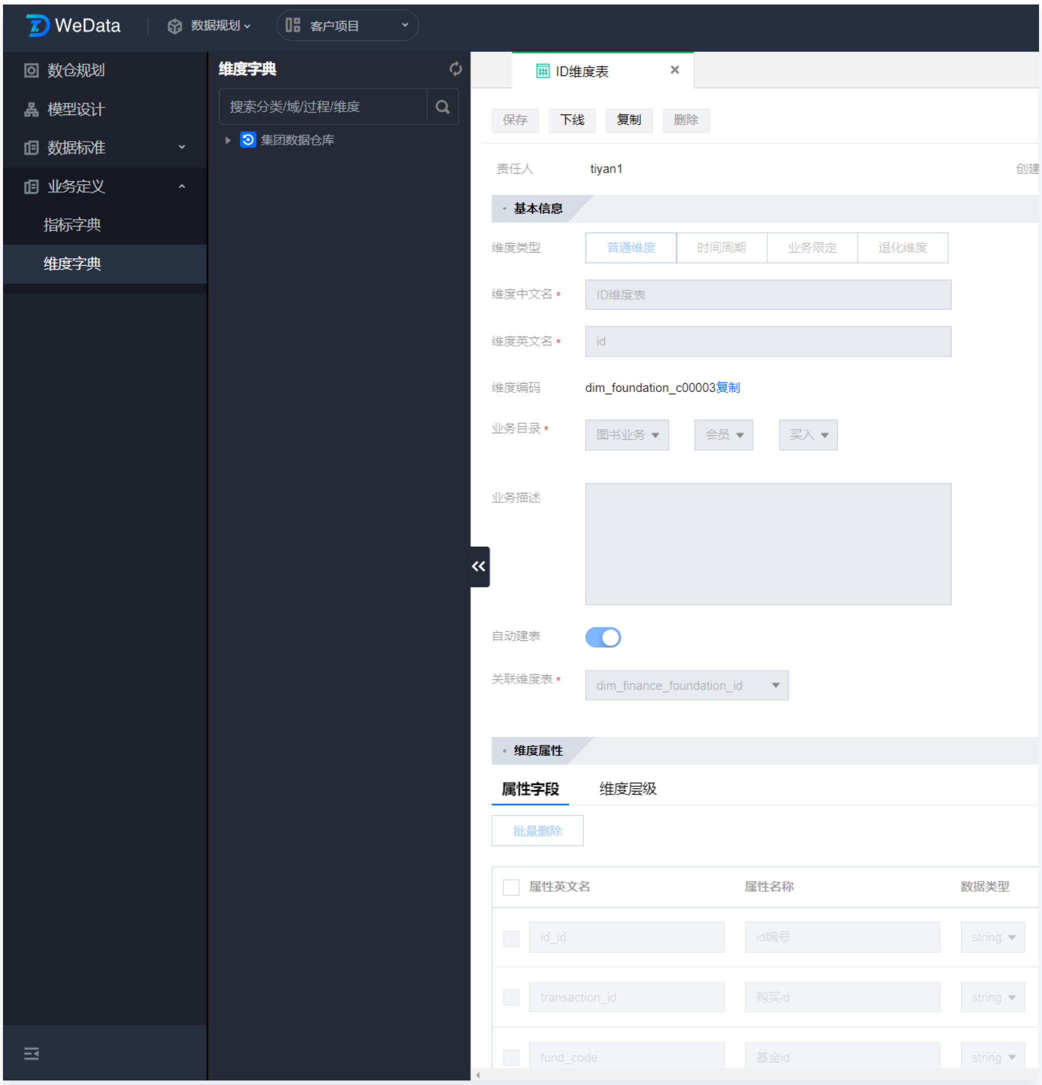
口径说明

Indicators are divided into two categories: basic metrics and derived metrics

1.1 Basic metrics are measurements that do not include dimensional conditions. They require definitions of their basic attributes, statistical calibers, and unit precisions. Derived metrics will inherit the unit of the basic metrics. The data for basic metrics comes from a specific field in the detail table, so they need to be associated in the indicator definition.

1.2 Derived metrics can be defined by adding dimensional conditions to basic metrics for specific characteristic ranges, such as user growth in a certain channel or product type. They can also be the result of combined calculations of multiple derived metrics under the same dimensional conditions, such as growth rate. Once defined, derived metrics are bound to a field in a summary table, facilitating indicator production.

2. Dimension Definition



责任人 tiyan1

维度类型 普通维度

维度中文名 * ID维度表

维度英文名 * id

维度编码 dim_foundation_c00003复制

业务目录 * 图书业务 会员 买入

业务描述

自动建表

关联维度表 * dim_finance_foundation_id

属性英文名	属性名称	数据类型
id_id	id编号	string
transaction_id	购买id	string
fund_code	基金id	string

Dimensions can be classified into the following categories:

2.1 Common dimension: This can be understood as the group by condition in SQL. A common dimension uniquely corresponds to a dimension table, associated during dimension modeling

2.2 Business Constraint: Also known as a modifier, it is used to filter tag characteristics from business dimensions

2.3 Time period: Time-based limiting conditions

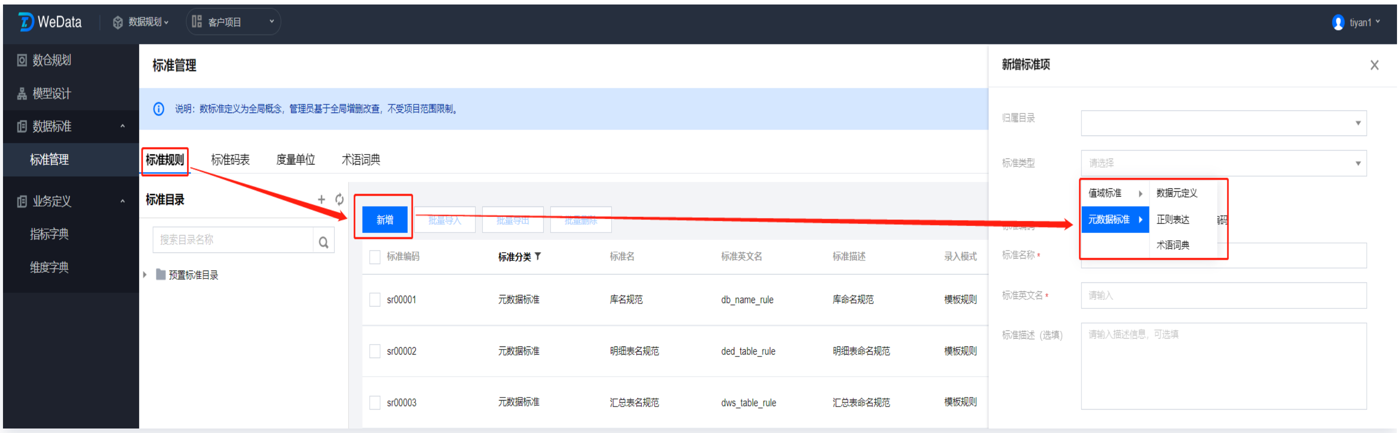
2.4 Degenerate Dimension: Dimensions reverted to the fact table. This usually happens when a dimension has no other content besides the primary key, even though it's a legitimate dimension key. Reverting it to the fact table reduces the number of associations and improves query performance.

Indicators and dimensions need to be defined and published sequentially to establish associations with the table model and be referenced by subsequent derived definitions, guiding the implementation of indicator production.

Data Standard

Model Metrics Definition: During the development and production process, operations must adhere to a unified data standard. Therefore, business objects need rules to be defined. Standards management involves defining standards across the following four modules:

1. Definition Standard Rules



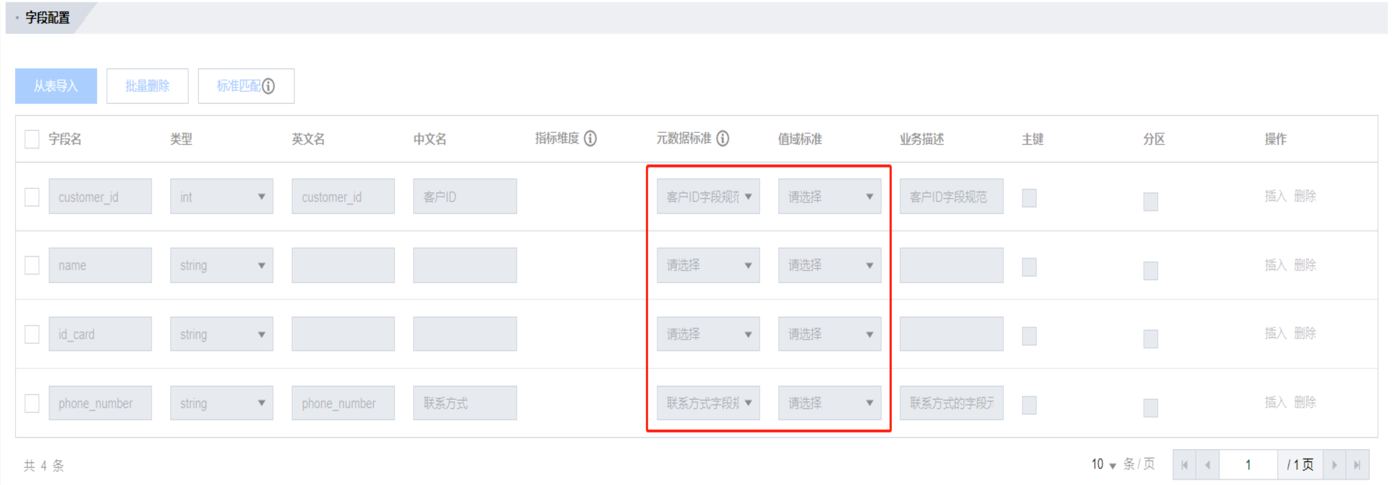
说明：数标准定义为全局概念，管理员基于全局增删改查，不受项目范围限制。

标准编码	标准分类	标准名	标准英文名	标准描述	录入模式
sr00001	元数据标准	库名规范	db_name_rule	库命名规范	模板规则
sr00002	元数据标准	明细表名规范	ded_table_rule	明细表命名规范	模板规则
sr00003	元数据标准	汇总表名规范	dws_table_rule	汇总表命名规范	模板规则

归属目录
标准类型：请选择
标准名称 *
标准英文名 *：请输入
标准描述（选填）：请输入描述信息，可选填

You can define standards at the table, field, and indicator levels. Metadata standards define the naming and type specifications of business objects, and value range standards define the characteristics of value ranges.

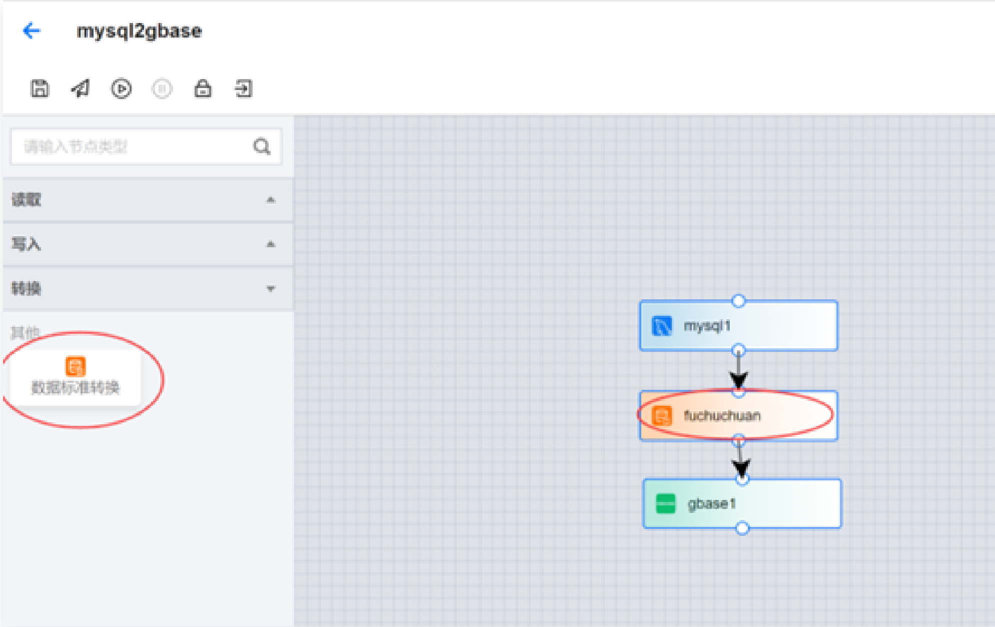
After rule release, you can perform association binding in the model design/physical model fields:



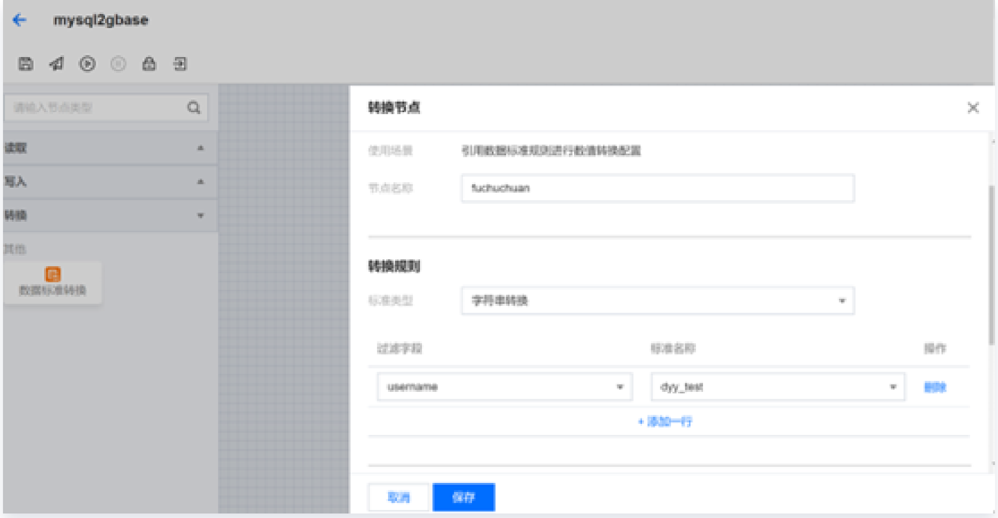
字段名	类型	英文名	中文名	指标维度	元数据标准	值域标准	业务描述	主键	分区	操作
customer_id	int	customer_id	客户ID		客户ID字段规范	请选择	客户ID字段规范	□	□	插入 删除
name	string				请选择	请选择		□	□	插入 删除
id_card	string				请选择	请选择		□	□	插入 删除
phone_number	string	phone_number	联系方式		联系方式字段规	请选择	联系方式的字段	□	□	插入 删除

共 4 条　10 条 / 页　1 / 1 页

You can also use ETL tasks for standard conversion tasks during subsequent data development processes:



Configure Conversion Rules:



转换节点

使用场景 引用数据标准规则进行数值转换配置

节点名称 fuchuchuan

转换规则

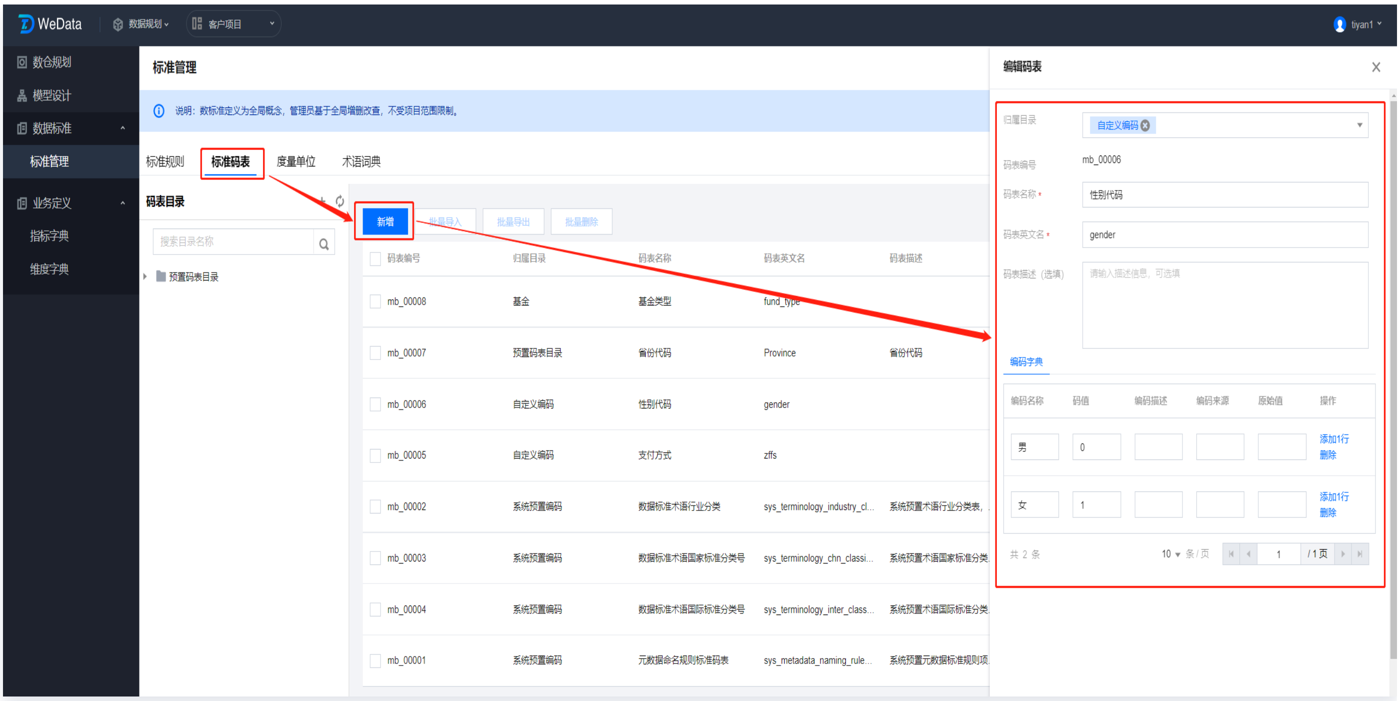
标准类型 字符串转换

过滤字段	标准名称	操作
username	dyy_test	删除

+ 添加一行

2. Definition Standard Encoding

 For data enumeration types, management requires standard encoding. After encoding is defined and released, it can be referenced in standard rules.



码表编号	归属目录	码表名称	码表英文名	码表描述
mb_00008	基金	基金类型	fund_type	
mb_00007	预置码表目录	省份代码	Province	省份代码
mb_00006	自定义编码	性别代码	gender	
mb_00005	自定义编码	支付方式	zffs	
mb_00002	系统预置编码	数据标准术语行业分类	sys_terminology_industry_cl...	系统预置术语行业分类表，...
mb_00003	系统预置编码	数据标准术语国家标准分类号	sys_terminology_chn_classi...	系统预置术语国家标准分类...
mb_00004	系统预置编码	数据标准术语国际标准分类号	sys_terminology_inter_class...	系统预置术语国际标准分类...
mb_00001	系统预置编码	元数据命名规则标准码表	sys_metadata_naming_rule...	系统预置元数据标准规则项...

归属目录：自定义编码

码表编号：mb_00006

码表名称 *：性别代码

码表英文名 *：gender

码表描述（选填）：请输入描述信息，可选填

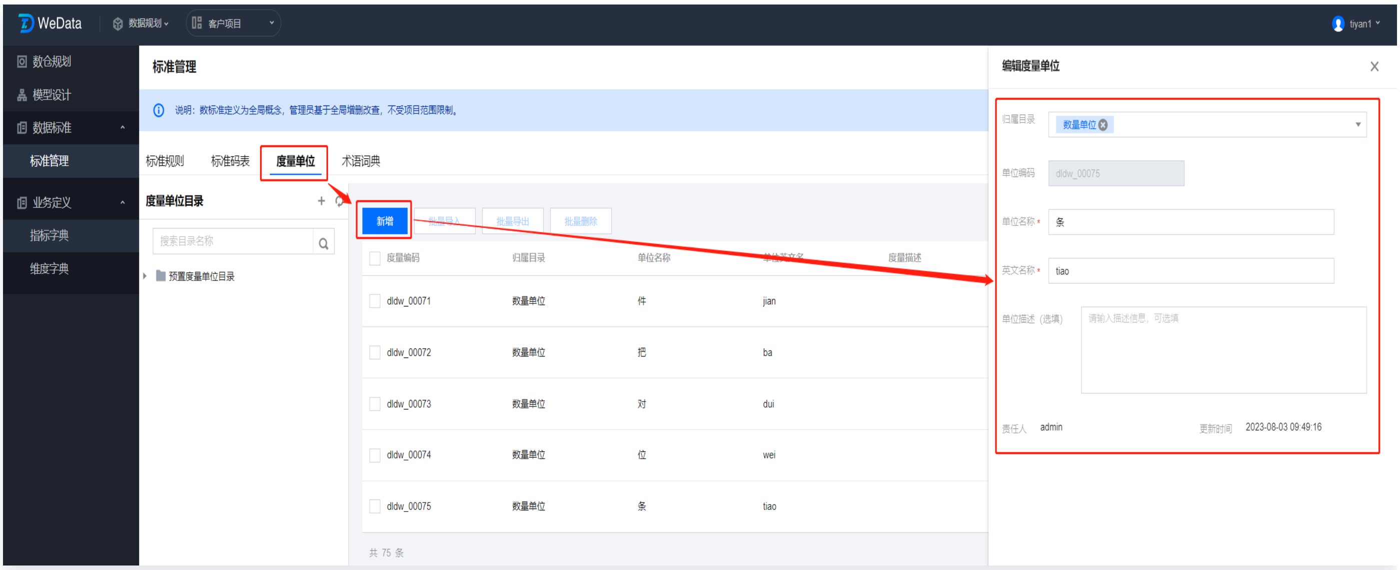
编码字典

编码名称	码值	编码描述	编码来源	原始值	操作
男	0				添加1行 删除
女	1				添加1行 删除

共 2 条　10 条 / 页　1 / 1 页

3. Definition Measurement Unit

When defining indicators, measurement units will be used. The system presets common units. For custom units, you can define them in this module.



度量编码	归属目录	单位名称	单位英文名	度量描述
dldw_00071	数量单位	件	jian	
dldw_00072	数量单位	把	ba	
dldw_00073	数量单位	对	dui	
dldw_00074	数量单位	位	wei	
dldw_00075	数量单位	条	tiao	

共 75 条

归属目录：数量单位

单位编码：dldw_00075

单位名称 *：条

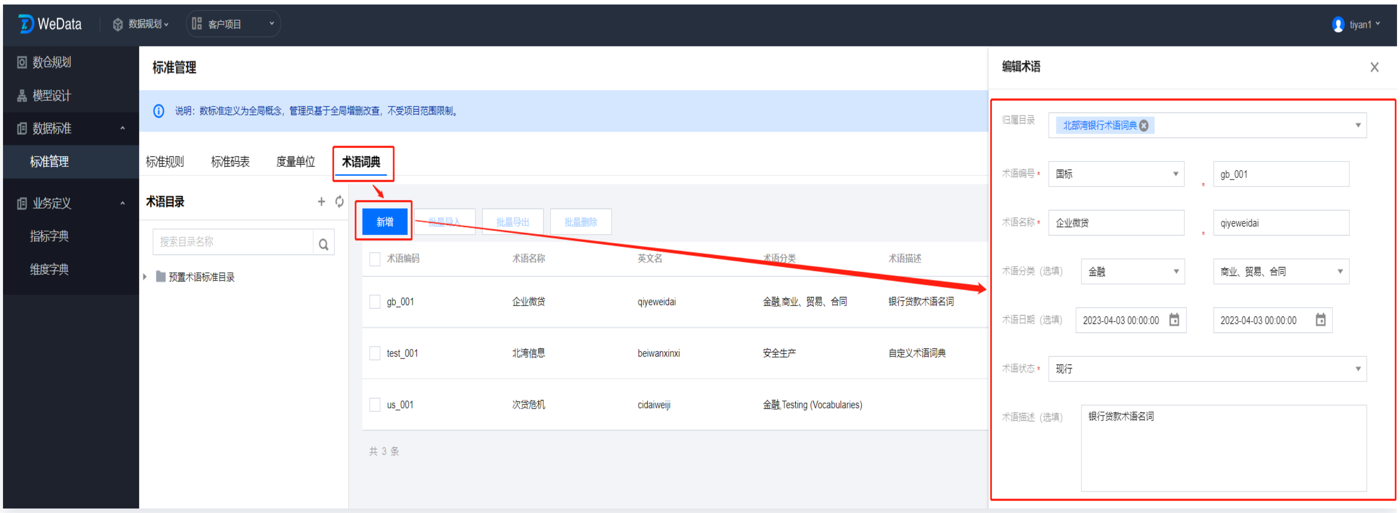
英文名称 *：tiao

单位描述（选填）：请输入描述信息，可选填

责任人：admin　更新时间：2023-08-03 09:49:16

4. Definition Terminology Dictionary

Industry standard metadata will be defined in bulk in the terminology dictionary. After definition release, it can be referenced in standard rules.



说明：数标准定义为全局概念，管理员基于全局增删改查，不受项目范围限制。

术语编码	术语名称	英文名	术语分类	术语描述
gb_001	企业微贷	qiyeweidai	金融,商业、贸易、合同	银行贷款术语名词
test_001	北湾信息	beiwanxinxi	安全生产	自定义术语词典
us_001	次贷危机	cidaiweiji	金融,Testing (Vocabularies)	

共 3 条

编辑术语

归属目录：北部湾银行术语词典

术语编号 *：国标 | gb_001

术语名称 *：企业微贷 | qiyeweidai

术语分类（选填）：金融 | 商业、贸易、合同

术语日期（选填）：2023-04-03 00:00:00 | 2023-04-03 00:00:00

术语状态 *：现行

术语描述（选填）：银行贷款术语名词