Alibaba Cloud DataWorks

Best Practices

Issue: 20190115

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.
5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectu

al property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).

6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 **Danger:** Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 **Warning:** Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 **Notice:** Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructio ns, best practices, tips, and other content that is good to know for the user.	 **Note:** You can use **Ctrl** + **A** to select all files.
>	Multi-level menu cascade.	**Settings** > **Network** > **Set network type**
Bold	It is used for buttons, menus, page names, and other UI elements.	Click **OK**.
`Courier font`	It is used for commands.	Run the `cd /d C:/windows` command to enter the Windows system folder.
`Italics`	It is used for parameters and variables.	`bae log list --instanceid` *`Instance_ID`*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *`[-all|-t]`*
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich` *`{stand | slave}`*

Contents

1 Simple mode and standard mode

The new version of DataWorks introduces both simple and standard modes, this article introduces you to the differences between simple and standard modes.

Simple Mode

A simple mode refers to a DataWorks project that corresponds to a MaxCompute project and cannot set up a development and production environment, you can only do simple data development without strong control over the data development process and table permissions.

The advantage of the simple mode is that the iteration is fast, and the code is submitted without publishing, it will take effect.

The risk of a simple mode is that the development role is too privileged to delete the tables under this project, there is a risk of table permissions.

Standard Mode

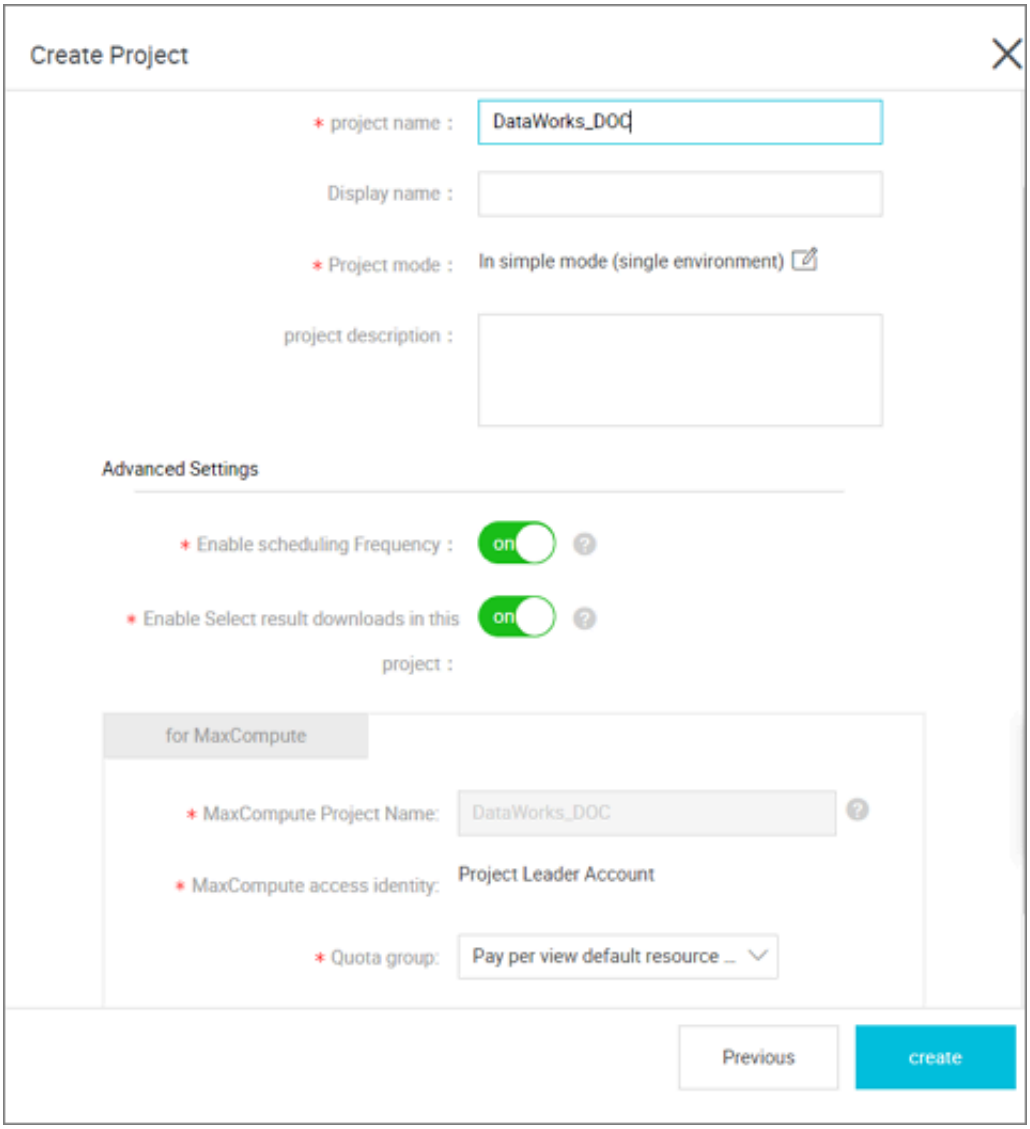
Standard mode refers to a DataWorks project corresponding to two MaxCompute projects, which can be set up to develop and produce dual environments, improve code development specifications and be able to strictly control table permissions, the operation of tables in production environments is prohibited, and the data security of production tables is guaranteed.

- All Task edits can be performed only in the development environment, and the Production Environment Code cannot be directly modified, reduce the production environment code modification entry, as much as possible to ensure the production environment code stability.
- The development environment does not turn on task scheduling by default, avoid the development of environmental project cycle operation and production of environmental projects to seize resources, the stability of the operation of production environment tasks is better guaranteed.
- The production environment runs with a default production account, all the tables produced by the production account belong to the main account, you need to use production tables during the development process, all of which need to be applied separately, better control of table permissions.

When creating a project, select **project mode** as the standard mode, fill in the project name and project description, the remaining configuration item select the default value.

 **Note:**

The MaxCompute access identity of the production environment cannot be modified to a personal account, otherwise, the data security of the production environment cannot be guaranteed.



2 Workshop

2.1 Workshop course introduction

This module introduces you to the design ideas and core capabilities of DataWorks, to help you gain insight into the ideas and capabilities of Alibaba Cloud DataWorks.

Course Overview

Course duration: Two hours, using an online learning method.

Course object: for all new and old users of DataWorks, such as Java engineer, product operation , HR, etc, as long as you are familiar with standard SQL, you can quickly master the basic skills of DataWorks, you don't need to know much about the principles of data warehouses and MaxCompute. However, it is also recommended that you further study the DataWorks course to gain insight into the basic concepts and functions of DataWorks.

Course objective: Take the common real-world massive log data analysis task as the curriculum background, after completing the course, you will be able to understand the main features of DataWorks, able to demonstrate content according to the course, independently complete data acquisition, data development, task operations and other data jobs common tasks.

This course includes the following:

- Product introduction: You will learn about DataWorks' development history, its overall architectu re, and its modules and their relationships.
- Data Acquisition: Learn How to synchronize data from different data sources to MaxCompute, how to quickly trigger task runs, how to view task logs, and so on.
- Data Processing: learn how to run a data flow chart, how to create a new data table, how to create a data process task node, how to configure periodic scheduling properties for tasks.
- Data quality: Learn how to configure monitoring rules for data quality for tasks, ensure that the task runs quality issues.

DataWorks introduction

DataWorks is a big data research and development platform, using MaxCompute as the main calculation engine, including data integration, data modeling, data development, operations and operations monitoring, data management, data security, data quality, and other product functions

. At the same time, with the algorithm platform PAI to get through, complete link from big data development to Data Mining and machine learning.

Data Collection

For more information on data acquisition, see *Data acquisition: log data upload*.

Data Processing

For details on data processing, see *Data processing: user portraits*.

Data quality

For more information on data quality, see *Data quality monitoring*.

Learning to answer questions

If you encounter problems in the learning process, you can add DingTalk groups: 11718465, consulting Alibaba cloud technical support.

2.2 Data acquisition: log data upload

Related Products

The big data products involved in this experiment are *MaxCompute (big data computing services)*. And *DataWorks (data factory, original big data development kit)*.

Prerequisites

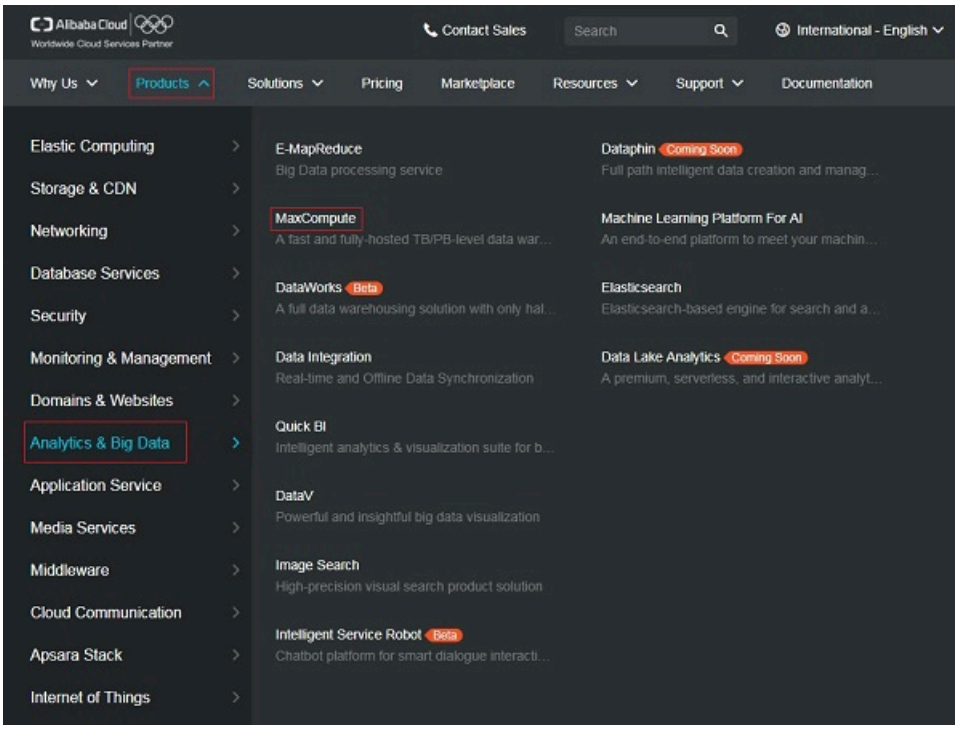
Before you begin this lab, you need to make sure you have an Alibaba Cloud account and have a real name.

Activate MaxCompute

Note:

If you have already activated MaxCompute, skip this step to create the project space directly.

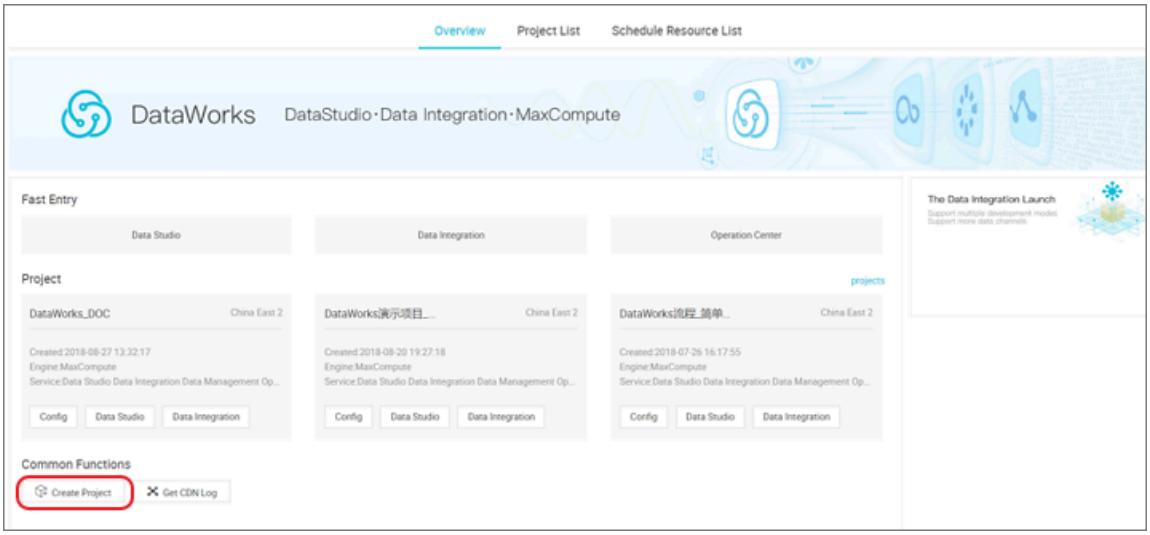
1. Log in to the *Alibaba Cloud website*, click **Log in** in the upper-right corner to fill in your Alibaba Cloud account and password.
2. Select **Products** > **Analytics & Big Data** > **MaxComputute** and go to the MaxCompute product details page.



3. Click **Start now**.
4. Select **Pay-As-You-Go**, click **Buy Now**.

Create Project

1. Log on to the *DataWorks console* by using a primary account.
2. You can create a MaxCompute project in two ways.
   - On the console overview page, go to **Common FunctionsCreate Project**.

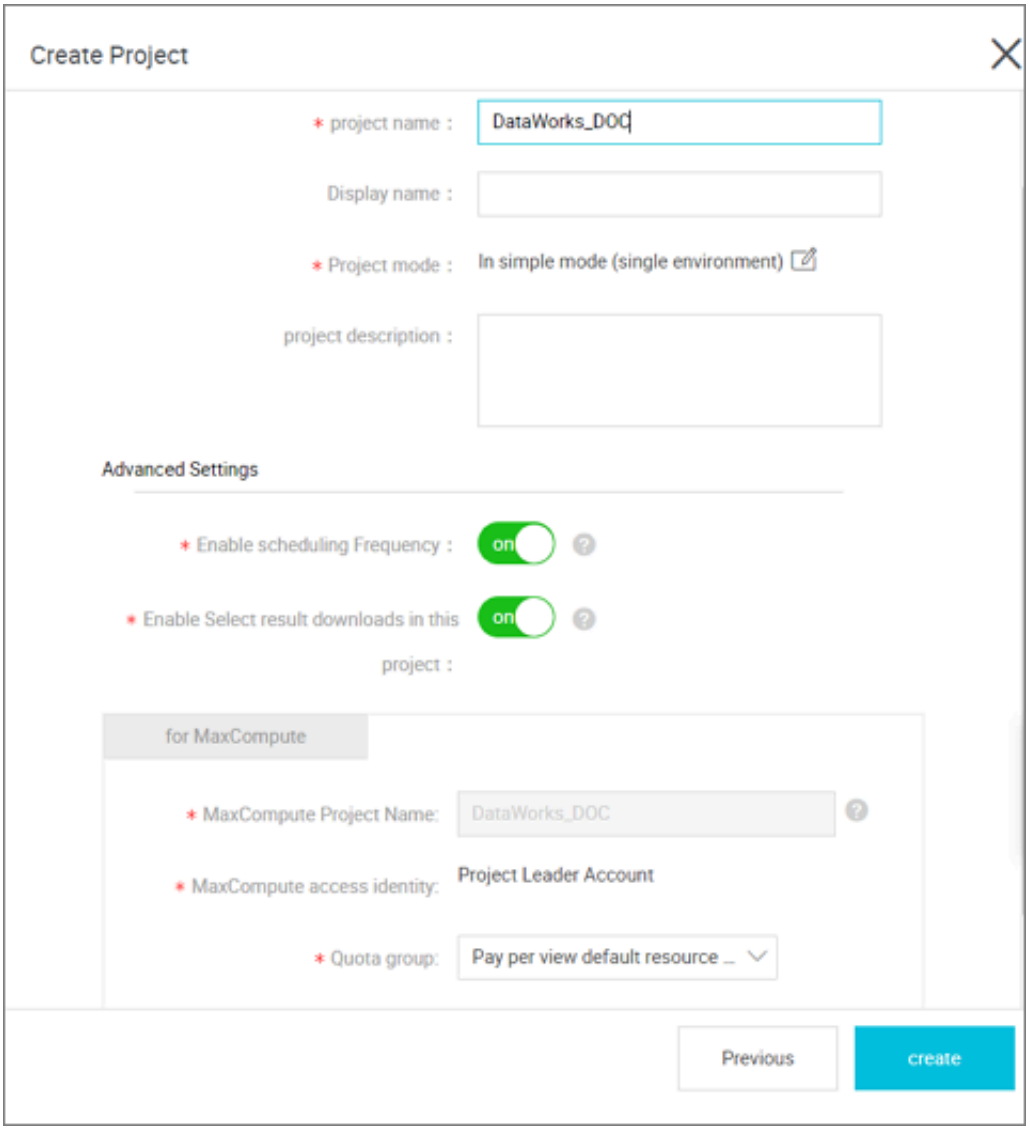


3. Fill in the configuration items in the **Create Project** dialog box. Select a region and a calculation engine service.

Note:

If you have not purchase the relevant services in the region, it is directly display that there is no service available in the Region. The data development, O&M center, and data management are selected by default.

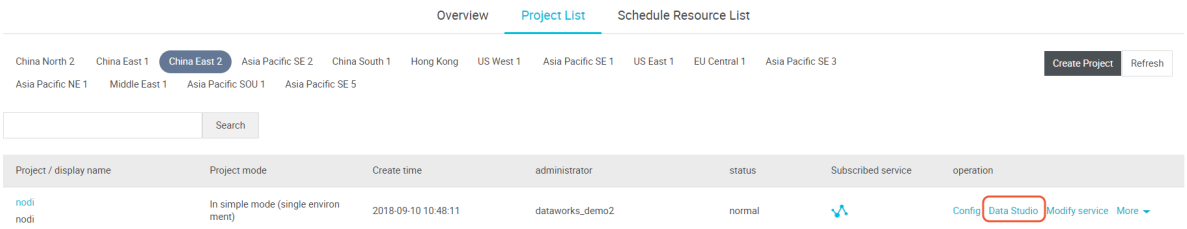
4. Configure the basic information and advanced settings for the new project, and click **Create project**.



Note:

- The project name needs to begin with a letter or underline, and can only contain letters, underscores, and numbers.
- The project name is globally unique, it is recommended that you use your own easy-to-distinguish name as the project space name for this lab.

5. Once the project has been created successfully, you can select the **Project List** page to **Data Studio** after viewing the project space.



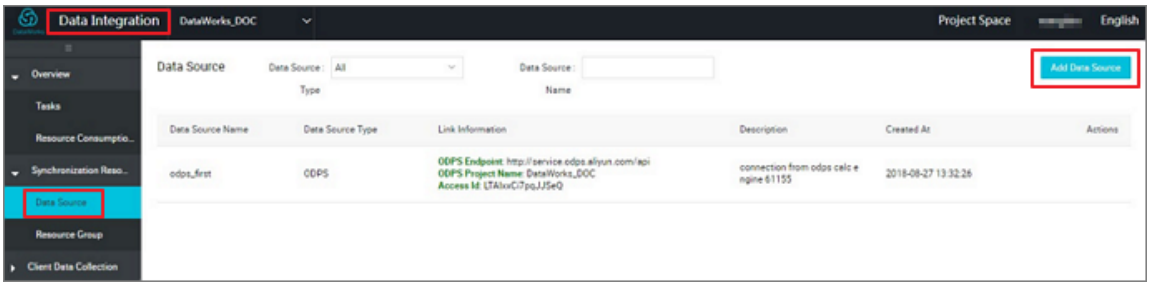
Create data source

Note:

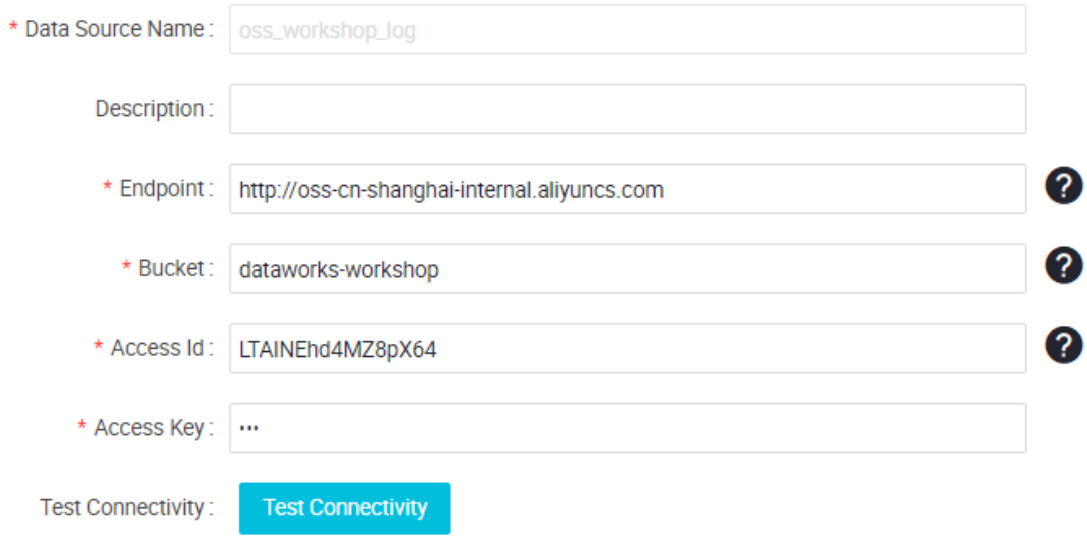
Based on the scenario simulated by this lab, you need to distribute to create both the OSS data source and the RDS data source.

- Create a new OSS data source

1. Select the **Data Integration** > **Data Source** Page, and click **Add Data Source**.



2. Select the data source type as OSS, with other configuration items as follows.



Parameters:

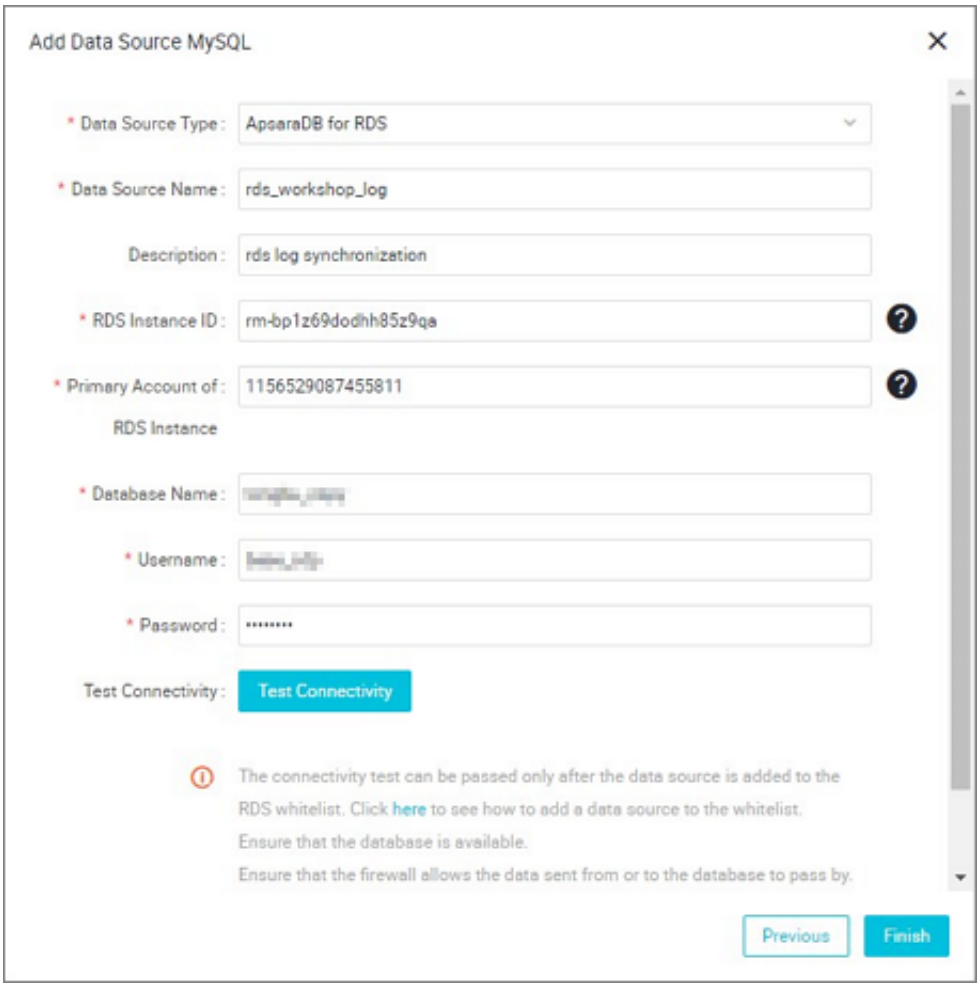
- Endpoint: http://oss-cn-shanghai-internal.aliyuncs.com
- bucket: dataworks-workshop
- AK ID: LTAINEhd4MZ8pX64
- AK Key: lXnzUngTSebt3SfLYxZxoSjGAK6IaF

3. Click **Test Connectivity**, and after the connectivity test passes, click **Finish** to save the configuration.

> **Note:**
> If the test connectivity fails, check your AK and the region in which the item is located. It is recommended to create the project in East China 2, and other regions do not guarantee network access.

- Add RDS Data Source

1. Select the **Data Integration** > **Data Source**Page, and click **Add Data Source**.
2. Select the data source type as MySQL, and fill in the configuration information.



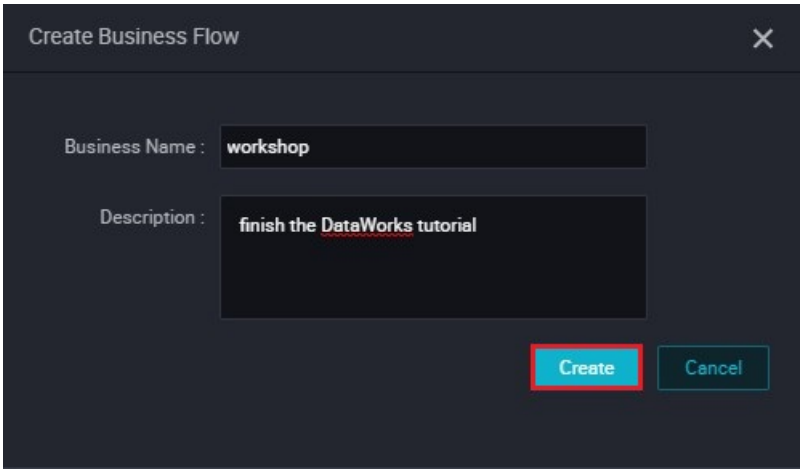
Parameters:

- Data source type: ApsaraDB for RDS
- Data source name: rds_workshop_log
- Data source description: RDS log data synchronization
- RDS instance name: rm-bp1z69dodhh85z9qa
- RDS instance buyer ID: 1156529087455811
- Database name: workshop
- Username/Password: workshop/workshop#2017

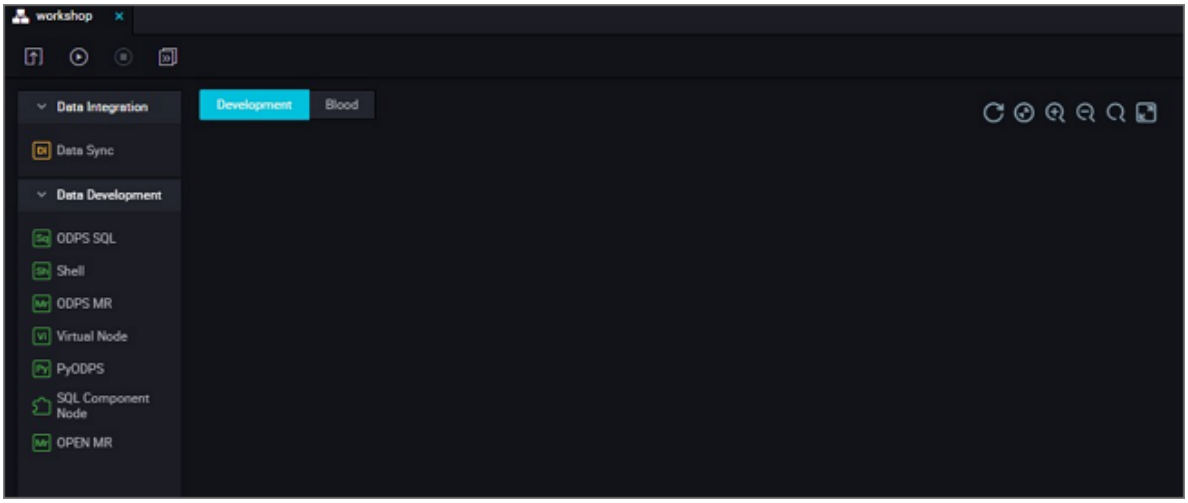
3. Click **Test Connectivity**, and after the connectivity test passes, click **Finish** to save the configuration.

Create a Business Flow

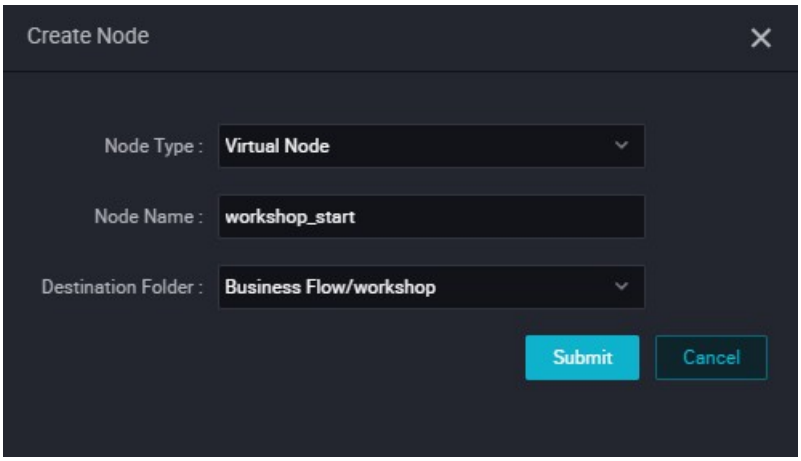
1. Right-click **Business Flow** under **Data Development**, select **Create Business Flow**.
2. Fill in the Business Flow name and description.

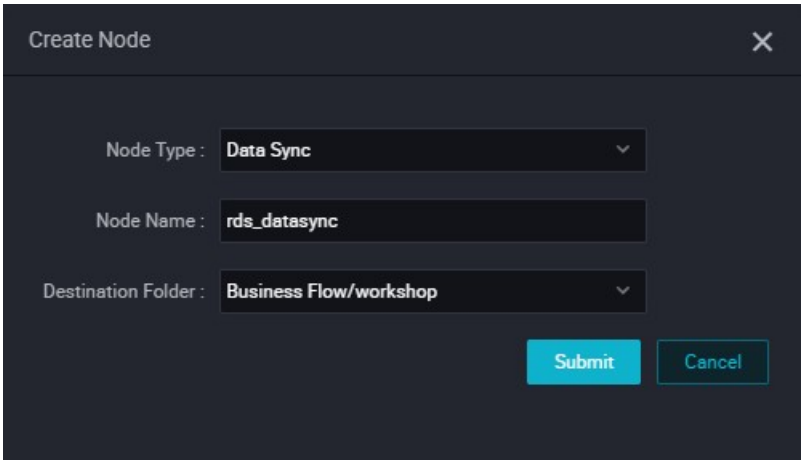


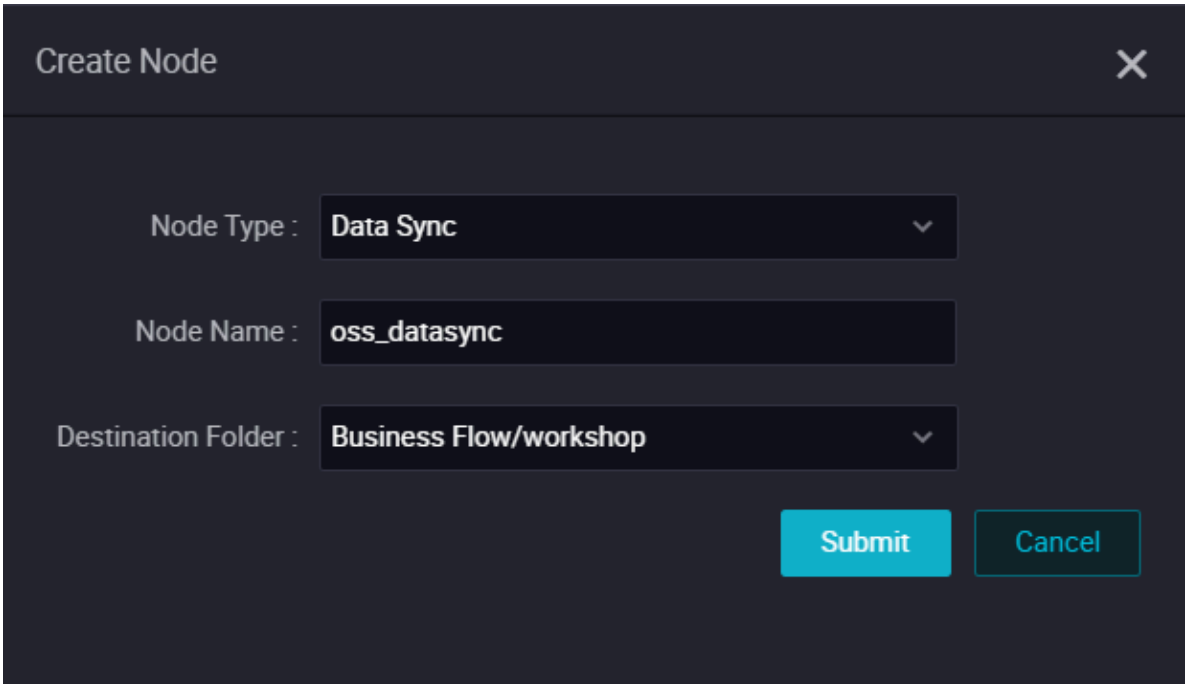
3. Click **Create** to complete the creation of the Business Flow.



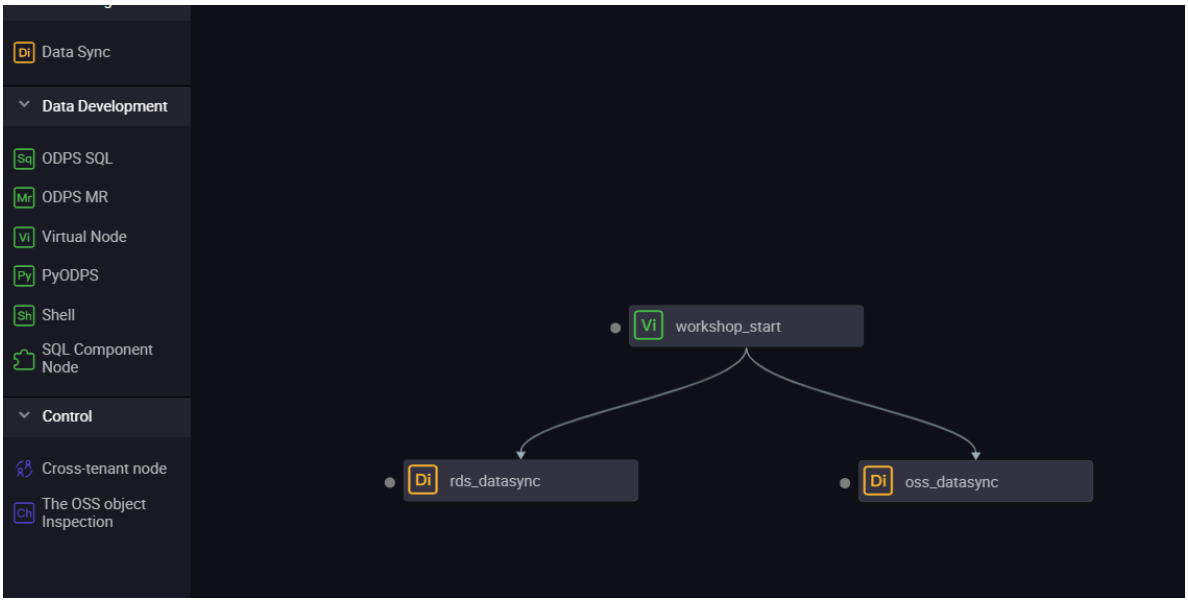
4. Enter the Business Flow Development Panel and drag a virtual node and two data sync nodes (oss_datasync and rds_datasync) into the Panel.







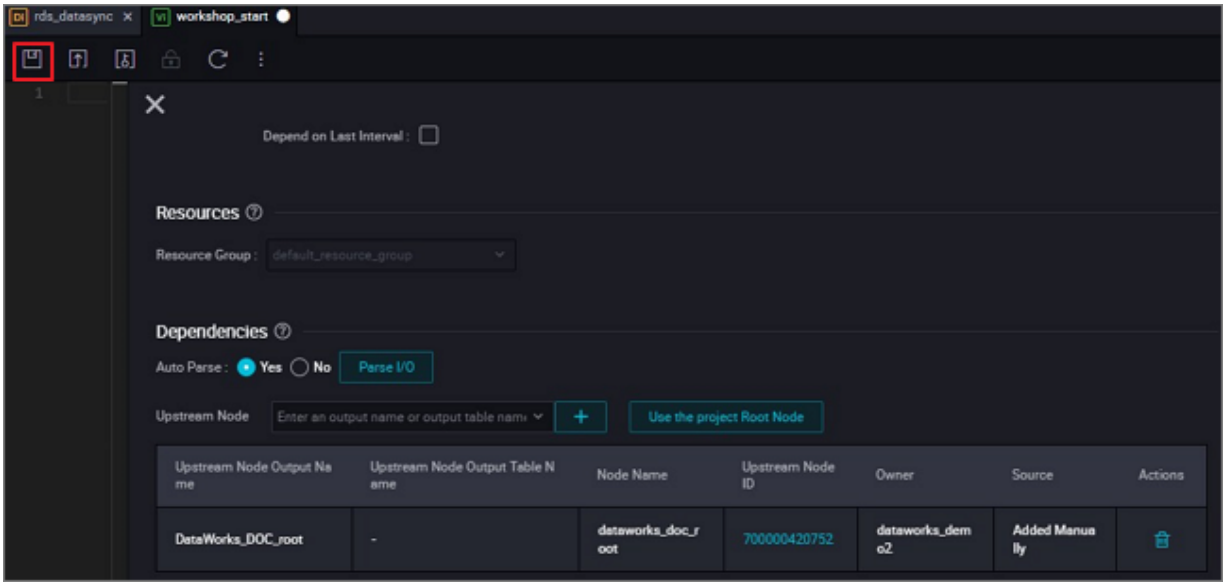
5. Drag the connection to set the workshop_start node to the upstream of both data synchronization nodes.



Configure workshop_start task

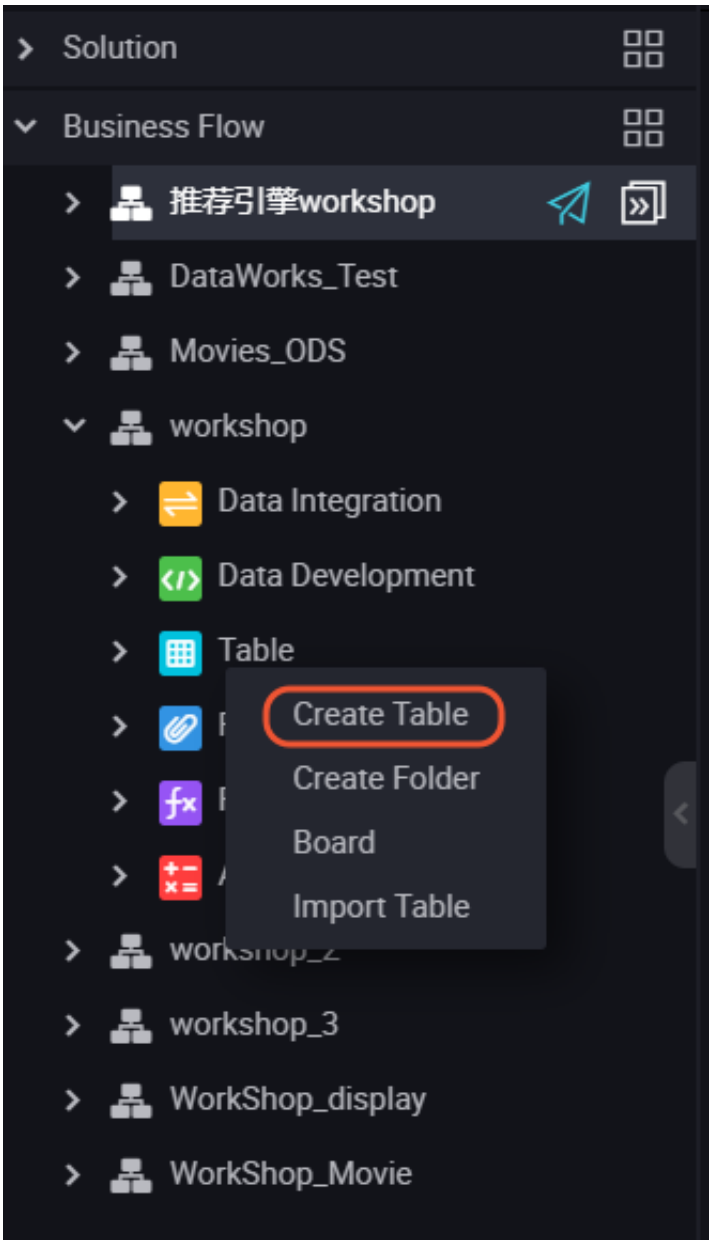
Since the new version sets the input and output nodes for each node, you need to set an input for the workshop_start node, the virtual node in the Business Flow can be set to the upstream node as the project root node, the project root node is generally named project name _ root.

You can configure it by clicking **Schedule**. When the task configuration is complete, click **Save**.

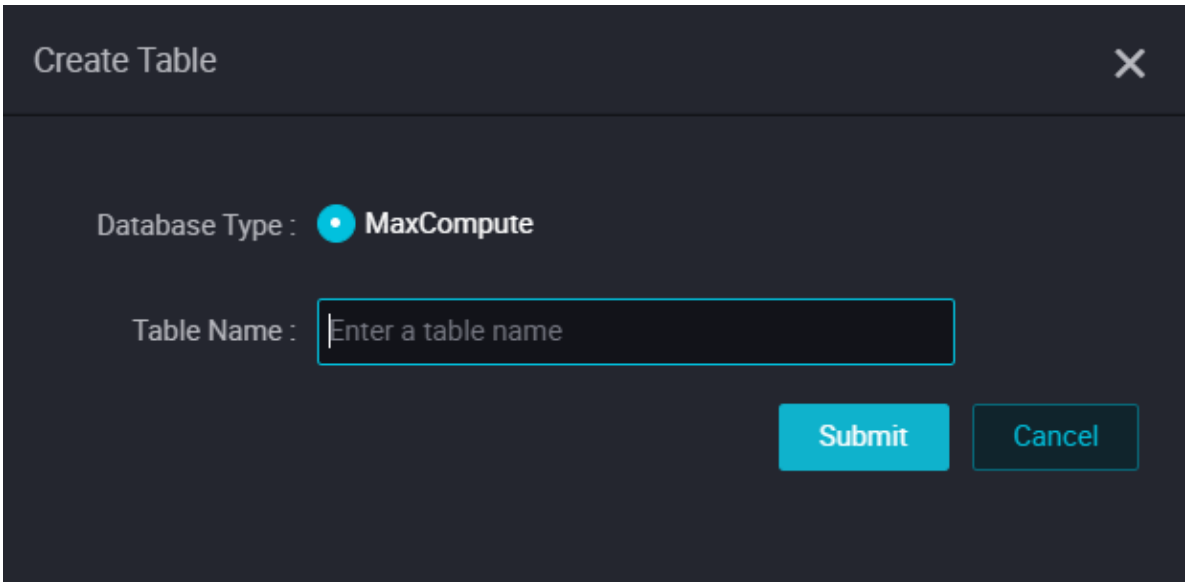


Create Table

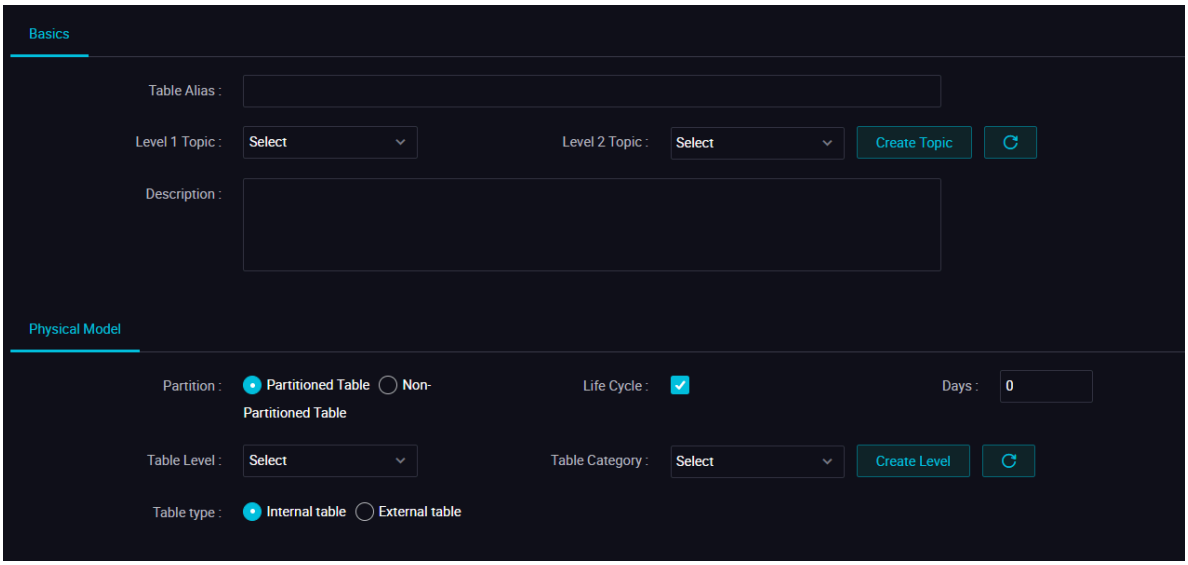
1. Right-click **Table** and choose **Create Table**.



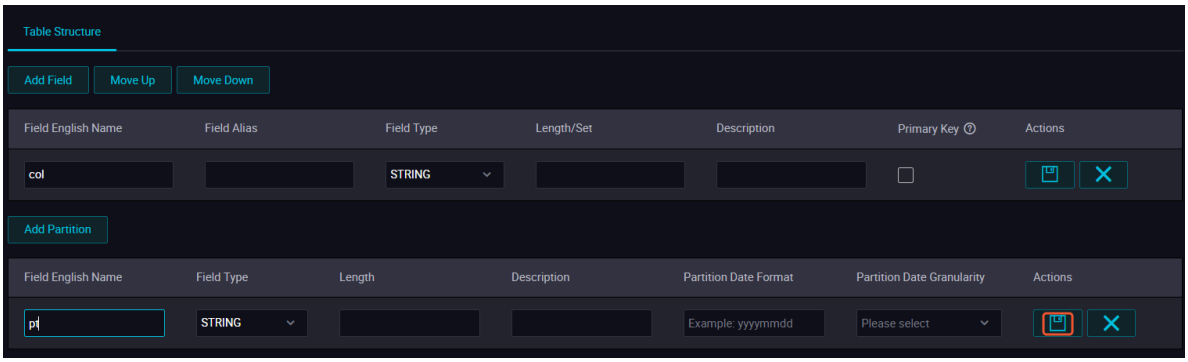
2. Type in **Table Name**(ods_raw_log_d and ods_user_info_d) for oss logs and RDS respectively.



3. Type in your **Table Alias** and choose **Partitioned Table**.



4. Type in the field and partition information,click **Submit to Development Environment** and **Submit to Production Environment**.



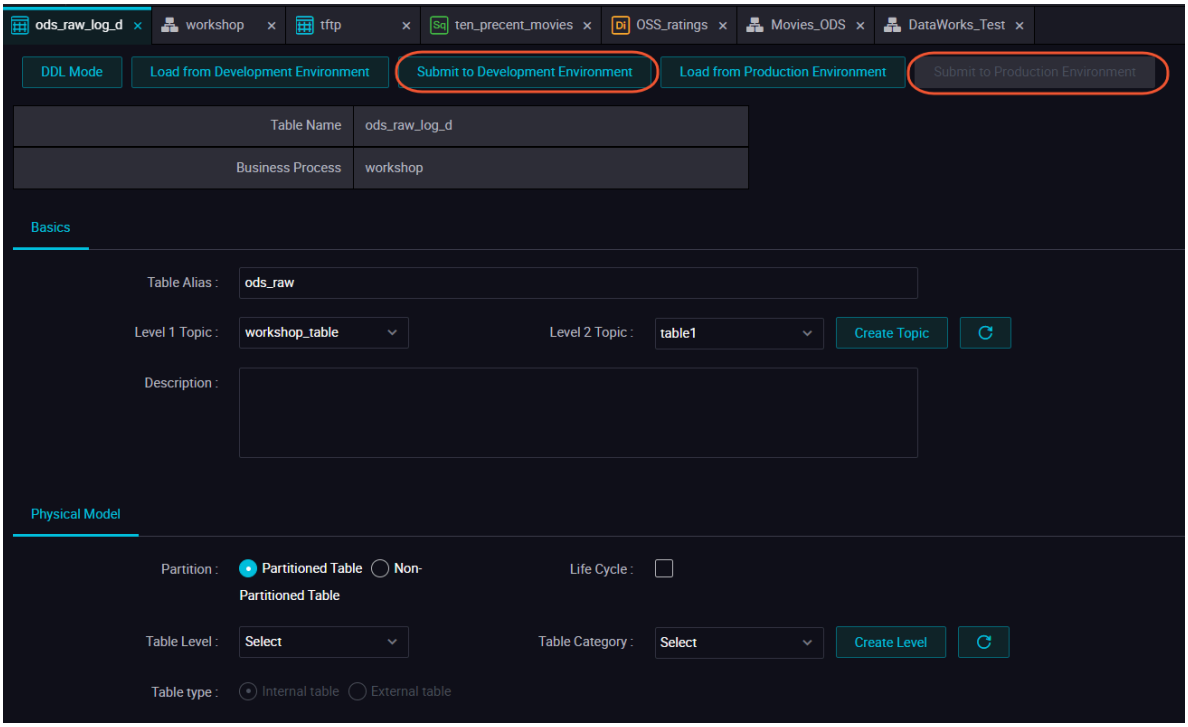
You can also click **DDL Mode**, use the following SQL statements to create tables.

```
//Creates a target table for oss logs
```

```
CREATE TABLE IF NOT EXISTS  ods_raw_log_d (
  Col_string
)
PARTITIONED BY (
  dt STRING
);

//Creates a target table for RDS
CREATE TABLE IF NOT  EXISTS ods_user_info_d (
  uid STRING COMMENT 'User ID',
  gender STRING COMMENT 'Gender',
  age_range STRING COMMENT 'Age range',
  zodiac STRING COMMENT 'Zodiac'
)
PARTITIONED BY (
  dt STRING
);
```

5. Click **Submit to Development Environment** and **Submit to Production Environment**. You can configure both of the tables in this way.

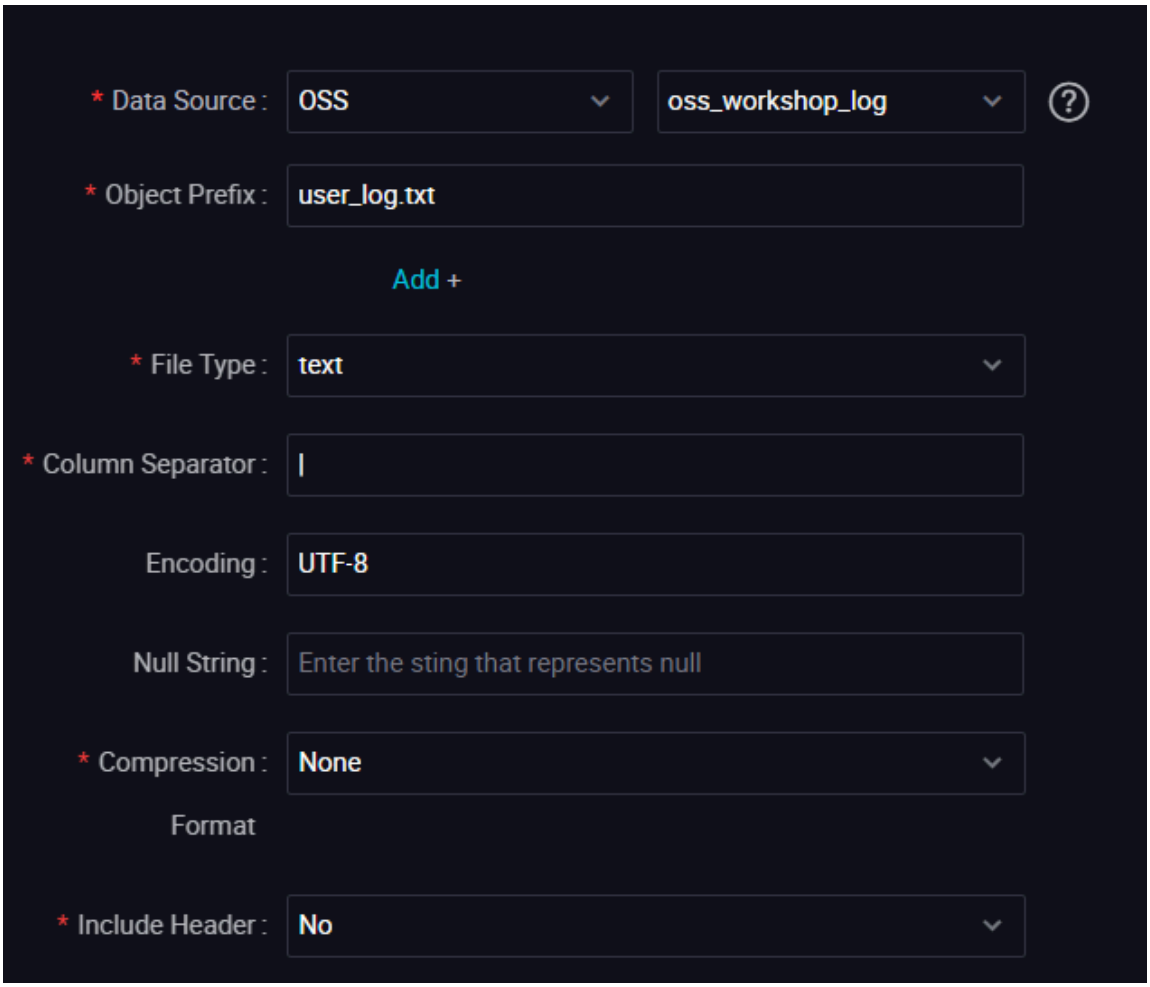


Configure the data synchronization task

- Configure the oss_datasync node

  1. Double-click the oss_datasync node node to go to the node configuration page.
  2. Select a data source.

     Select the data source as the maid in the oss data source.

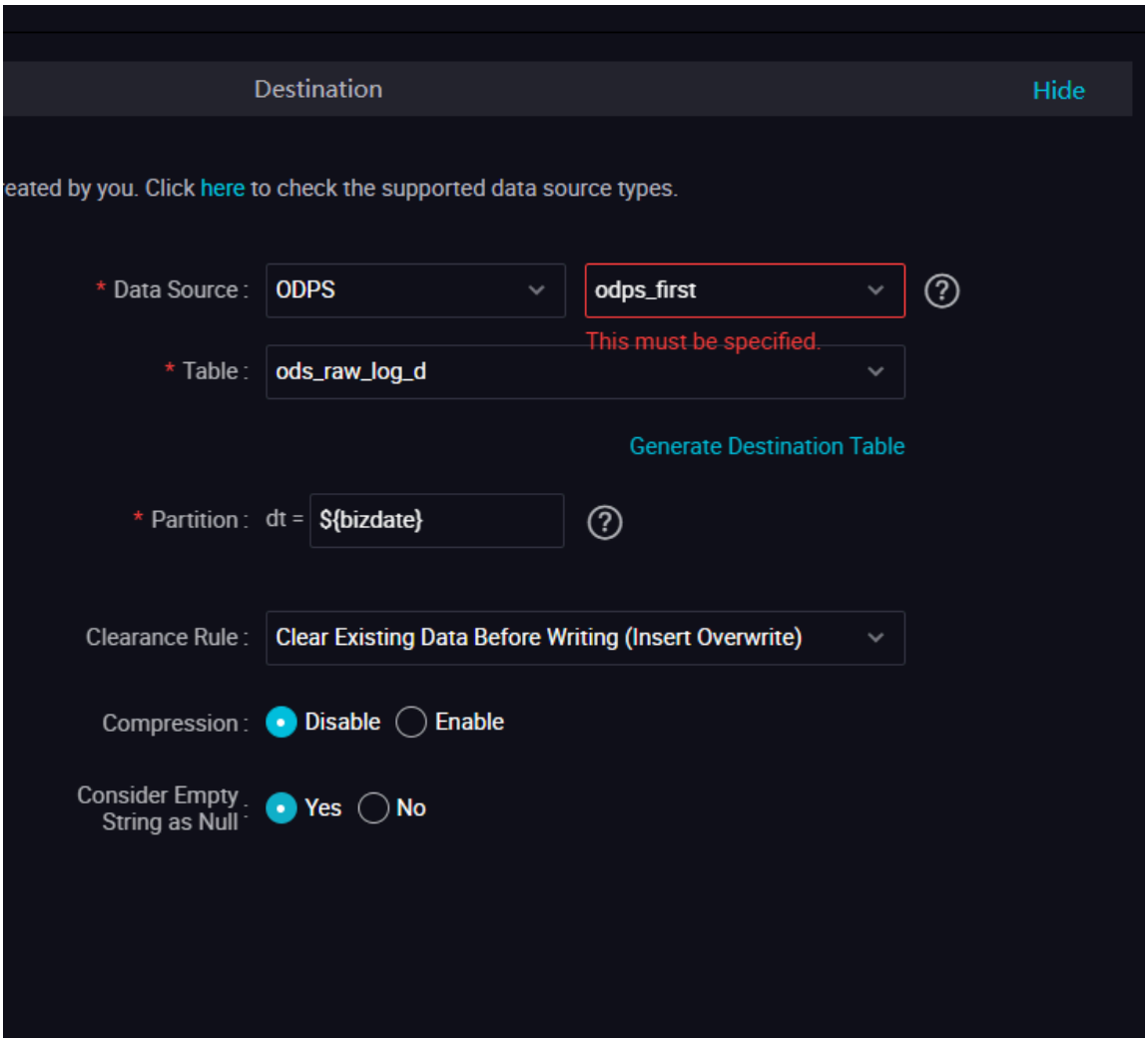


Parameters:

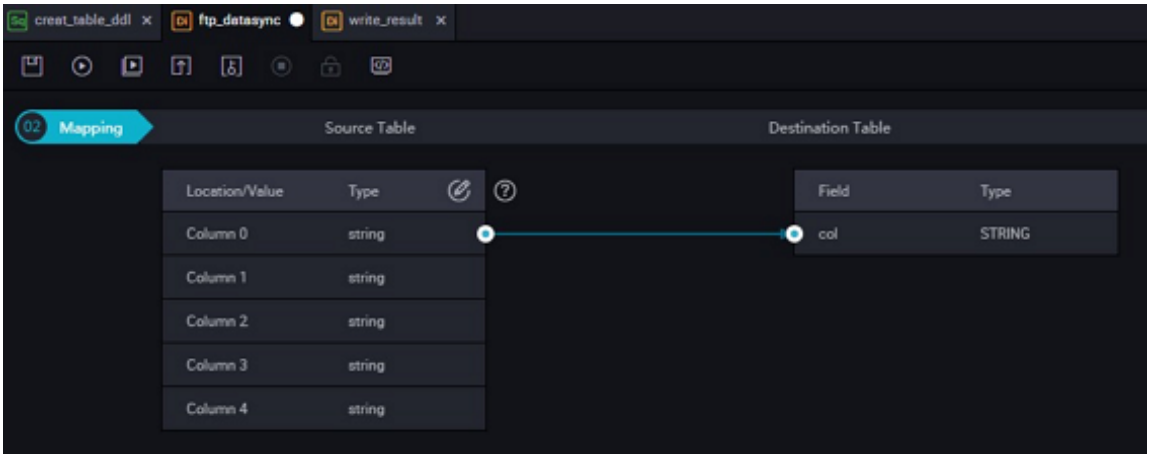
- Data source: oss_workshop_log
- Object Prefix: /user_log.txt
- Column Separator: |

3. Select data destination

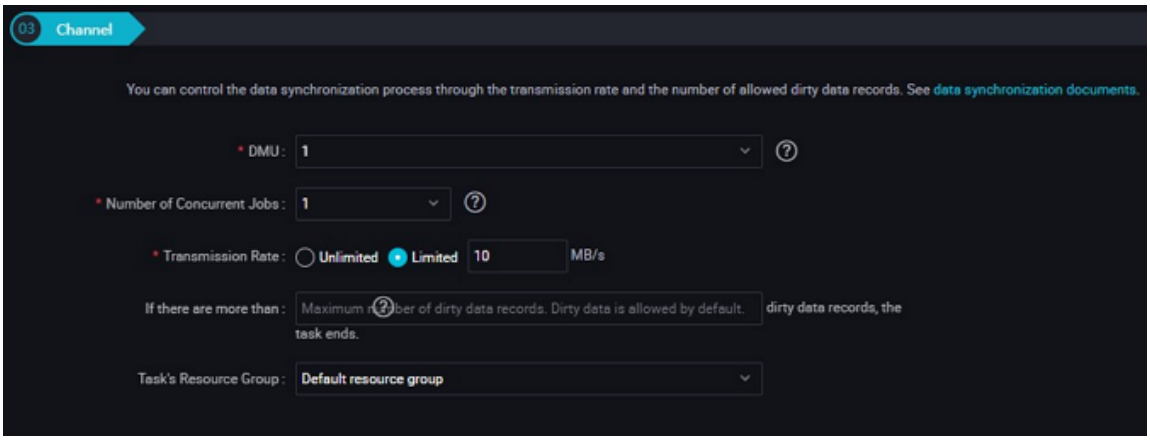
   Select the data destination is ods_raw_log_d in the odps_first data source. Both partition information and cleanup rules take the system default, the default configuration of the partition is ${bizdate}.



4. Configure the field mapping, connect the fields that you want to synchronize.



5. Configure **Transmission Rate** with a maximum operating rate of 10 Mb/s.

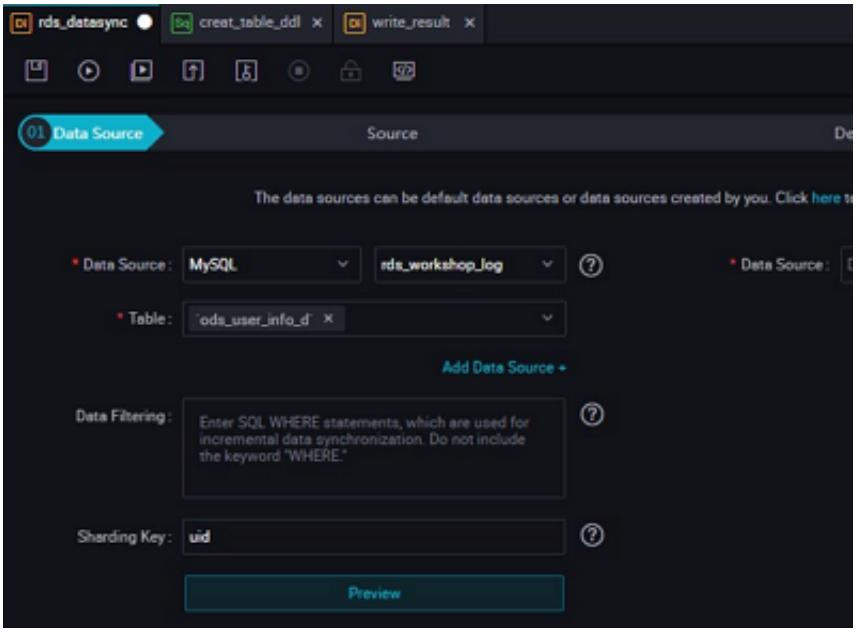


6. Verify that the current task is configured and can be modified. After the confirmation is correct, click **Save** in the upper left corner.
7. Closes the current task and returns to the Business Flow configuration panel.

- Configure the rds_datasync node Node

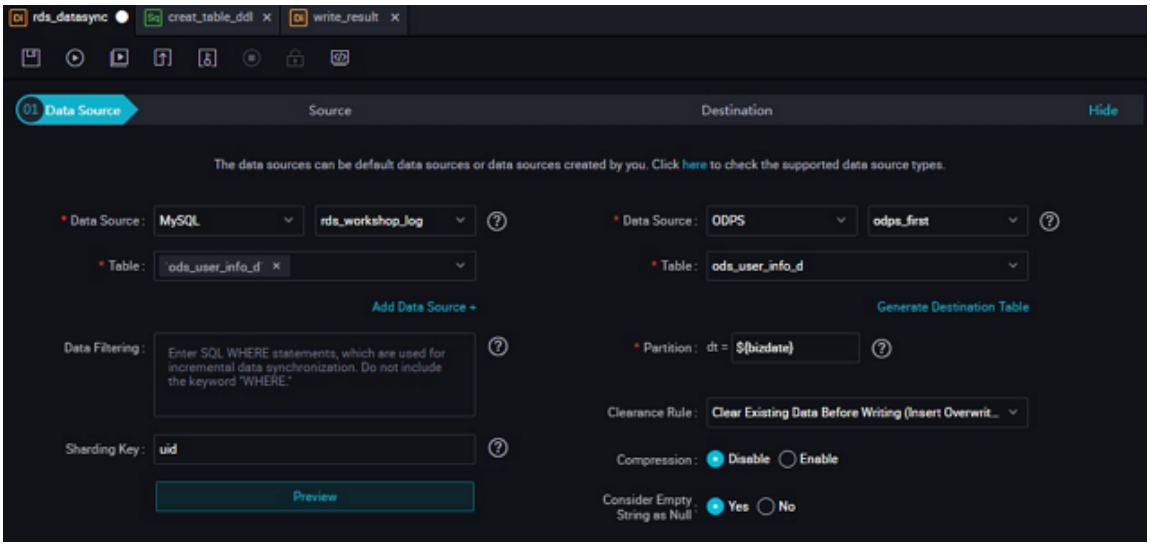
1. Double-click the rds_datasync node node to go to the node configuration page.
2. Select a data source.

   Select the data source that is located in the MySQL data source rds_workshop_log, and the table is named as ods_user_info_d, the split key uses the default to generate columns.

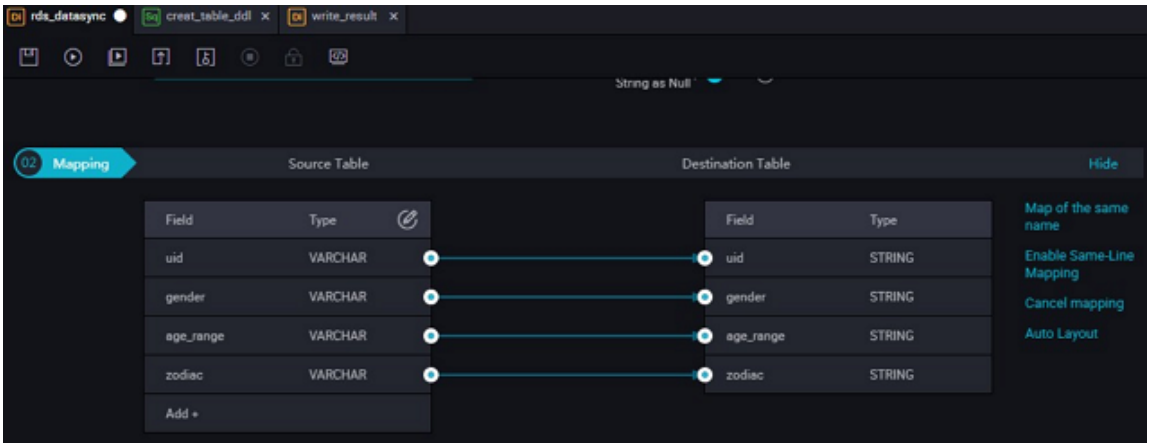


3. Select data destination

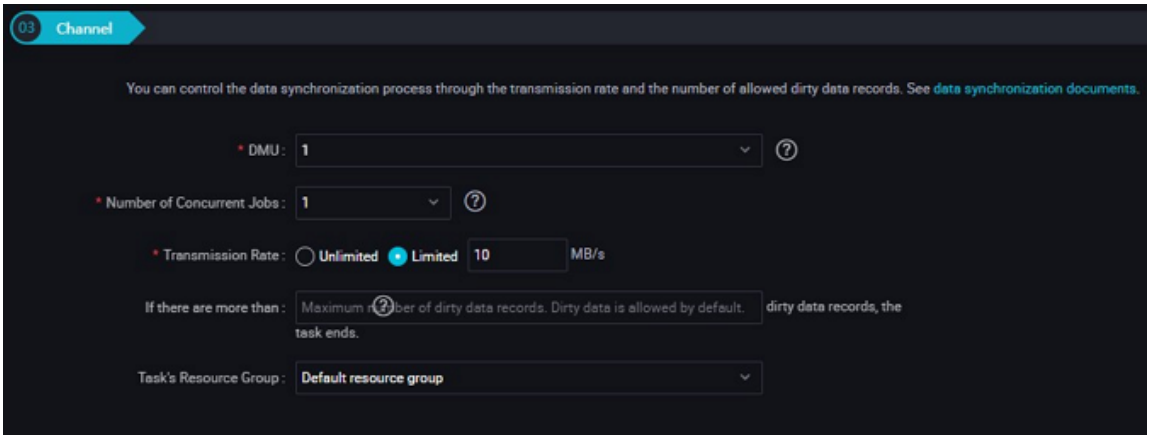
   Select the data destination ods_user_info_d in the data source named odps_first. Both partition information and cleanup rules take the system default, the default configuration of the partition is ${bizdate}.



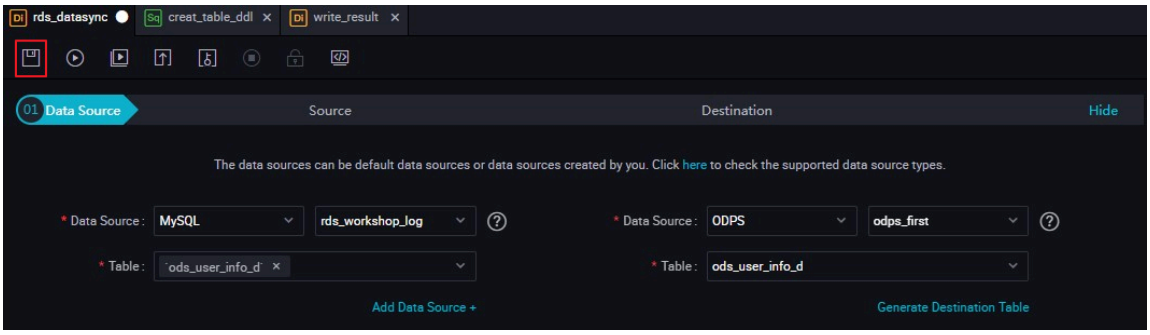
4. Configure the field mapping, default in association with the name mapping.



5. Configure **Transmission Rate** with a maximum operating rate of 10 Mb/s.



6. Verify that the current task is configured and can be modified. After the confirmation is correct, click **Save** in the upper left corner.



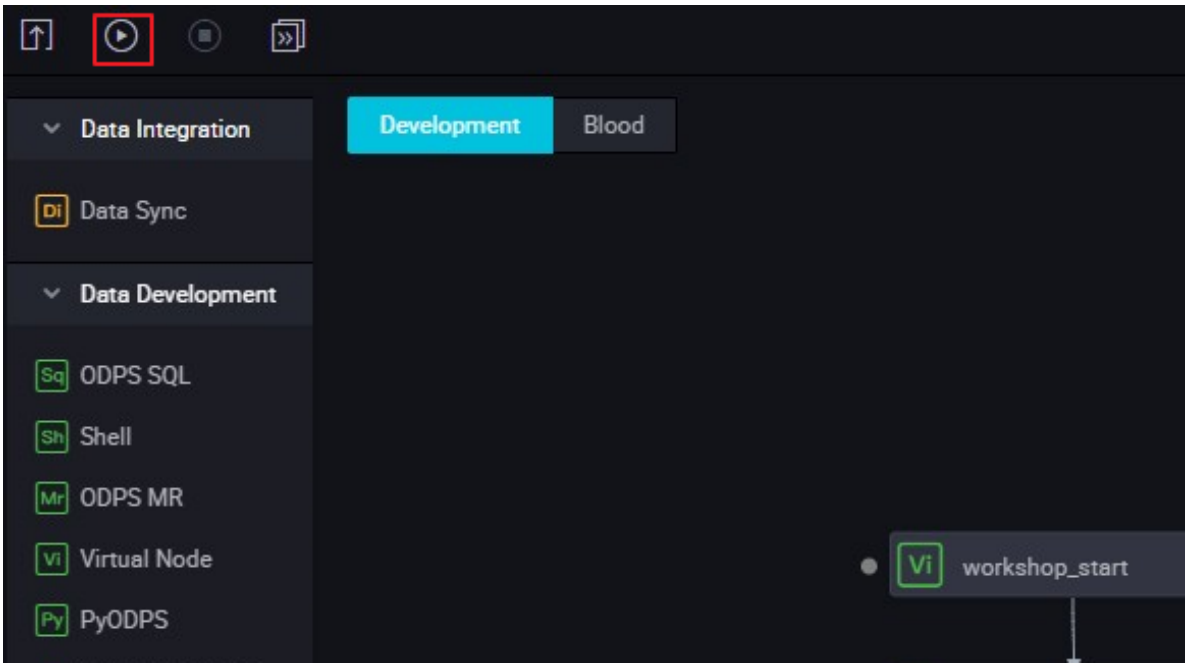
7. Closes the current task and returns to the Business Flow configuration panel.

Submit Business Flow tasks

1. Click **Submit** to submit the current Business Flow.
2. Select the nodes in the submit dialog box, and check the **Ignore Warning on I/O Inconsistency**, click **Submit**.

Run workflow task

1. Click **Run**.



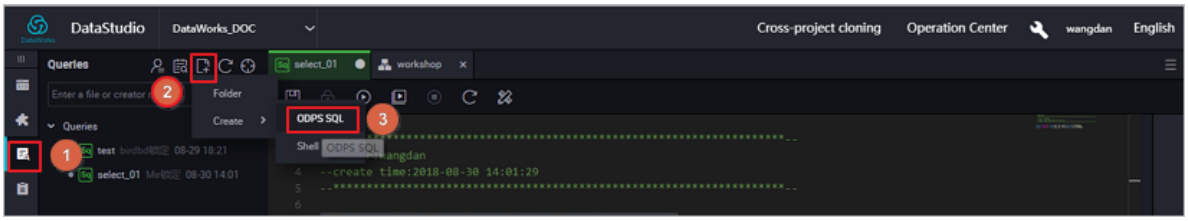
During a task run, you can view the run status.

2. Right-click the SQL task and select **view log**.
3. Right-click the OSS _ Data Synchronization task and select **view log**.
4. Right-click the RDS _ Data Synchronization task and select **view log**.

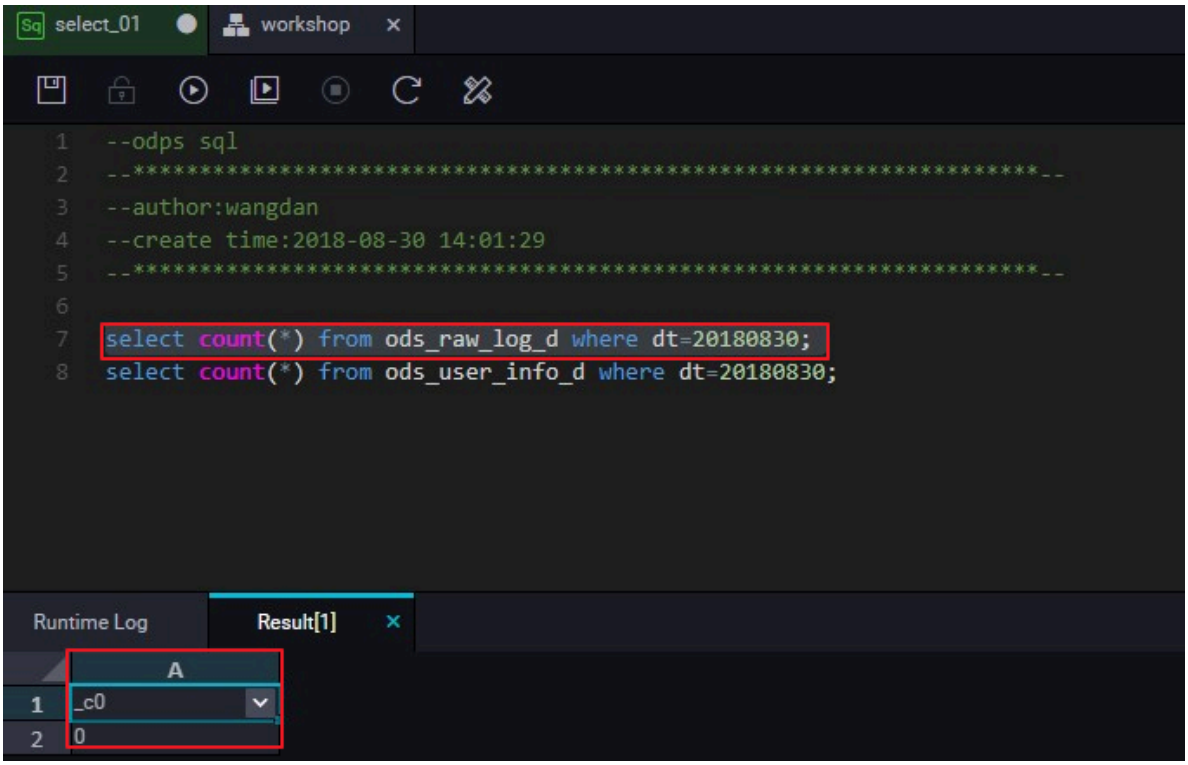
Check if the data is successfully imported into MaxCompute

1. Click **temporary query** in the left-hand navigation bar.

2. Select **New** > **ODPS SQL**.



3. Write and execute SQL statement to check the entries imported into ods_raw_log_d.



4. Also write and execute SQL statements to view the number of imported ods_user_info_d records.

> **Note:**
> The SQL statement is as follows, where the partition columns need to be updated to the business date, if the task runs on a date of 20180717, the business date is 20180716.

```
Check that data was written to MaxCompute
select count(*) from ods_raw_log_d where dt=business date;
select count(*) from ods_user_info_d where dt=business date;
```

Next step

Now that you've learned how to synchronize the log data, complete the data acquisition, you can continue with the next tutorial. In this tutorial, you will learn how to calculate and analyze the collected data. For more information, see *Data processing: user portraits*.

2.3 Data processing: user portraits

This article shows you how to process log data that has been collected into MaxCompute through dataworks.

Note:
Before you begin this experiment, please complete the operation in*Data acquisition: log data upload*.

Create data tables

You can refer to *Data acquisition: log data upload* to create data tables.

- Create ods_log_info_d table

  1. Right click **Table** in the workshop business flow. Click **Create Table** and enter the table's name ods_log_info_d. You can then click DDL Mode to type in the table creation SQL statements.

     The following are table creation statements:

```
CREATE TABLE IF NOT EXISTS ods_log_info_d (
  IP string comment 'IP address ',
  uid STRING COMMENT 'User ID',
  Time string comment 'time': MI: ss ',
  Status string comment 'server return status code ',
  Bytes string comment 'the number of bytes returned to the Client
 ',
  Region string comment ', get' from IP ',
  Method string comment 'HTTP request type ',
  URL string comment 'urle ',
  Protocol string comment 'HTTP Protocol version number ',
  Referer string comment 'source ures ',
  Device string comment 'terminal type ',
  Identity string comment 'Access type crawler feed user unknown'
)
PARTITIONED BY (
  dt STRING
);
```

  2. Click **Submit to Development Environment** and **Submit to Production Environment**.

- Create dw_user_info_all_d table

  The method of creating a new report table is identical to that of a table statement as follows:

```
-- Create a copy table
CREATE TABLE IF NOT EXISTS dw_user_info_all_d (
  uid STRING COMMENT 'User ID',
  gender STRING COMMENT 'Gender',
  age_range STRING COMMENT 'Age range',
  zodiac STRING COMMENT 'Zodiac sign'
  Region string comment ', get' from IP ',
  Device string comment 'terminal type ',
```

```
  Identity string comment 'Access type crawler feed user unknown ',
  Method string comment 'HTTP request type ',
  URL string comment 'url ',
  Referer string comment 'source url ',
  Time string comment 'time': MI: ss'
)
PARTITIONED BY (
  DT string
);
```

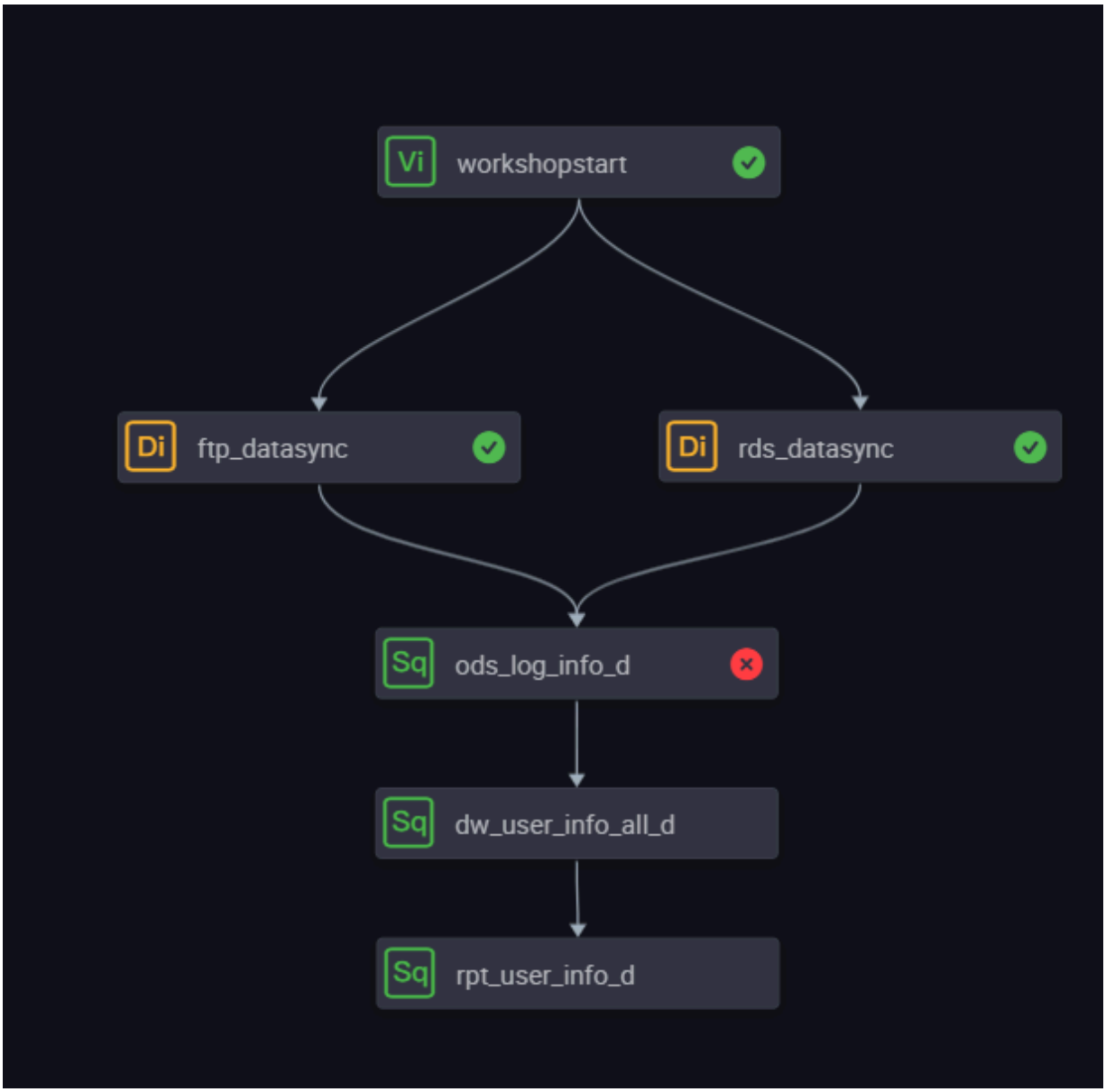
- Create rpt_user_info_d table

  The following are table creation statements:

```
-- Create a copy table
Create Table if not exists rpt_user_info_d(
  uid STRING COMMENT 'User ID',
  Region string comment ', get' from IP ',
  Device string comment 'terminal type ',
  PV bigint comment 'cv ',
  gender STRING COMMENT 'Gender',
  age_range STRING COMMENT 'Age range',
  zodiac STRING COMMENT 'Zodiac sign'
)
PARTITIONED BY (
  DT string
);
```

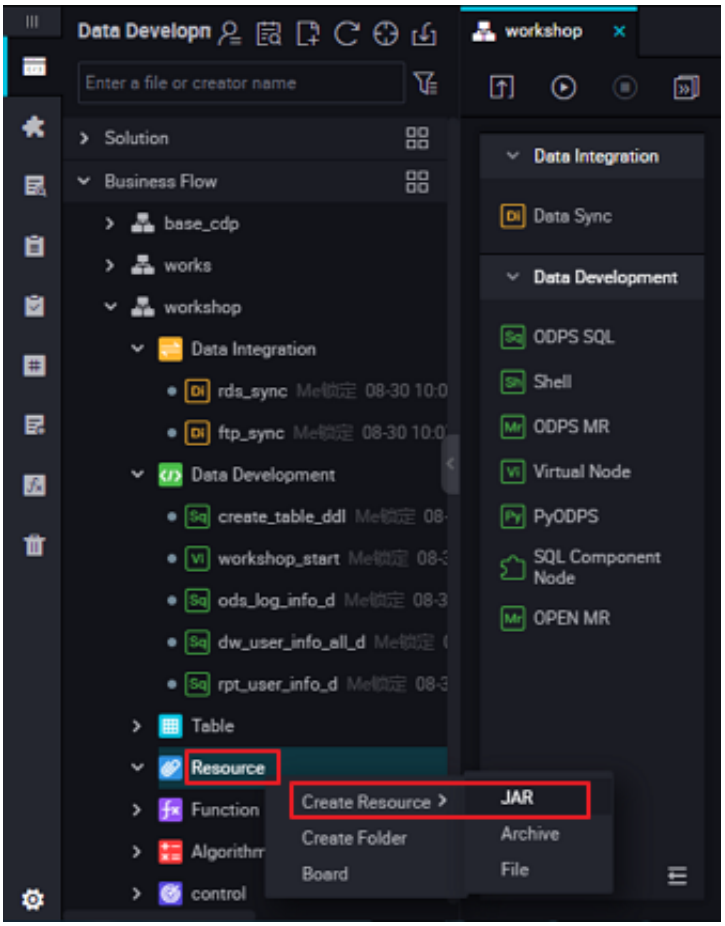
Business Flow Design

Open the Workshop Business Flow and drag three ODPS SQL nodes amed as "ods_log_info_d、dw_user_info_all_d、rpt_user_info_d" into the canvas, n, and configure dependencies.

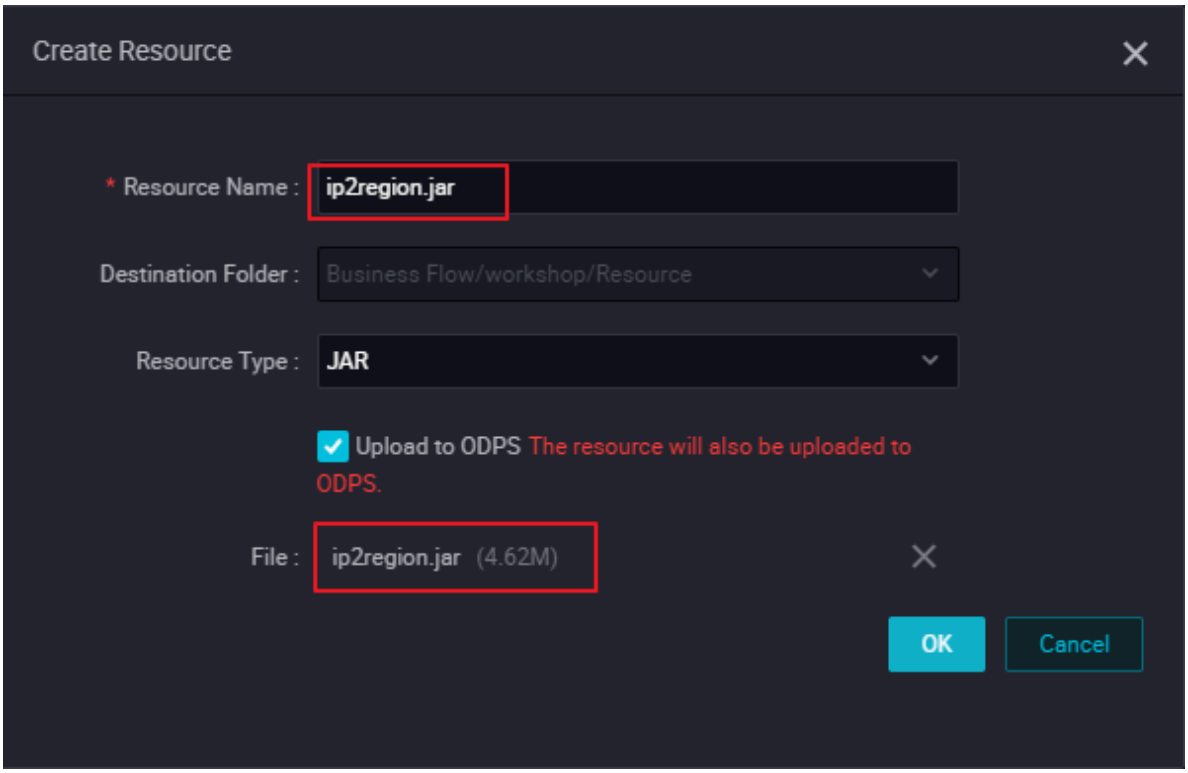


Creating user-defined functions

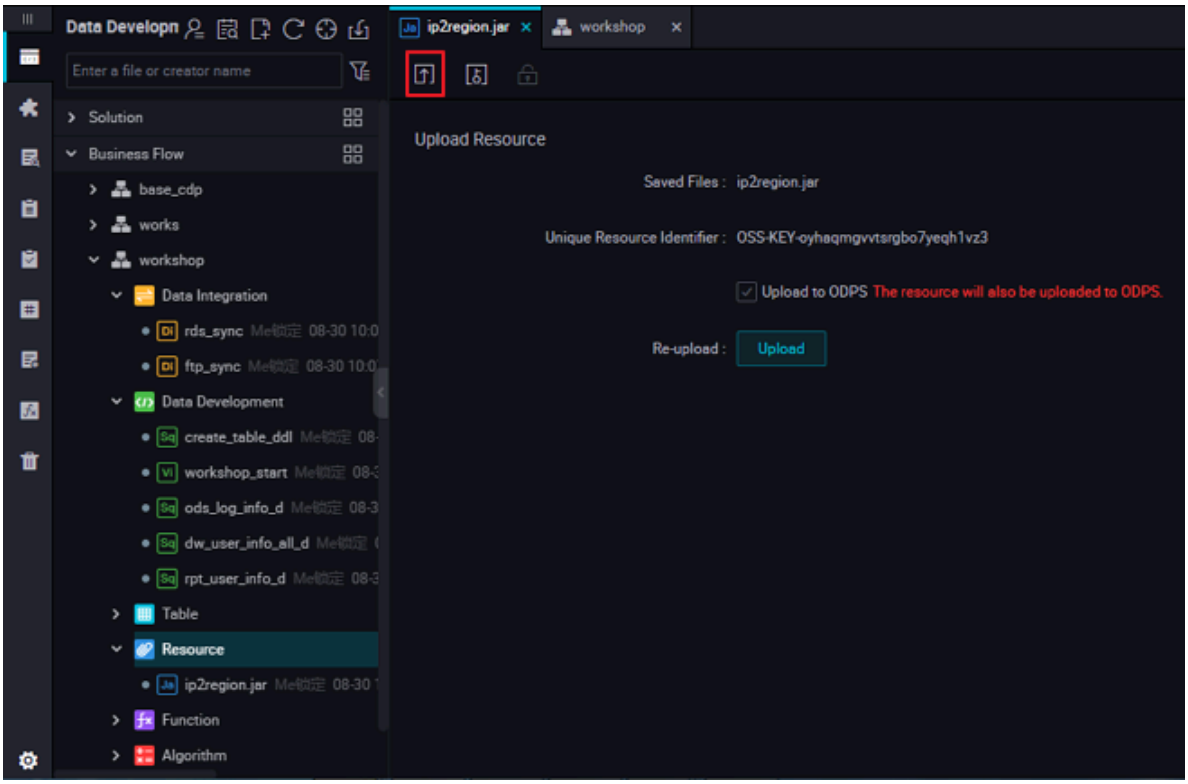
1. Download *ip2region.jar*.
2. Right-click **Resource**, and select **Create Resource** > **jar**.



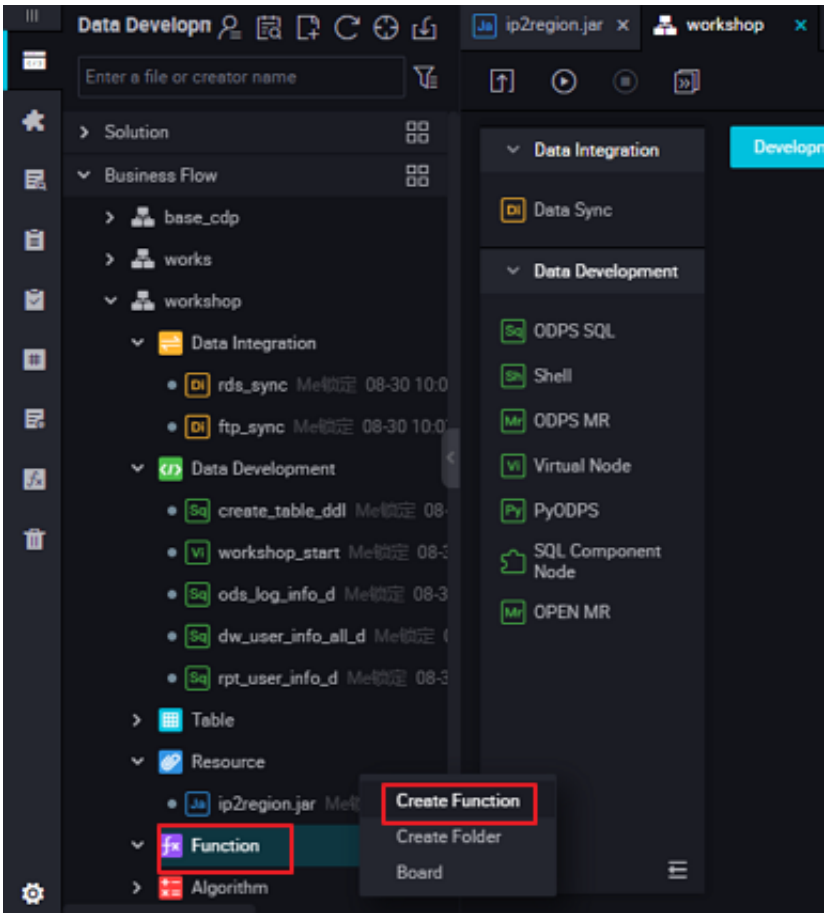
3. Click **Select File**, select ip2region. jar that has been downloaded locally, and click **OK**.



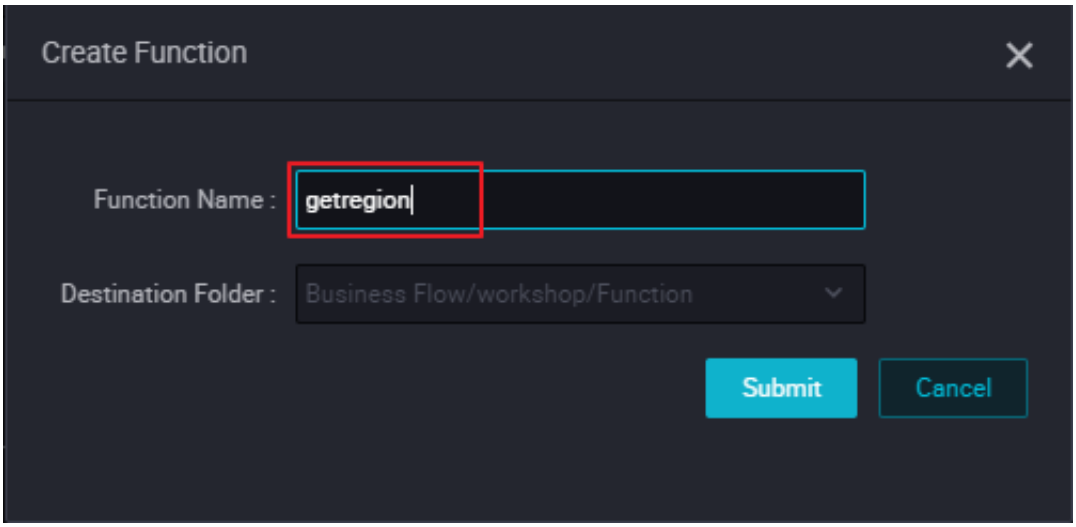
4. After the resource has been uploaded to dataworks, click **Submit**.



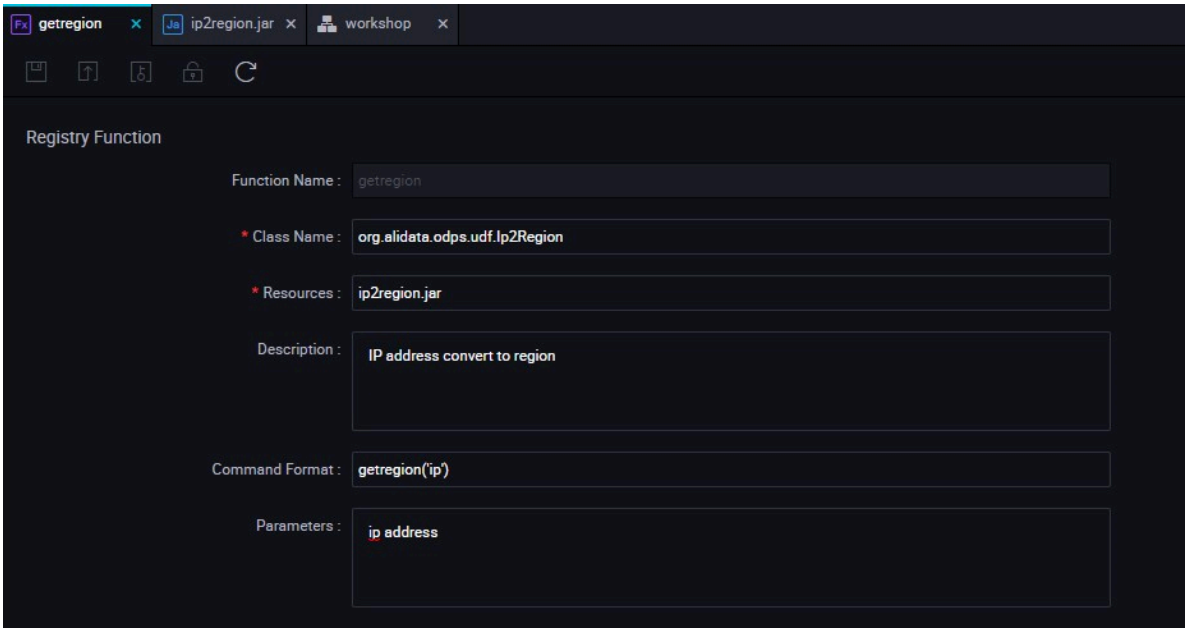
5. Right-click a **function** and select **Create Function**.



6. Enter the function name getregion, select the Business Flow to which you want to belong, and click **Submit**.



7. Enter the function configuration in the **Registry Function** dialog box, specify the class name, description, command format, and parameter description.



Parameters:

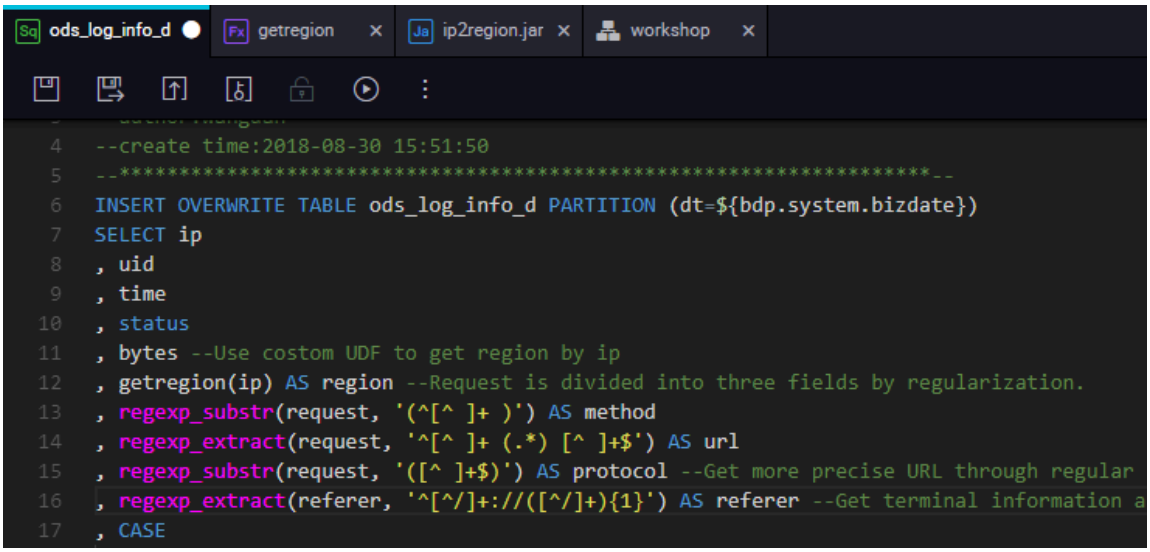
- Function Name: getregion
- Class Name: org. alidata. ODPS. UDF. ip2region
- Resource list: ip2region. Jar
- Description: IP address translation area
- Command Format: getregion ('IP ')
- Parameter description: IP Address

8. Click **Save** and **submit**.

Configure ODPS SQL nodes

- Configure ods_log_info_d Node

1. Double-click the ods_log_info_d node to go to the node configuration page and write the processing logic.

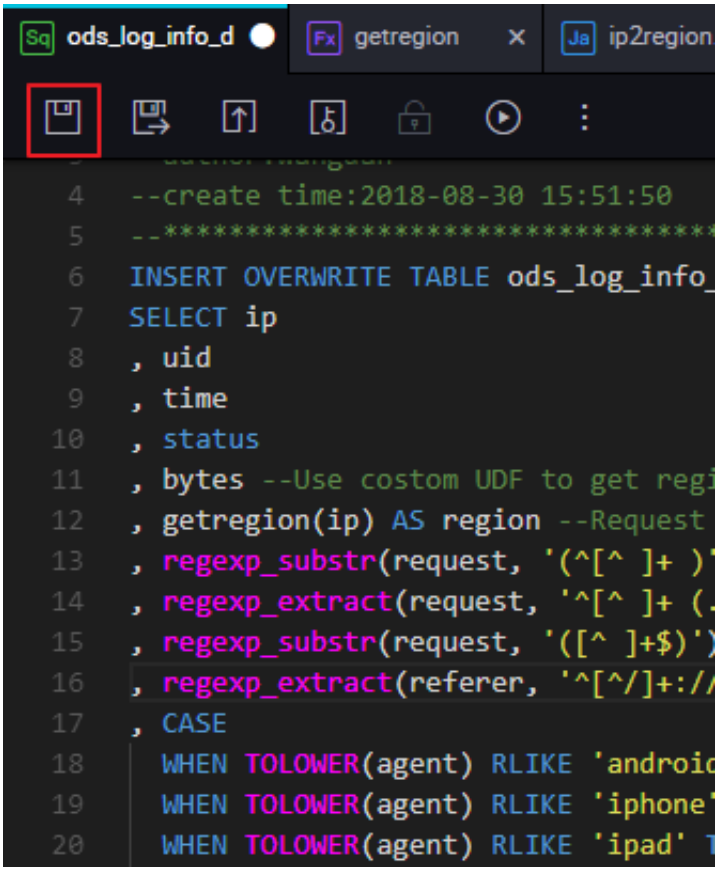


The SQL logic is as follows:

```
INSERT OVERWRITE TABLE ods_log_info_d PARTITION (dt=${bdp.system.
bizdate})
SELECT ip
  , uid
  , Time
  , Status
  , Bytes-use a custom UDF to get a locale over IP
  , Getregion (IP) as region -- the request difference is divided
into three fields through the regular
  , Regexp_substr (request, '(^ [^] +)') as Method
  ), Regexp_extract (request, '^ [^] + (. *) [^] + $ ') As URL
  ), FIG (request, '([^] + $ )') as protocol-get more precise URLs
 with regular clear refer
  , (Referer, '^ [^/] +: /([^/] +) {1 }') as Referer-Get terminal
information and access form through agent
  , Case
    When tolower (agent) rlike 'android 'then 'android'
    WHEN TOLOWER(agent) RLIKE 'iphone' THEN 'iphone'
    WHEN TOLOWER(agent) RLIKE 'ipad' THEN 'ipad'
    WHEN TOLOWER(agent) RLIKE 'macintosh' THEN 'macintosh'
    WHEN TOLOWER(agent) RLIKE 'windows phone' THEN 'windows_phone'
    WHEN TOLOWER(agent) RLIKE 'windows' THEN 'windows_pc'
    ELSE 'unknown'
  End as Device
  , Case
    WHEN TOLOWER(agent) RLIKE '(bot|spider|crawler|slurp)' THEN '
crawler'
    WHEN TOLOWER(agent) RLIKE 'feed'
    OR regexp_extract(request, '^[^ ]+ (. *) [^ ]+$') RLIKE 'feed
' THEN 'feed'
    WHEN TOLOWER(agent) NOT RLIKE '(bot|spider|crawler|feed|slurp
)'
    AND agent RLIKE '^[Mozilla|Opera]'
```

```
    AND regexp_extract(request, '^[^ ]+ (. *) [^ ]+$') NOT RLIKE '
feed' THEN 'user'
    ELSE 'unknown'
  END AS identity
  FROM (
    SELECT SPLIT(col, '##@@')[0] AS ip
    , SPLIT(col, '##@@')[1] AS uid
    , SPLIT(col, '##@@')[2] AS time
    , SPLIT(col, '##@@')[3] AS request
    , SPLIT(col, '##@@')[4] AS status
    , SPLIT(col, '##@@')[5] AS bytes
    , Split (cola, '# @') [6] As Referer
    , SPLIT(col, '##@@')[7] AS agent
  FROM ods_raw_log_d
  Where dt = $ {BDP. system. Date}
) a;
```

2. Click **Save**.



- Configure dw_user_info_all_d Node

1. Double-click the dw_user_info_all_d node to go to the node configuration page and write the processing logic.

   The SQL logic is as follows:

```
INSERT OVERWRITE TABLE dw_user_info_all_d PARTITION (dt='${bdp.
system.bizdate}')
SELECT COALESCE(a.uid, b.uid) AS uid
  , b.gender
  , b.age_range
  , B. flavdiac
```

```
  , a.region
  , a.device
  , a.identity
  , a.method
  , a.url
  , a.referer
  , a.time
FROM (
  SELECT *
  From fig
  WHERE dt = ${bdp.system.bizdate}
) a
LEFT OUTER JOIN (
  SELECT *
  FROM ods_user_info_d
  WHERE dt = ${bdp.system.bizdate}
) b
ON a.uid = b.uid;  WHERE dt = ${bdp.system.bizdate}
) a;
```

2. Click **Save**.

- Configure a rpt_user_info_d Node

1. Double-click the fig node to go to the node configuration page and write the processing logic.

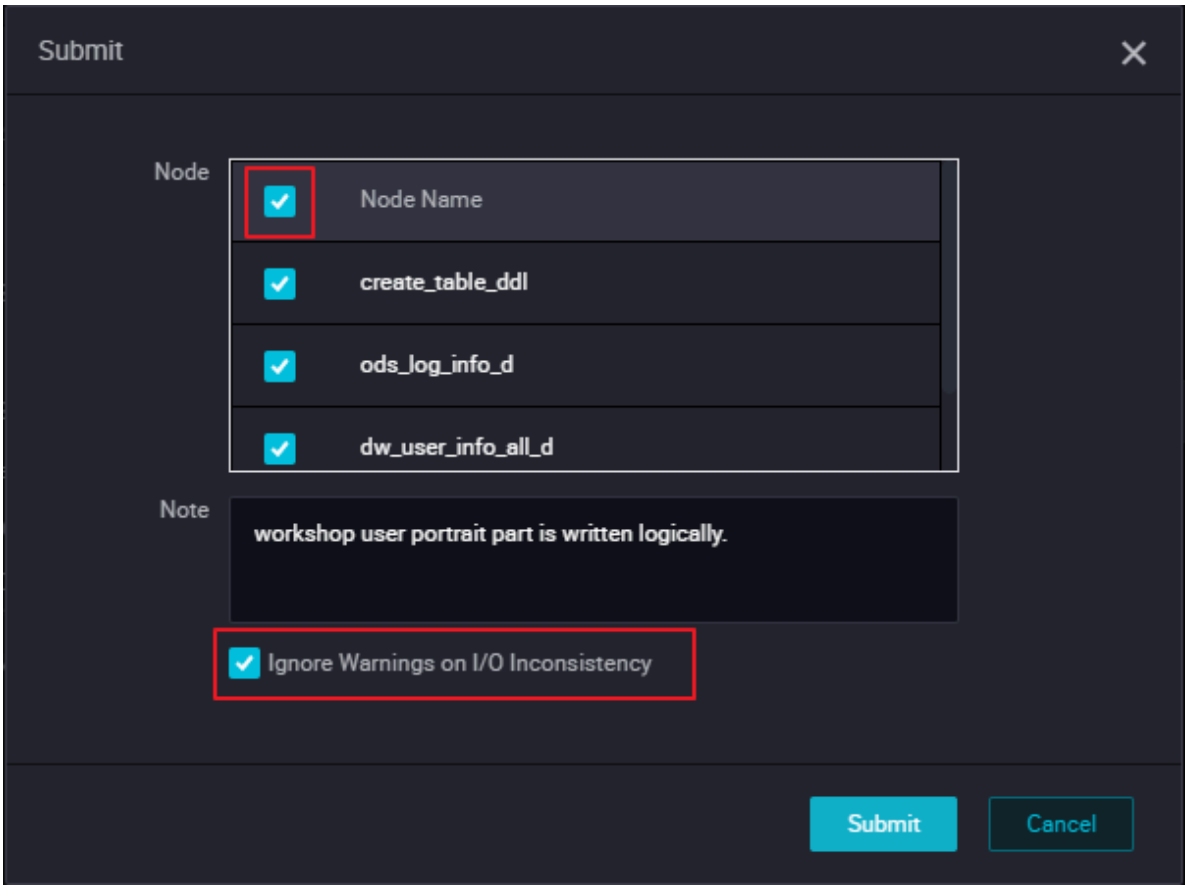
   The SQL logic is as follows:

```
INSERT OVERWRITE TABLE rpt_user_info_d PARTITION (dt='${bdp.system
.bizdate}')
SELECT uid
  , MAX(region)
  , MAX(device)
  , COUNT(0) AS pv
  , MAX(gender)
  , MAX(age_range)
  , MAX(zodiac)
FROM dw_user_info_all_d
WHERE dt = ${bdp.system.bizdate}
GROUP BY uid;p.system.bizdate}
) a;
```

2. Click **Save**.

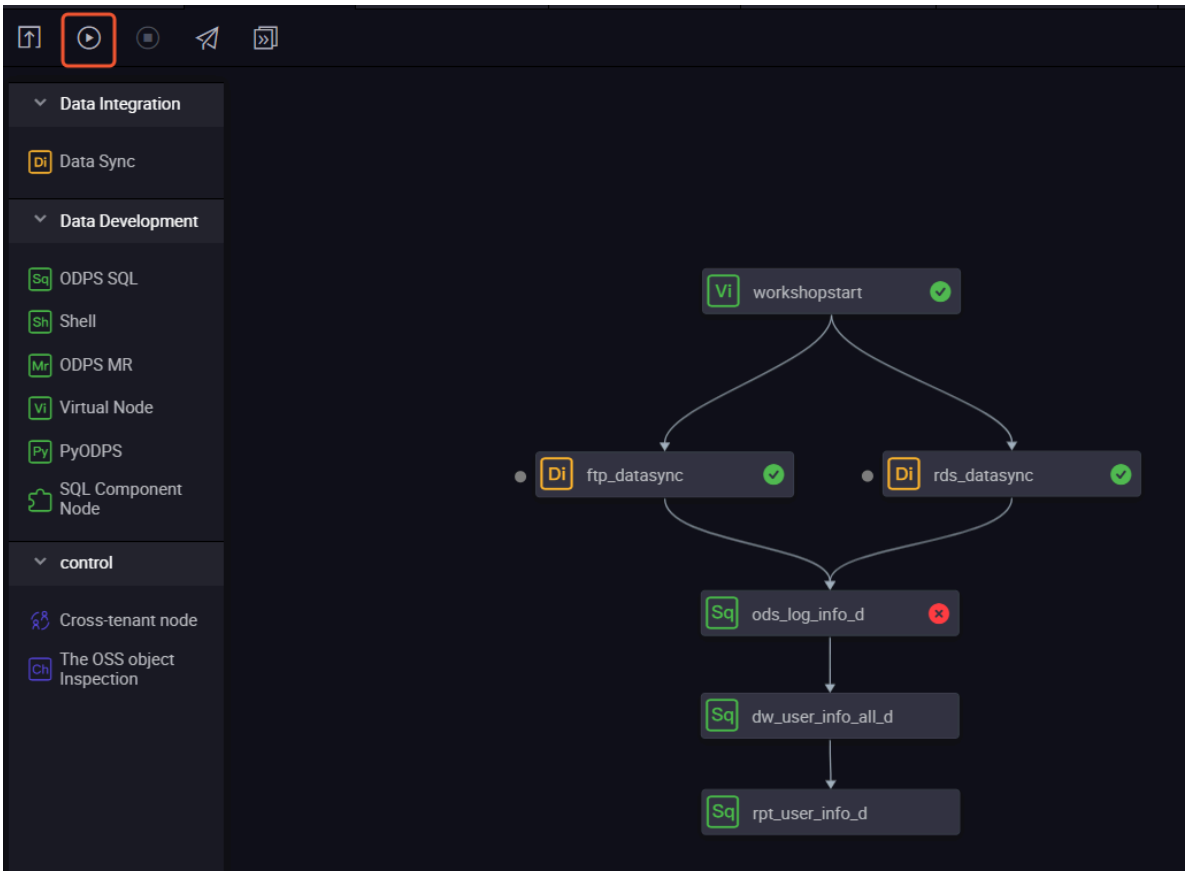
Submitting Business Flows

1. Click **Submit** to submit the node tasks that have been configured in the Business Flow.
2. Select the nodes that need to be submitted in the **Submit**dialog box, and check the **Ignore Warnings on I/O Inconsistency**, click **Submit**.

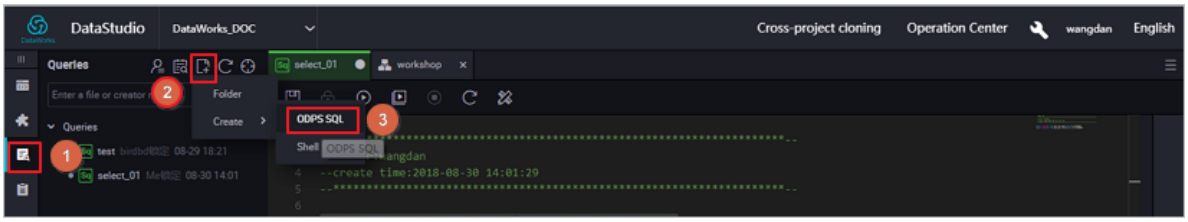


Running Business Flows

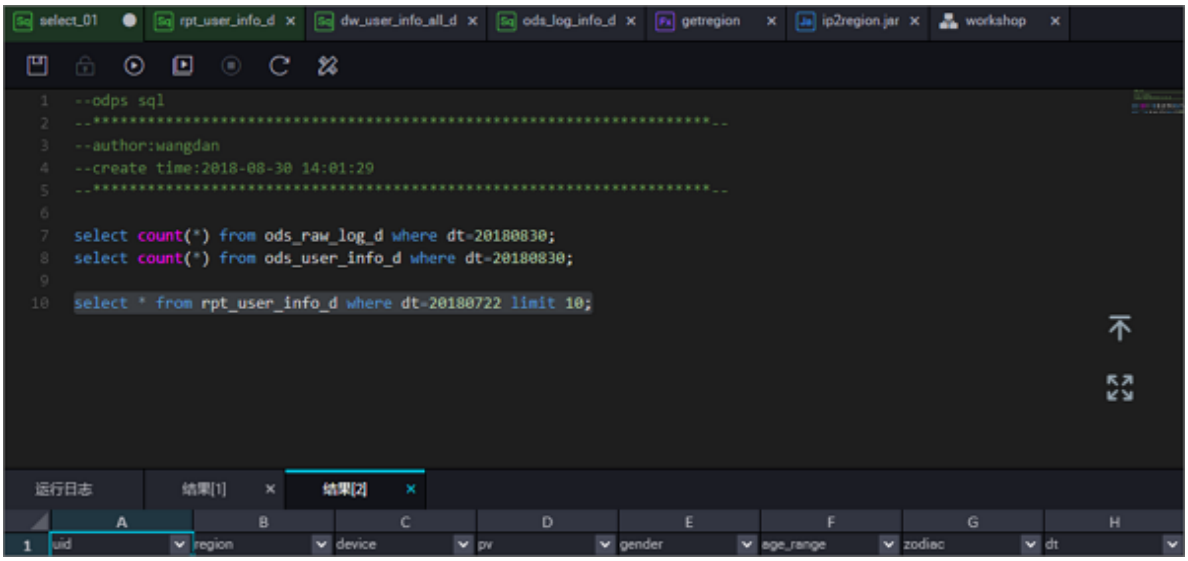
1. Click **Run** to verify the code logic.



2. Click **Queries** in the left-hand navigation bar.
3. Select **New** > **ODPS SQL**.



4. Write and execute SQL statements, Query Task for results, and confirm data output.



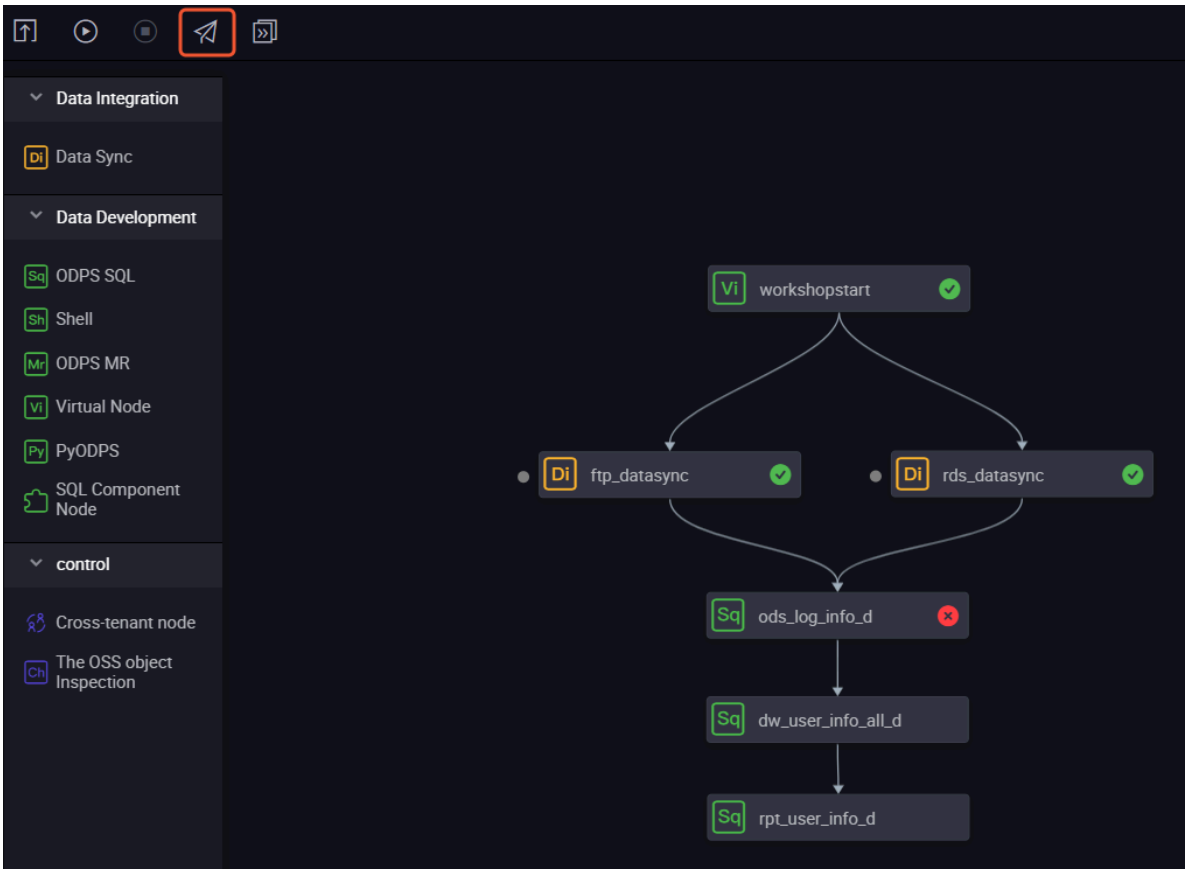
The query statement is as follows:

```
--- View the data in the data box
Select * From glaswhere DT ''business day'' limit 10;
```

Publishing Business Flow

After the Business Flow is submitted, it indicates that the task has entered the development environment, but the task of developing an environment does not automatically schedule, so the tasks completed by the configuration need to be published to the production environment (before publishing to the production environment, test this task code).

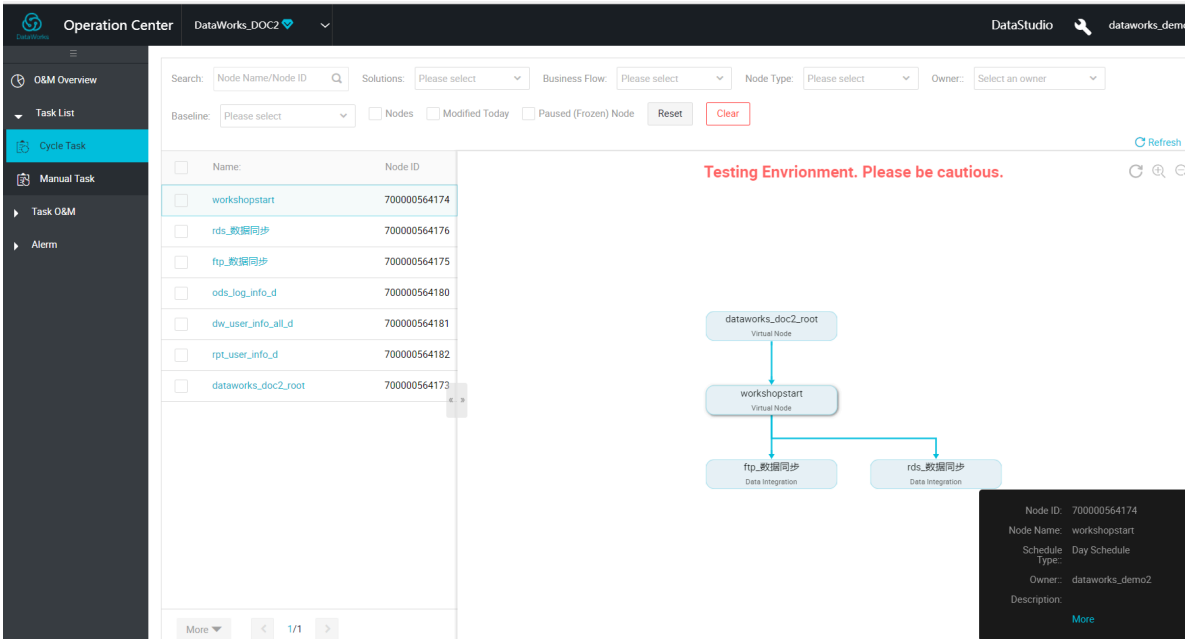
1. Click **Publish** To Go To The publish page.



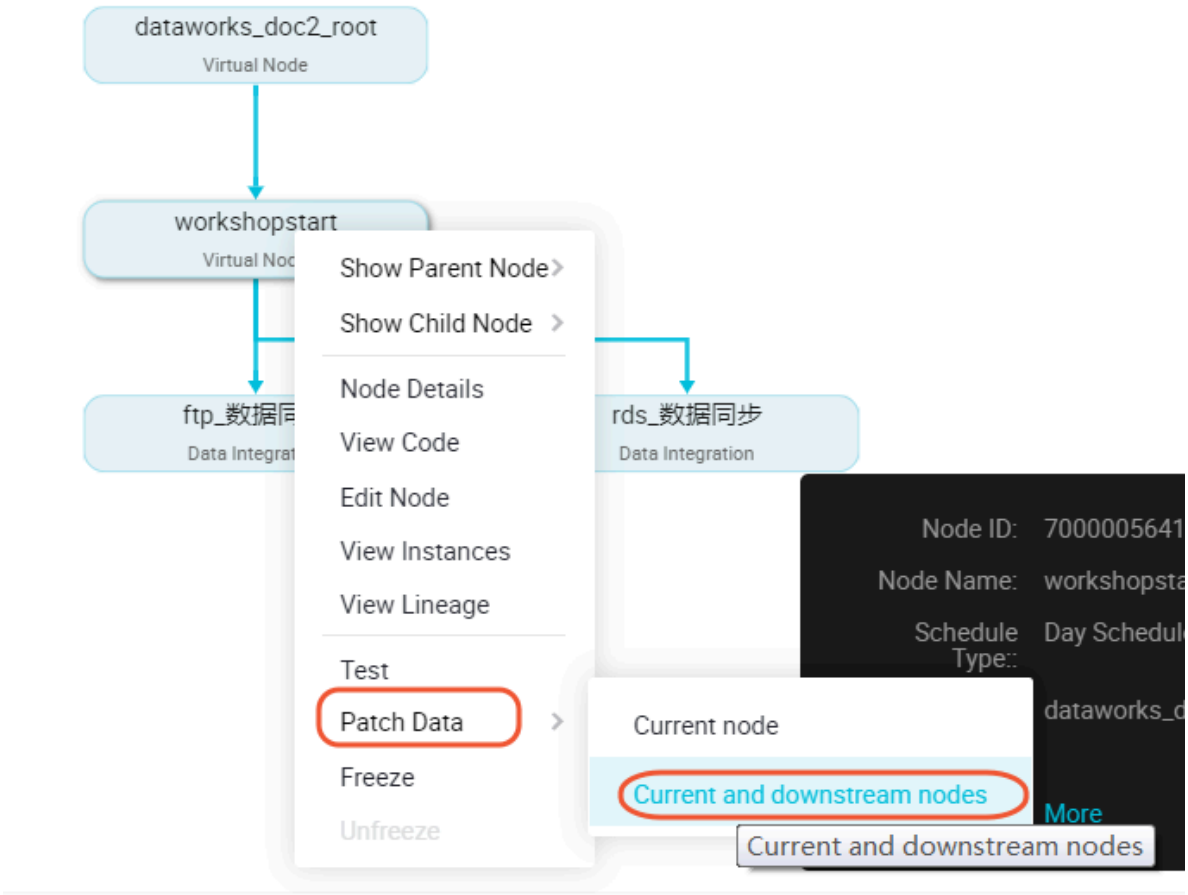
2. Select the task to publish and click **Add To Be-Published List**.
3. Enter the list of pending releases, and click **Pack and publish all**.
4. View published content on the **Publish Package List** page.

Run tasks in production

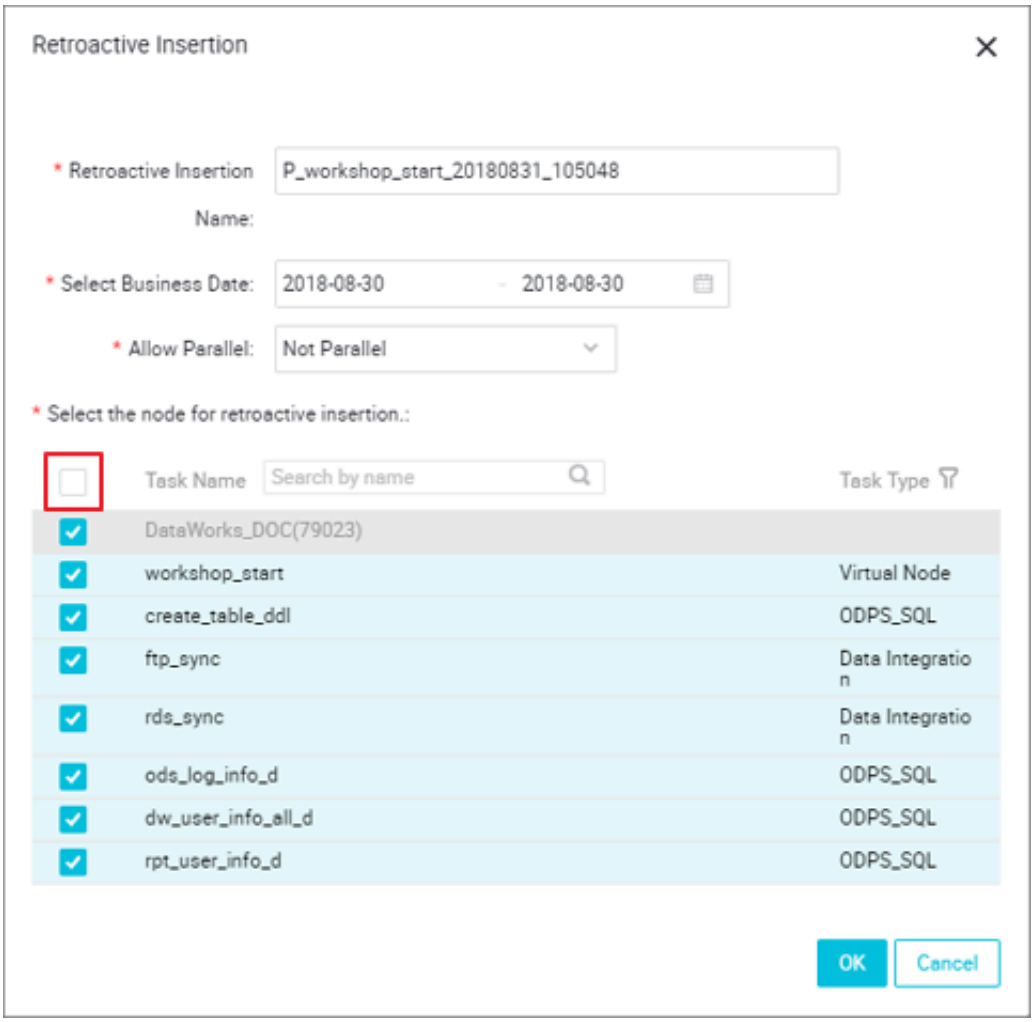
1. After the task has been published successfully, click **Operation center**.
2. Select Workshop Business Flows in the **Task List**.



3. Right-click the workshop_start node in the DAG graph and select **Patch Data** > **Current and downstream nodes**.

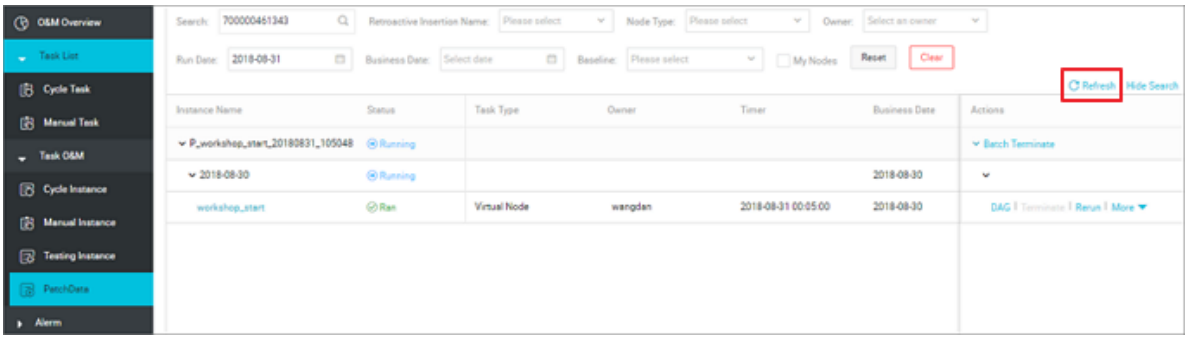


4. Check the task that needs to fill the data, enter the business date, and click **OK**.



When you click **OK**, you automatically jump to the patch data task instance page.

5. Click **Refresh** until the SQL task runs successfully.



Next step

Now that you 've learned how to create SQL tasks, how to handle raw log data, you can continue with the next tutorial. In this tutorial, you will learn how to set up data quality monitoring for tasks completed by your development, ensures the quality of tasks running. For more information, see*Data quality monitoring* .

2.4 Data quality monitoring

The paper mainly discusses how to monitor the data quality in the process of using the data workshop, set up quality monitoring rules, monitor alerts and tables.

Prerequisites

Please complete the experiment*Data acquisition: log data upload* and *Data processing: user portraits*before proceeding with this experiment.

Data quality

Data quality (DQC), is a one-stop platform that supports quality verification, notification, and management services for a wide range of heterogeneous data sources. Currently, Data Quality supports monitoring of MaxCompute data tables and DataHub real-time data streams. When the offline MaxCompute data changes, the Data Quality verifies the data, and blocks the production links to avoid spread of data pollution. Furthermore, Data Quality provides verification of historical results. Thus, you can analyze and quantify data quality. In the streaming data scenario, Data Quality can monitor the disconnections based on the DataHub data tunnel. Data Quality also provides orange and red alarm levels, and supports alarm frequency settings to minimize redundant alarms.

The process of using data quality is to configure monitoring rules for existing tables. After you configure a rule, you can run a trial to verify the rule. When the trial is successful, you can associate this rule with the scheduling task. Once the association is successful, every time the scheduling task code is run, the data quality validation rules are triggered to improve task accuracy. Once the subscription is successful, the data quality of this table will be notified by mail or alarm whenever there is a problem.

Note:

The data quality will result in additional costs.

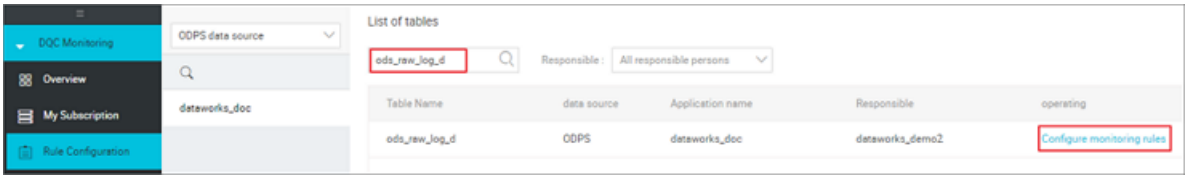
Add Table Rule Configuration

If you have completed the log data upload and user portrait experiments, you will have the following table: ods_raw_log_d, ods_user_info_d, ods_log_info_d, dw_user_info_all_d, rpt_user_info_d.

The most important thing in data quality is the configuration of table rules, so how to configure table rules is reasonable? Let's take a look at how the tables above be configured with table rules.

- ods_raw_log_d

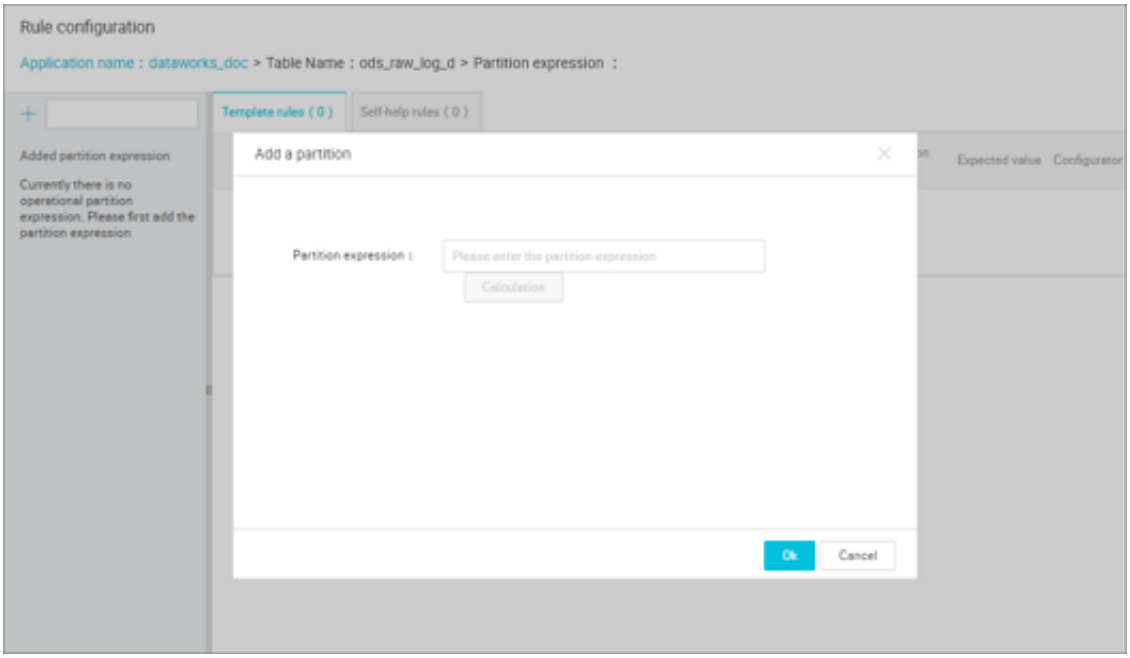
  You can see all the table information under the item in the *data quality*, now you are going to configure the data quality monitoring rules for the ods_raw_log_d data sheet.



Select the ods_raw_log_d table and click **Rule Configuration** to go to the following page.

You can review the data sources for this ods_raw_log_d table. The data for ods_raw_log_d table is from ftp. Its partition is ${bdp.system.bizdate} format and is written into the table ("dbp. system.bizdate" is the date to get to the day before).

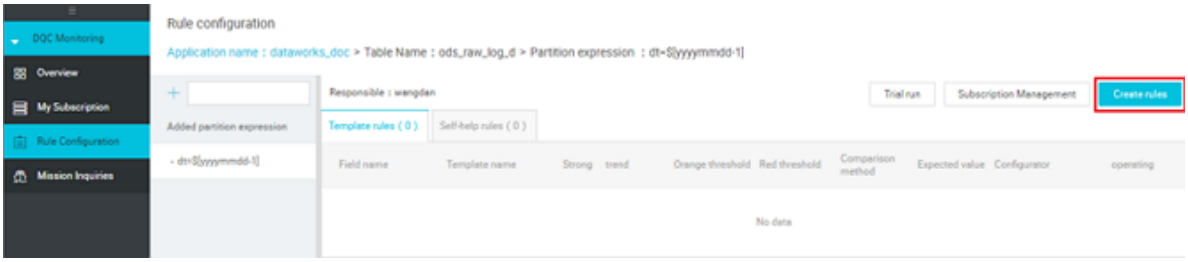
For this type of daily log data, you can configure the partition expression for the table. There are several kinds of partition expressions, and you can select dt = $ [yyyymmdd-1]. Refer to the documentation*Parameter configuration* for detailed interpretation of scheduling expressions.



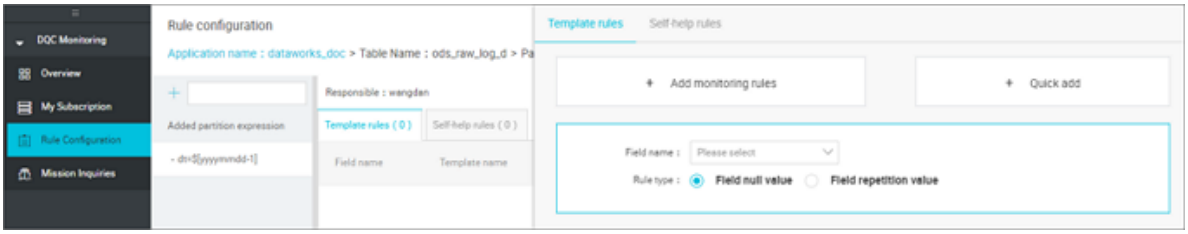
Note:

If there is no partition columns in the table, you can configure it as no partition. Depending on the real partition value, you can configure the corresponding partition expression.

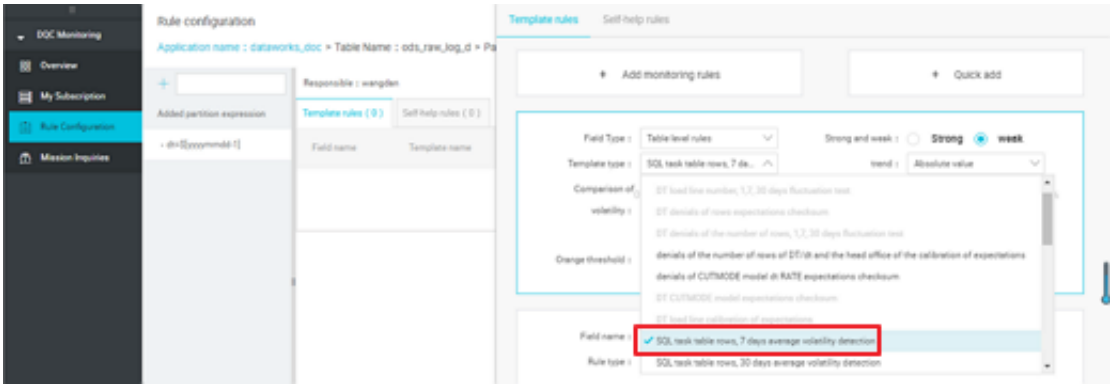
After confirm, you can see the interface below and choose to **create rules**.



When you select to create a rule, the following interface appears.



Click **Add monitoring rule** and a prompt window appears for you to configure the rule.



The data in this table comes from the log file that is uploaded by FTP as the source table. You need to determine whether there is data in this table partition as soon as possible. If there is no data in this table, you need to stop the subsequent tasks from running as if the source table does not have data, the subsequent task runs without meaning.

Note:

Only under strong rules does the red alarm cause the task to block, setting the instance state to failure.

When configuring rules, you need to select the template type as the number of table rows, sets the strength of the rule to strong. Click the **Save** button after the settings are completed.

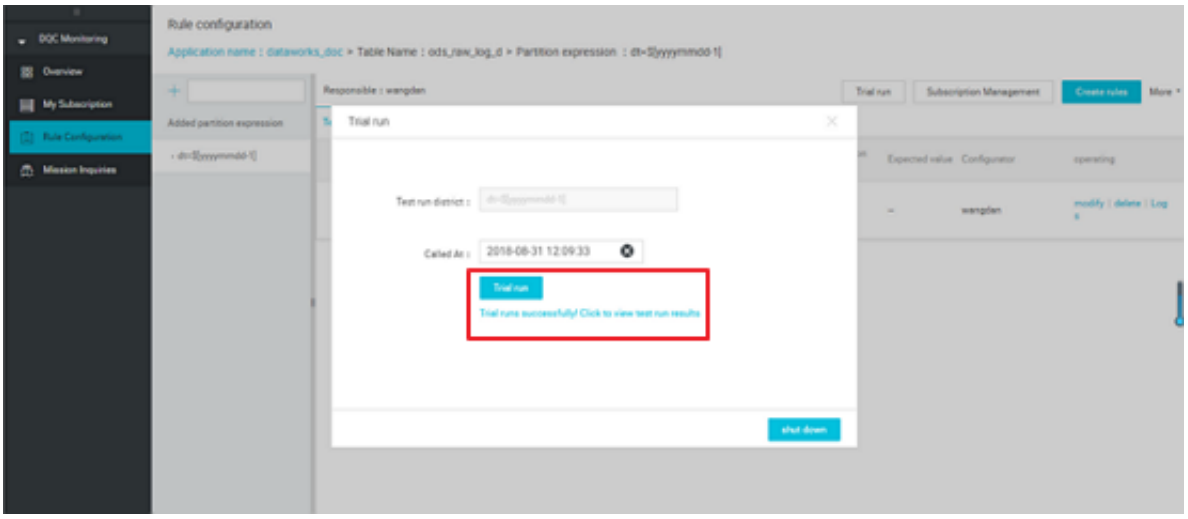
Note:

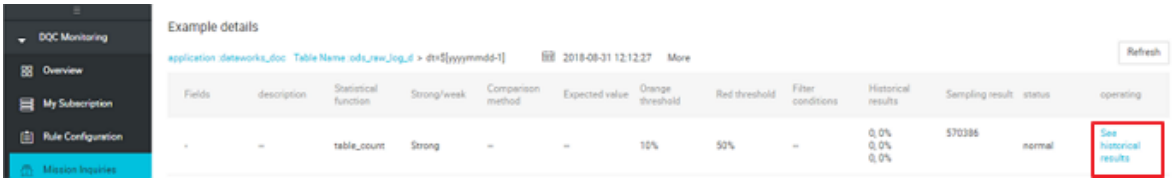
This configuration is primarily to avoid the situation that there is no data in the partition, which causes the data source for the downstream task to be empty.

Rules test

In the upper-right corner, there is a node test button that can be used to verify configured rules . The test button can immediately trigger the validation rules for data quality.

When you click the test button, you are prompted for a window to confirm the test date. After a run is clicked, there will be a prompt information below telling you to jump to the test results by clicking prompt information.





Note:

According to the test results, the data of the Mission output can be confirmed to be in line with the expectations. It is recommended that once each table rule is configured, a trial operation should be carried out to verify the applicability of the table rules.

When the rules are configured and the trial runs are successful, you need to associate the table with its output task. In this way, every time the output task of the table is run, the validation of the data quality rules is triggered to ensure the accuracy of the data.

Associated Scheduling

Data quality support being associated with scheduling tasks. After the table rules and scheduling tasks are bound, when the task instance is run, the data quality check is triggered. There are two ways to schedule table ruless:

- Perform table rule associations in operations center tasks.
- Association in the regular configuration interface for data quality.

Operations Center Association Table rules

In care center, in cycle tasks, locate the **ftp_datasync** task, and in **more**, select **configure data monitoring**.

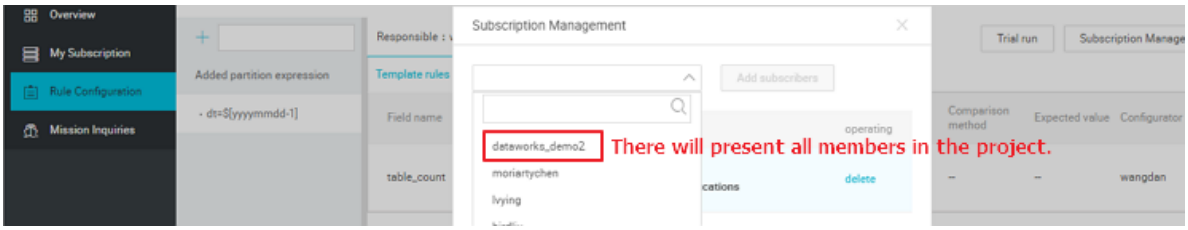
Enter the monitored table name in the burst window, as well as the partition expression. The table entered here is named as ods_user_info_d and the partition expression is dt = $ [yyyymmdd-1].

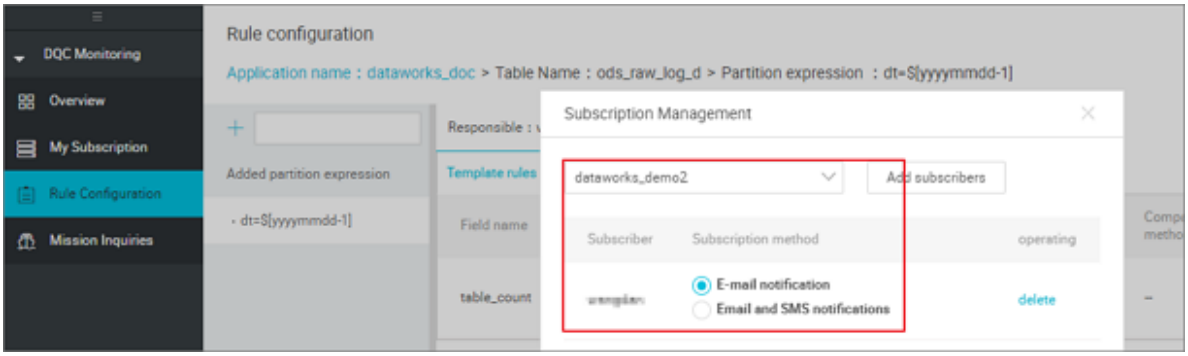
Click **Configure** to quickly go to the rule configuration interface.

Configure task subscriptions

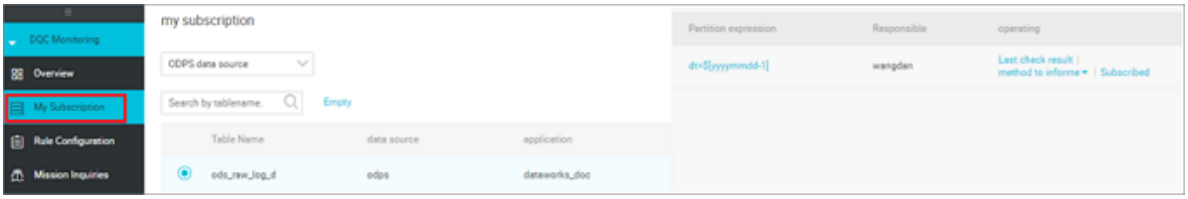
After the associated scheduling, every time the scheduling task is run, the data quality verificati on is triggered. Data quality supports setting up rule subscriptions, and you can set up subscriptions for important tables and their rules, set up your subscription to alert you based on the results of the data quality check. If the data quality check results are abnormal, notificati ons are made based on the configured alarm policy.

Click **Subscription Management** to set up subscription methods. Mail notifications, email and SMS notifications are currently supported.





After the subscription management settings are set up, you can view and modify them in **My Subscription**.



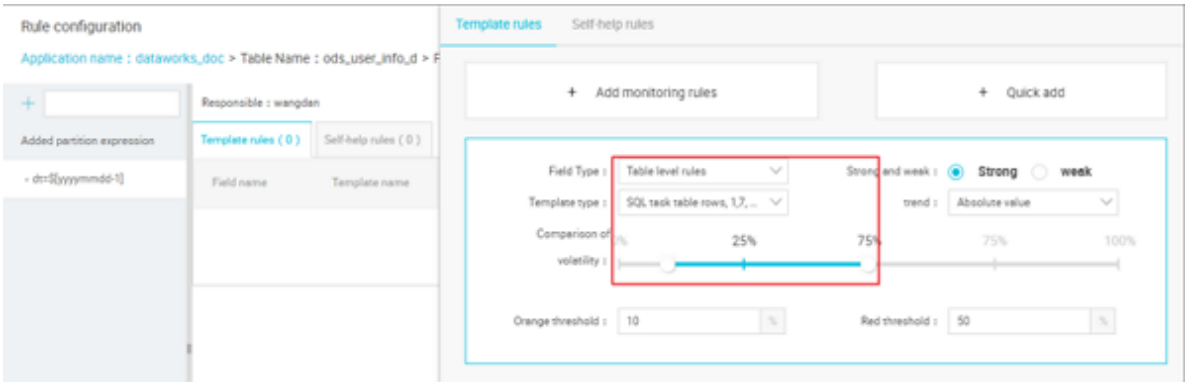
It is recommended that you subscribe to all rules so that the verification results are not notified in a timely manner.

- ods_user_info_d

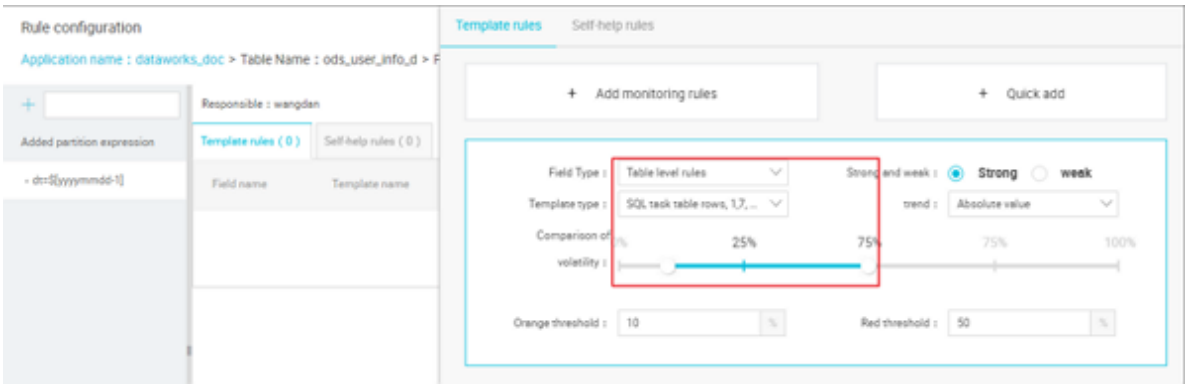
  The data in the ods_user_info_d table is from RDS database. When you configure rules, you need to configure the table to check the number of rows and the unique validation of the primary key to avoid duplication of data.

Similarly, you need to configure a monitoring rule for a partition field first, and the monitoring time expression is: dt = $[yyyymmdd-1]. After successful configuration, you can see a successful partition configuration record in the partition expression that has been added.

After the partition expression is configured, click **Create Rule** on the right to configure the validation rules for data quality. Add monitoring rules for table rows, rule intensity is set to strong, comparison mode is set to expectations greater than 0.



Add column-level rules and set primary key columns to monitor columns. The template type is: the number of repeated values in the field is verified, and the rule is set to weak, the comparison mode is set to a field where the number of duplicate values is less than 1. After the setting is completed, click the bulk **Save** button.



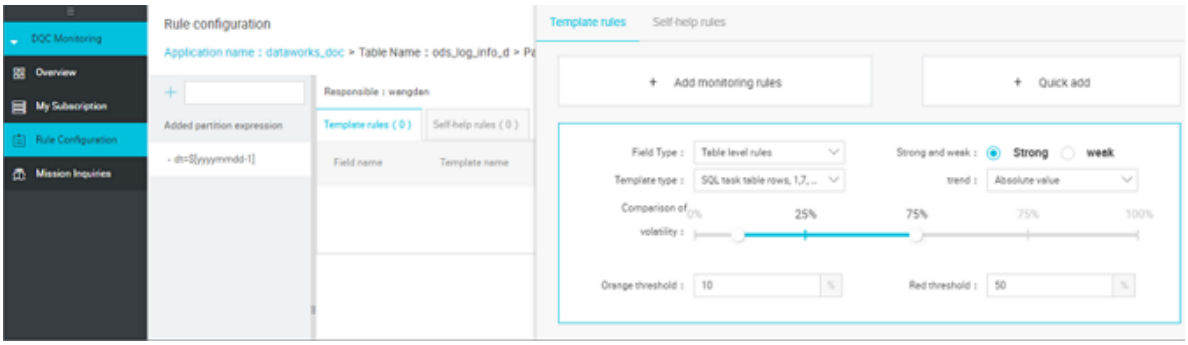
Note:

This configuration is primarily designed to avoid duplication of data which may result in contamination of downstream data.

ods_log_info_d

The data of this ods_log_info_d table mainly is the analysis of the data in the table. Because the data in the log cannot be configured for excessive monitoring, you only need to configure the validation rules that is not empty for the table data. The partition expression for the first configuration table is: dt = $[yyyymmdd-1]

The configuration table data is not an empty calibration rule, and the rule strength should be set to strong. The comparison is set to an expected value of not equal to 0, and after the setup is complete, click the **Save** button.



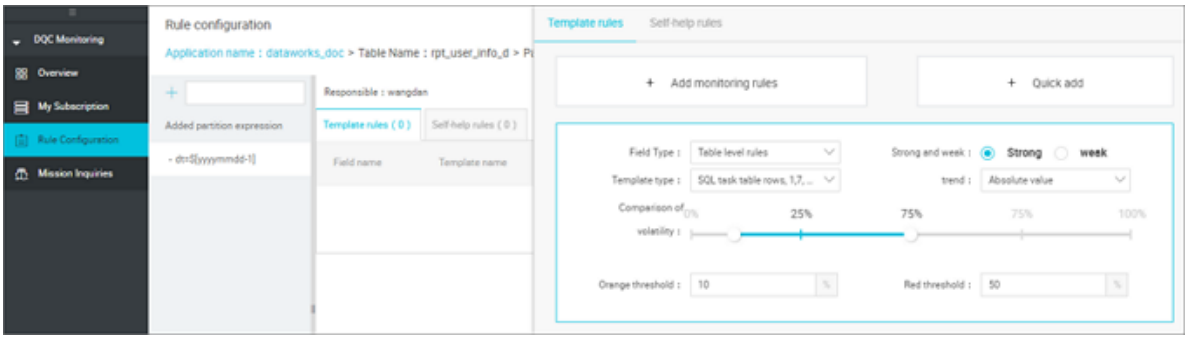
- dw_user_info_all_d

  This dw_user_info_all_d table is a summary of data for both the ods_user_info_d table and the ods_log_info_d table, because the process is relatively simple, the ODS layer is also configured with a rule that the number of table rows is not empty, so the table does not have the data quality monitoring rules configured to save on computing resources.

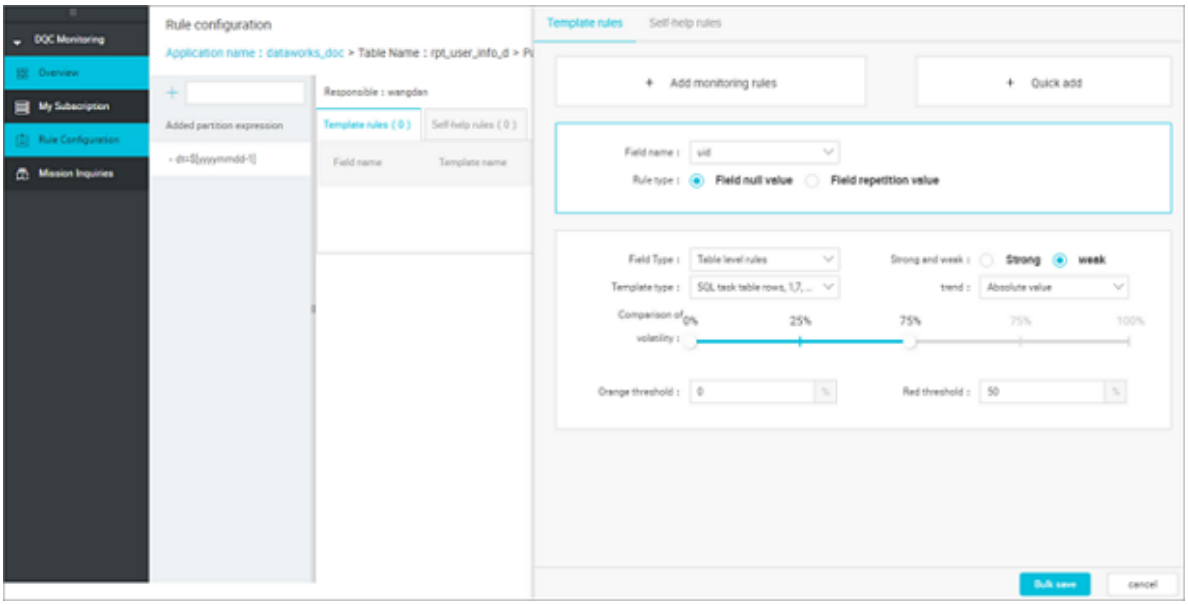
- rpt_user_info_d

  The rpt_user_info_d table is the result table after the data aggregation. Based on the data in this table, you can monitor the number of table rows for fluctuations, and verify the unique values for primary keys. Partition expression for the first configuration table: dt = $[yyyymmdd-1]

Then you may configure the monitoring rules: Click **Create rule** on the right, and click **Add Monitoring Rules** to monitor columns. The number of repeated values in the field is verified, and the rule is set to weak. The comparison mode is set to field repeat values less than 1.



Continue to add monitoring and table rules.



As you may notice, the lower are the tables in the data warehouse, the more times the strong rules are set. That's because the data in the ODS layer is used as the raw data in the warehouse and you need to ensure the accuracy of its data, avoiding poor data quality in the ODS layer, and stop it in time.

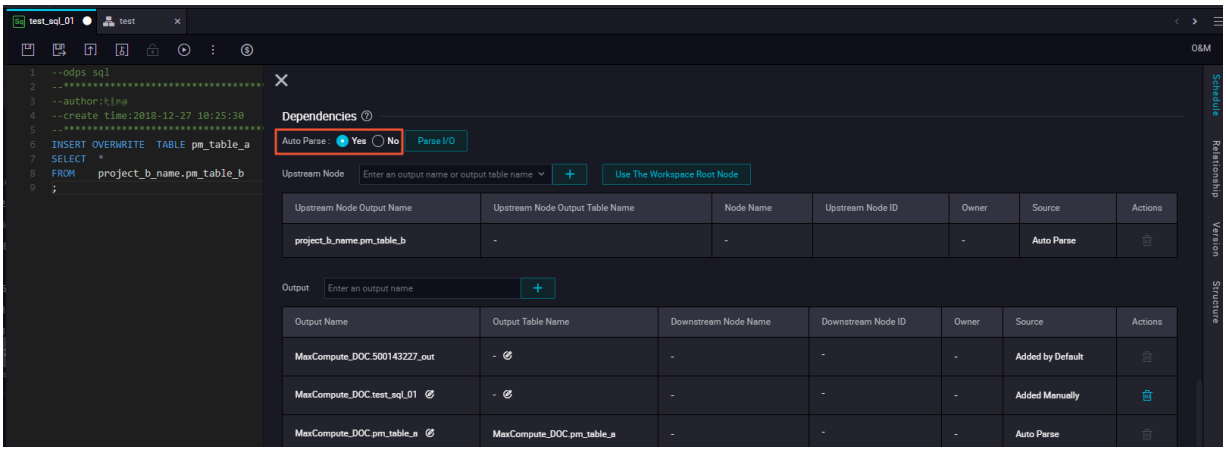
Data quality also provides an interface for task queries on which you can view the validation results for configured rules.

3 Best practices for setting scheduling dependencies

In DataWorks V2.0, when configuring scheduling dependencies, dependencies between tasks need to be set according to the output name of the current node as an associated item. This article details how to configure the input and output of task scheduling dependencies.

How to configure the node input of a task

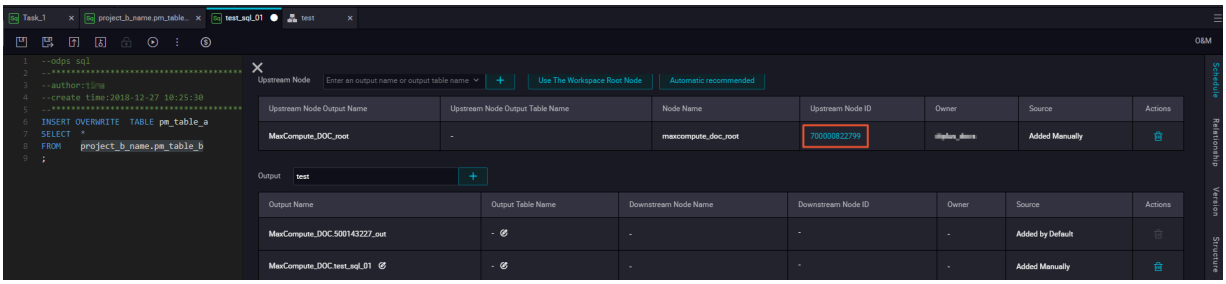
There are two ways to configure the node input: one is to use the automatic code parsing function to resolve the dependency of the task, the other is to manually enter the task dependency (manually entering the **Upstream Node Output Name**).



Note:

When manually entering an upstream node, the input is **Output Name** of the parent node. If the parent node task name does not match the parent node's output name, be sure to enter the node output name correctly.

When configuring an upstream node, you may encounter problems with the upstream node parsed automatically is an invalid upstream dependency. A method of identifying whether dependencies are valid: view the parsed upstream dependencies and check if the value is displayed in the **Upstream Node ID** column, as shown in the following figure.

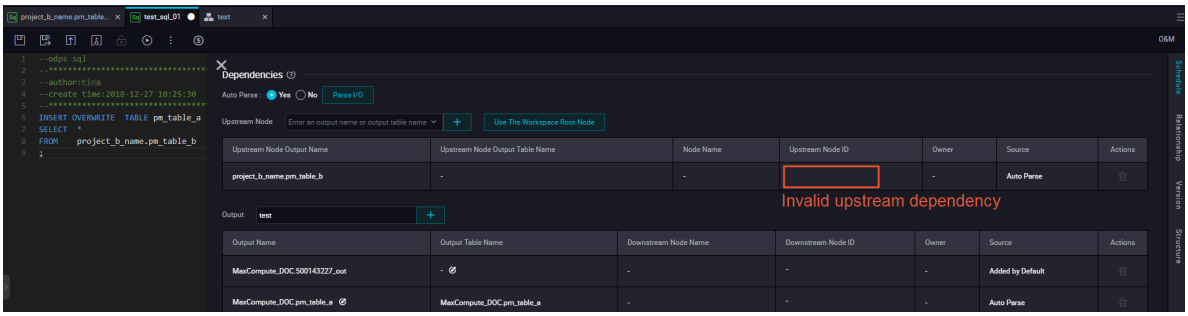


The configuration of task dependencies is essentially to set the dependencies between two nodes. Only the nodes that exist will be able to set up valid dependencies, task dependencies can be set successfully.

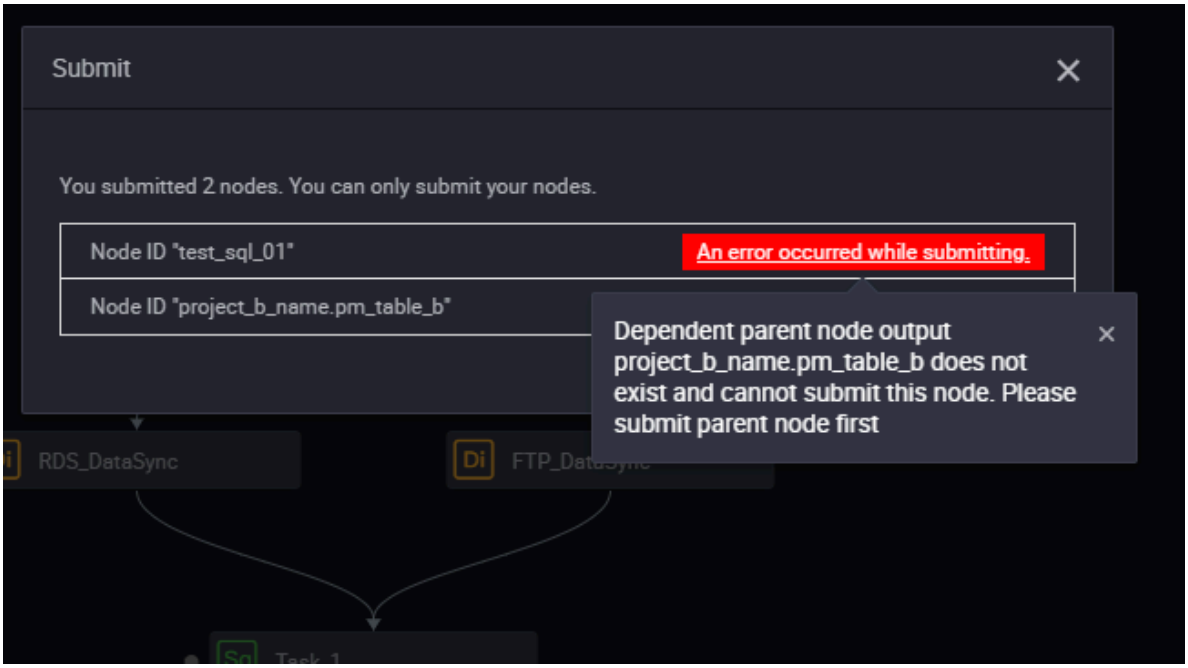
Invalid upstream dependency

Invalid upstream dependencies are usually in two cases.

1. The parent node does not exist.



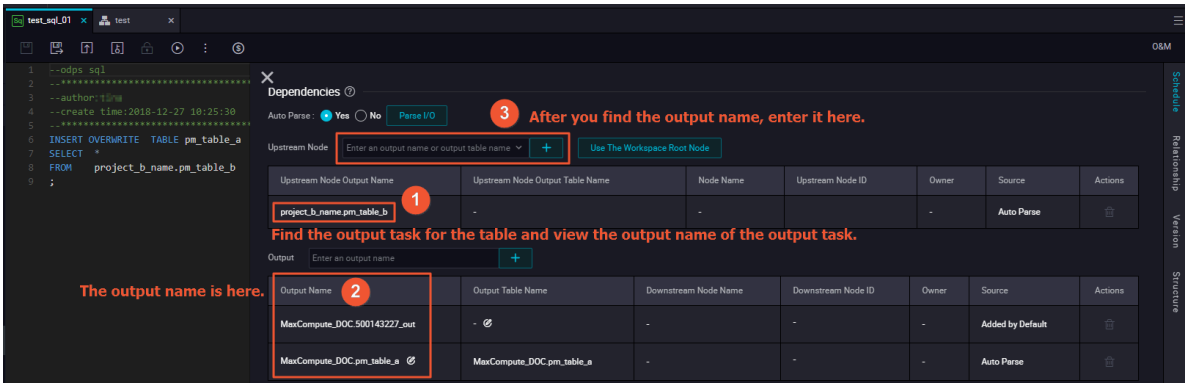
2. The parent node output does not exist.



Invalid upstream dependencies typically occur because the parsed parent node output name does not exist. In this case, it may be due to the fact that the table "project_b_name.pm_table_b" does

not output task, or the node output is configured incorrectly for the table output task and can't be parsed. There are two solutions:

1. Confirm that the table has an output task.
2. Confirm what the output name of this table's output task is, and manually enter the node output name into the dependent upstream node.



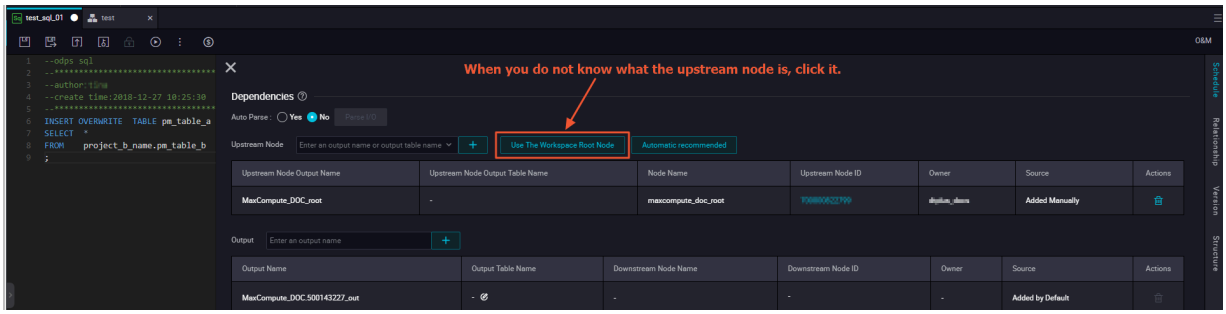
Note:

When you enter an upstream node manually, you enter the parent node's output name. If the parent node task name does not match the parent node's output name, be sure to enter the node output name correctly.

For example, the output name of the upstream node A is A1, and downstream node B depends on node A. At this point, enter A1 in the input box of the upstream node, and click the plus sign on the right to add it.

How to configure upstream dependencies

If your table is extracted from the source library and there is no upstream, you can click **Use The Workspace Root Node** to obtain upstream dependencies.



How to configure the node output of a task

The simplest way to efficiently configure the node output is: the node name, the node output name and the node output table name share the same name and three in one. The advantages are as follows.

1. You can quickly know which table this task is operating on.
2. It is possible to quickly know how far this task will impact if it fails.
3. When you use auto parsing to configure task dependencies, as long as the node output is consistent with the three-in-one rule, the precision performance of automatic parsing is greatly improved.

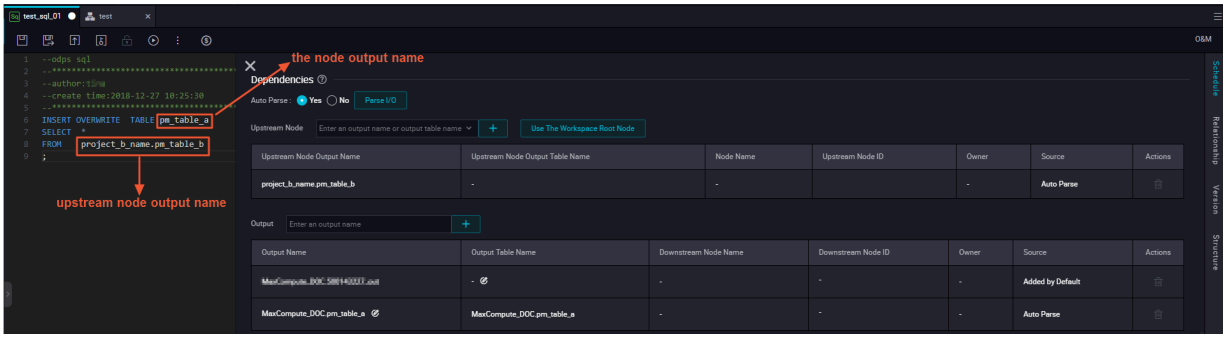
Automatic parsing

Automatic parsing: refers to automatically parse scheduling dependencies by the code. Implementation principle: only table names can be obtained in the code, and the automatic parsing function can parse the corresponding output task according to the table name.

For example, the type node code is shown below.

```
INSERT OVERWRITE TABLE pm_table_a SELECT * FROM project_b_name.
pm_table_b;
```

The dependencies parsed are as follows.



DataWorks can automatically parse the node which this node needs to be dependent on `project_b_name` to output `pm_table_b`, and the final output of the node `pm_table_a` . Therefore, the resolution is that the parent node output name is `project_b_name.pm_table_b`, and the node output name is `project_name.pm_table_a`(The project name is MaxCompute_DOC).

- If you do not want to use dependencies that are parsed from the code, select **No**.
- If there are many tables in the code that are temporary tables: For example, the table beginning with **t_** is a temporary table. Then the table is not parsed as the schedule dependency. The

definition of temporary tables is that you can define which form the table begins with is a temporary table by project configuration.

- If a table in the code is both the output table and referenced table (depended table), it is parsed only as the output table.
- If a table in the code is referenced or output for multiple times, only one scheduling dependency is parsed.

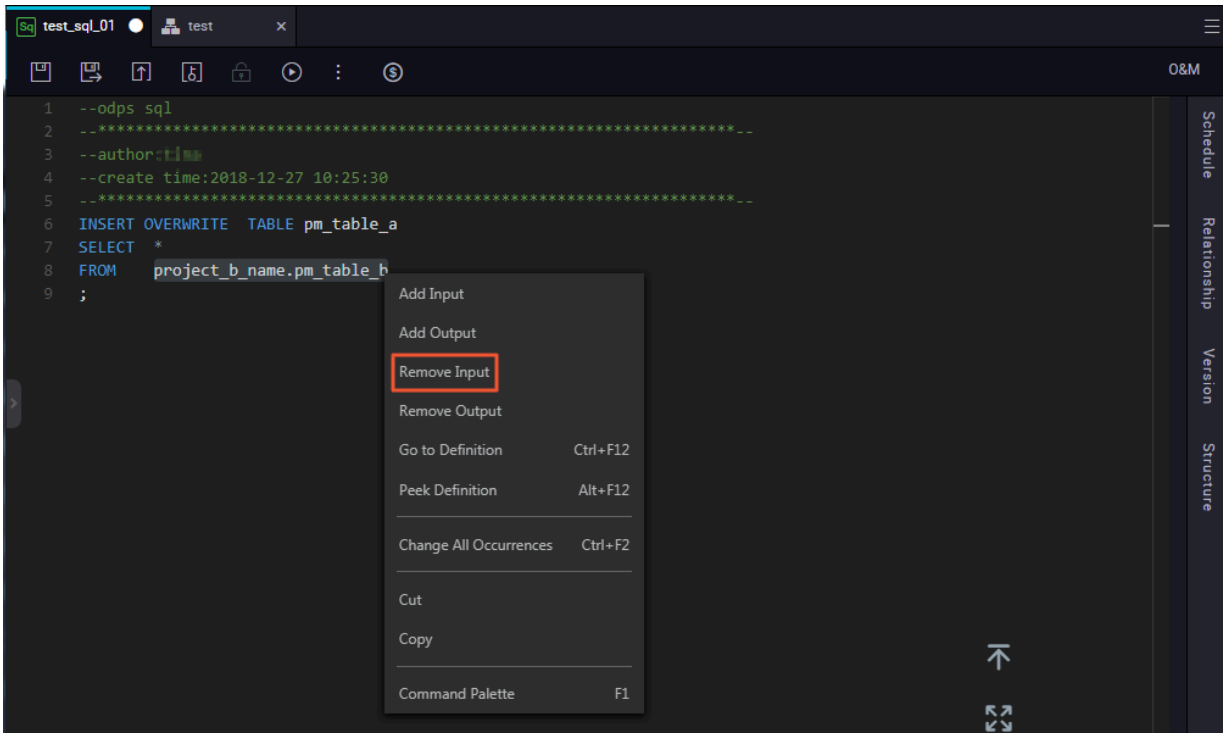
Note:

By default, a table with a name starting with **t_** is recognized as a temporary table. Auto parsing does not resolve the temporary table. If the table with a name starting with **t_** is not a temporary table, contact your project administrator to modify it in the project configuration.

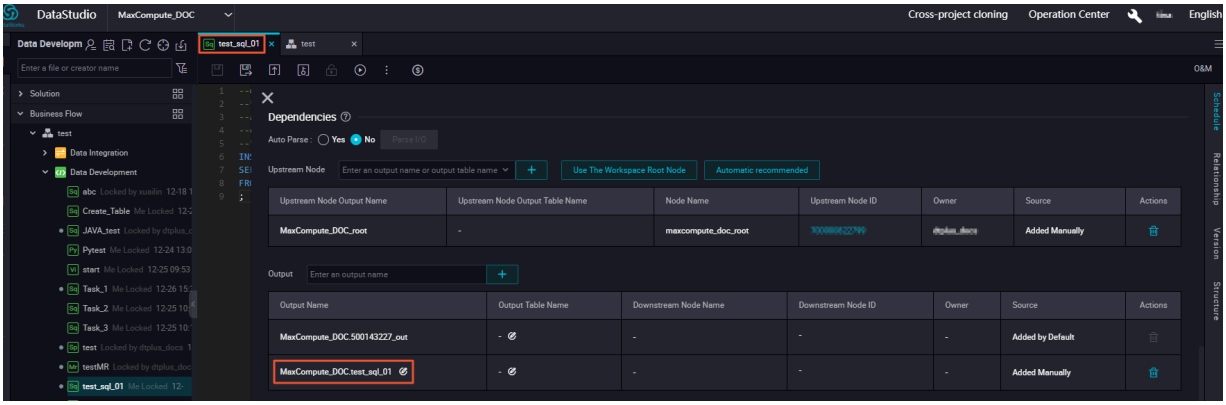
How to delete the input and output of a table

When you're in the process of data development, you often use static tables (data is uploaded to a table from a local file), this static data does not actually output task. At this time, when configuring dependencies, you need to delete the input of the static table: if the static table does not satisfy the form of **t_**, it will not be processed as a temporary table, in which case you need to delete the input of the static table.

You select the table name in the code, click **Remove Input**.



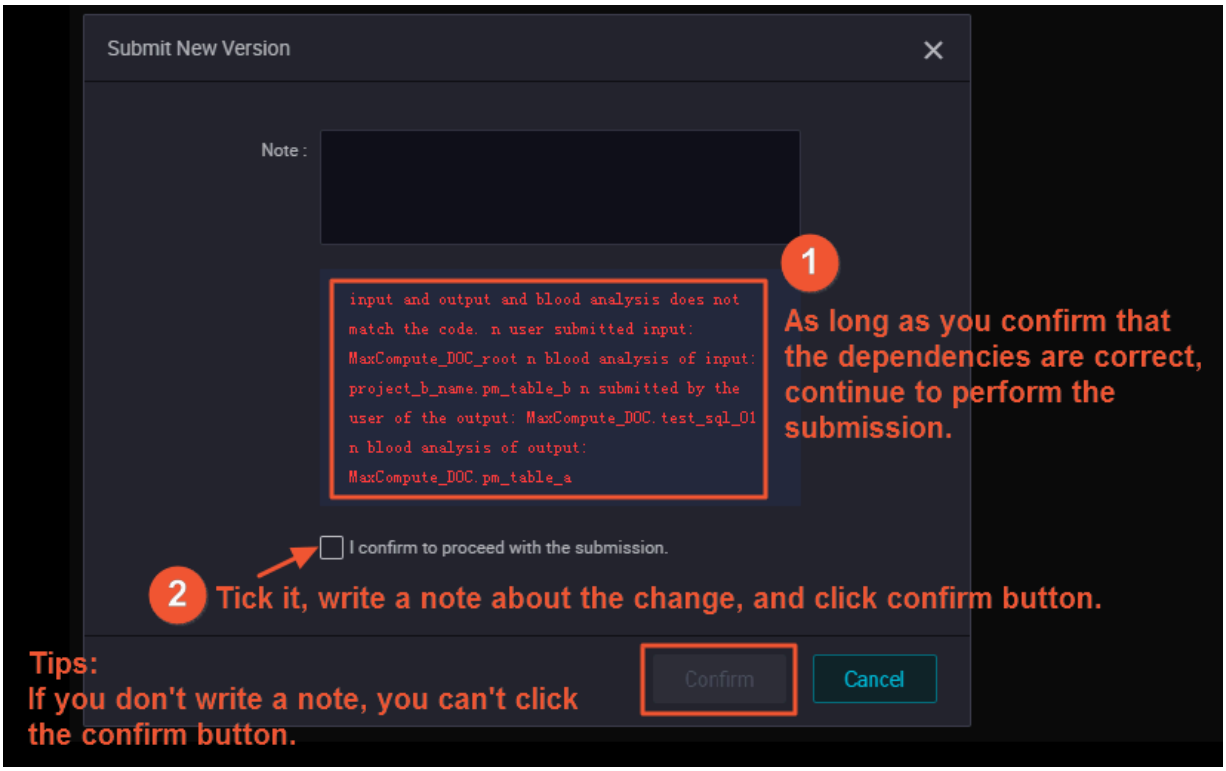
If you are upgrading from DataWorks to DataWorks V2.0, we set the node output for the migrated DataWorks task to `ProjectName.NodeName` for you by default.



Attentions

When the task dependency configuration is complete, the submitted window shows an option: whether confirm to proceed with the submission when the input and output does not match the code blood analysis.

The premise of this option is that you have confirmed that the dependencies are correct. If you cannot confirm, you can confirm the dependencies as described above.


Submit New Version
Note :
input and output and blood analysis does not match the code. n user submitted input: MaxCompute_DOC_root n blood analysis of input: project_b_name.pm_table_b n submitted by the user of the output: MaxCompute_DOC.test_sql_01 n blood analysis of output: MaxCompute_DOC.pm_table_a
1
As long as you confirm that the dependencies are correct, continue to perform the submission.
I confirm to proceed with the submission.
2
Tick it, write a note about the change, and click confirm button.
Tips:
If you don't write a note, you can't click the confirm button.
Confirm
Cancel