# Alibaba Cloud DataWorks

## Best Practices

Issue: 20190228

# 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 intellectual 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:<br>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:<br>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:<br>Take the necessary precautions to save exported data containing sensitive information. |
| | This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user. |  Note:<br>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]* |

| Style | Description | Example |
| --- | --- | --- |
| {} or {a\|b} | It indicates that it is a required value, and only one item can be selected. | `swich {stand \| slave}` |

# Contents

# 1 Workshop

## 1.1 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 architecture, 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.

## 1.2 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)*.

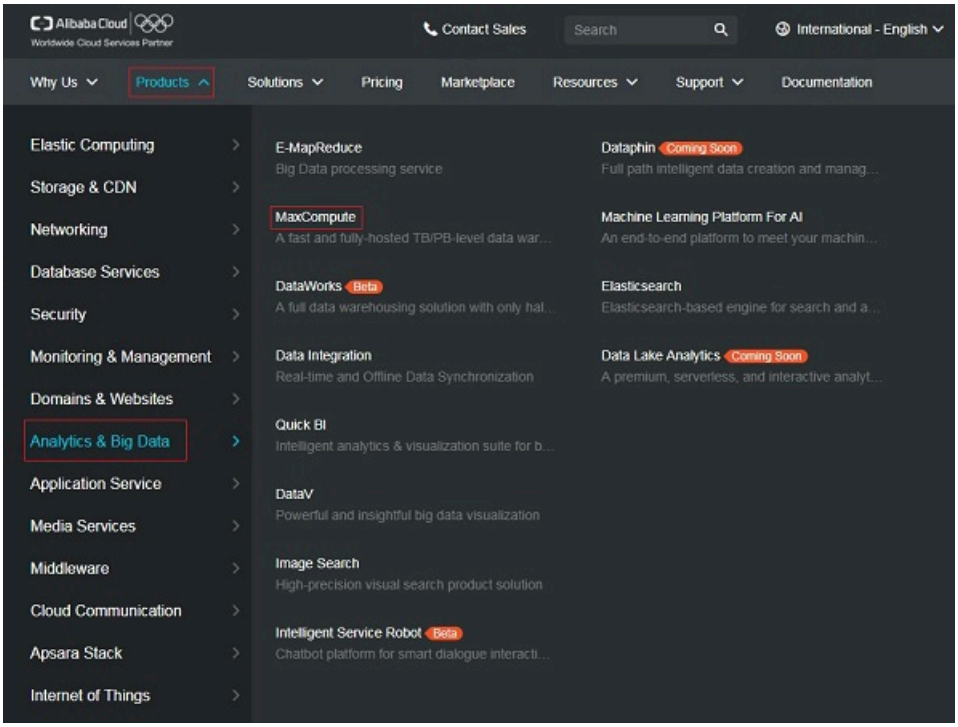
### Prerequisites

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 > MaxCompututе and go to the MaxCompute product details page.



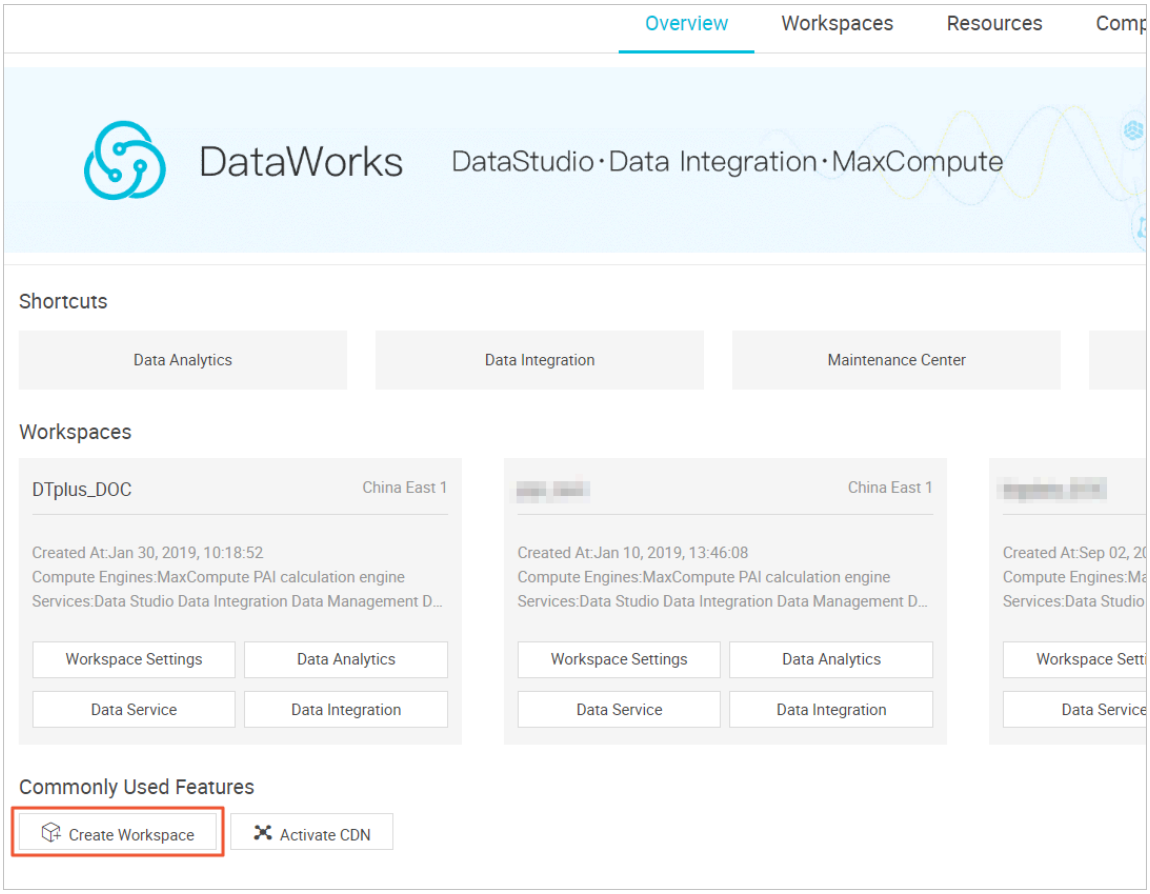

3. Click Start now.
4. Select Pay-As-You-Go, click Buy Now.

## Create workspace

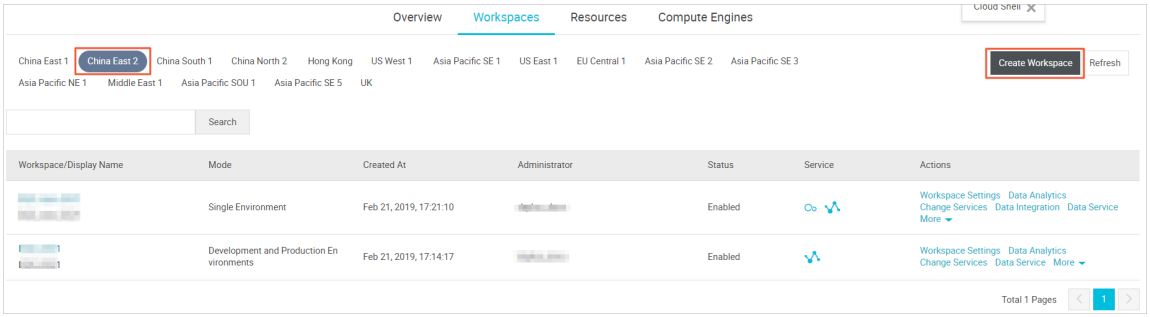
1. Log on to the *DataWorks console* by using a primary account.

2. You can create a workspace in two ways.

   - On the console Overview page, go to Commonly Used Features > Create Workspace.




   - On the console Workspace page, select region, and then click Create Workspace in the upper right corner.

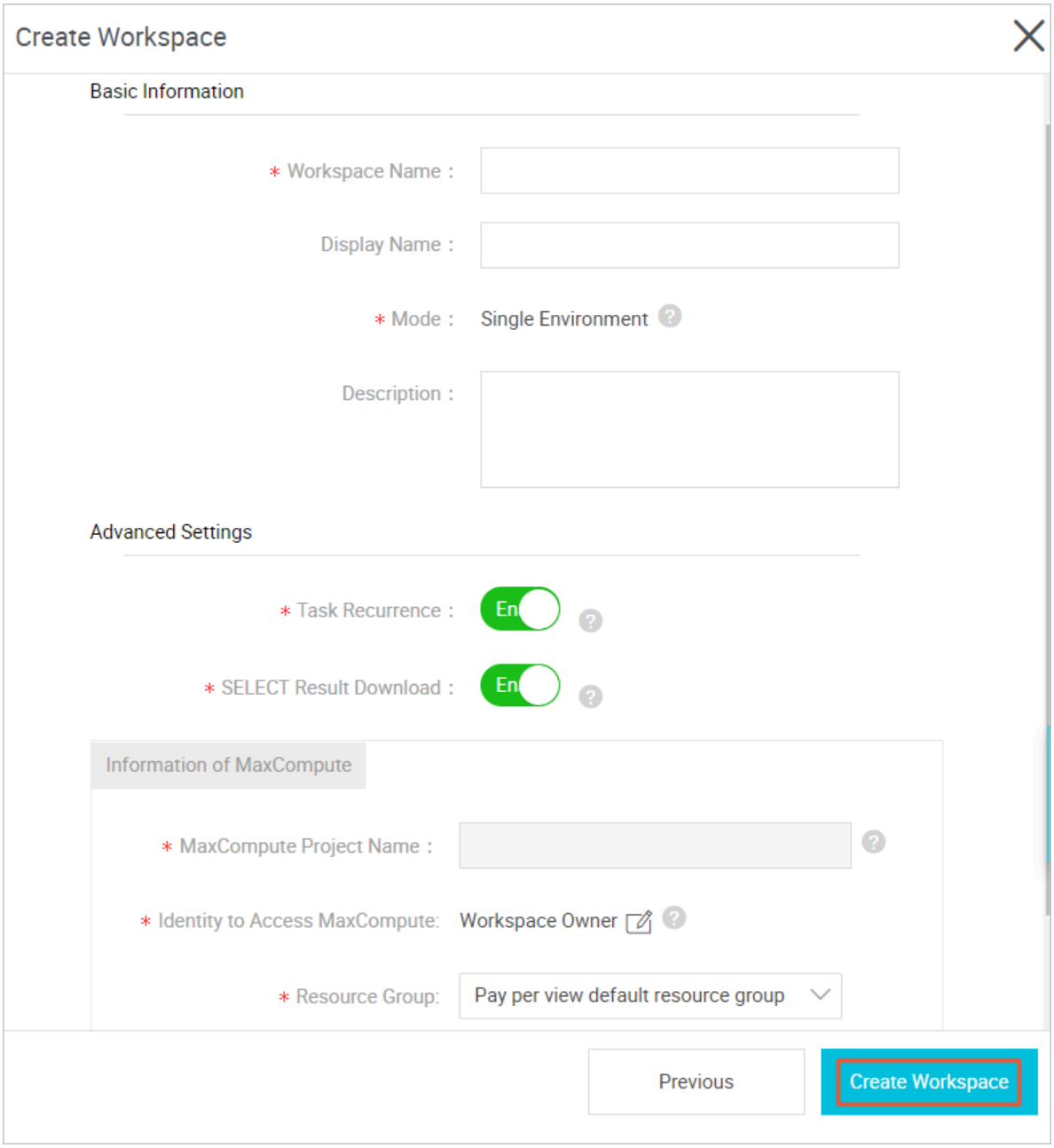



3. Fill in the configuration items in the Create Workspace dialog box. Select a region and a calculation engine service.

**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 analytics, O&M, and administration are selected by default.

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

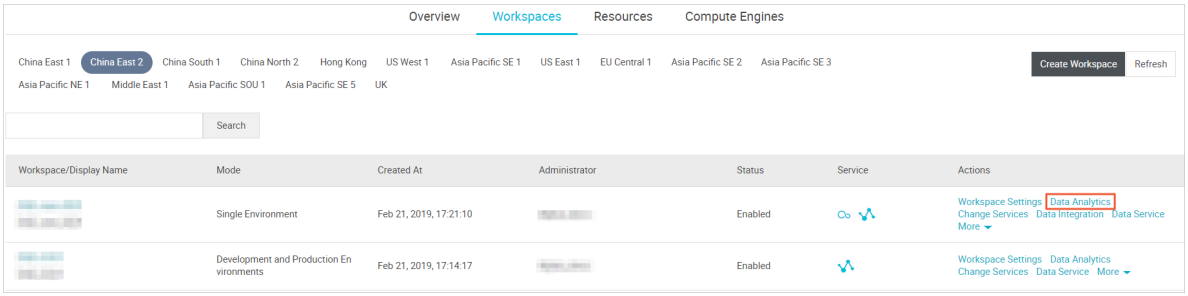



**Note:**

- The workspace name needs to begin with a letter or underline, and can only contain letters, underscores, and numbers.

- The workspace 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 workspace has been created successfully, you can select the Workspace page to Data Analytics after viewing the workspace space.



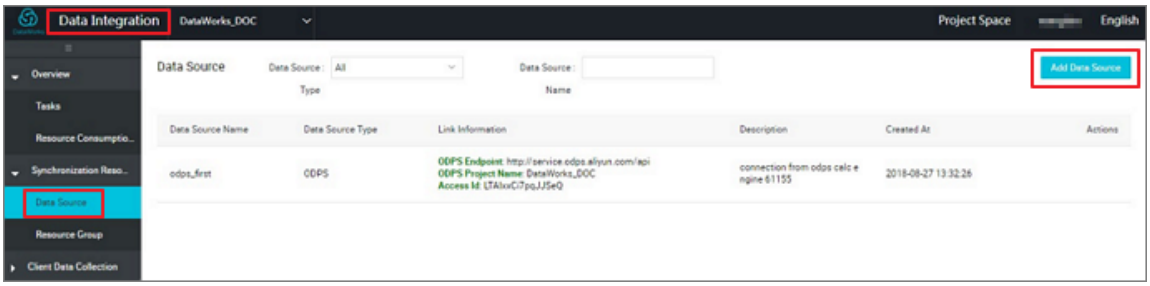

## Create data source

**Note:**

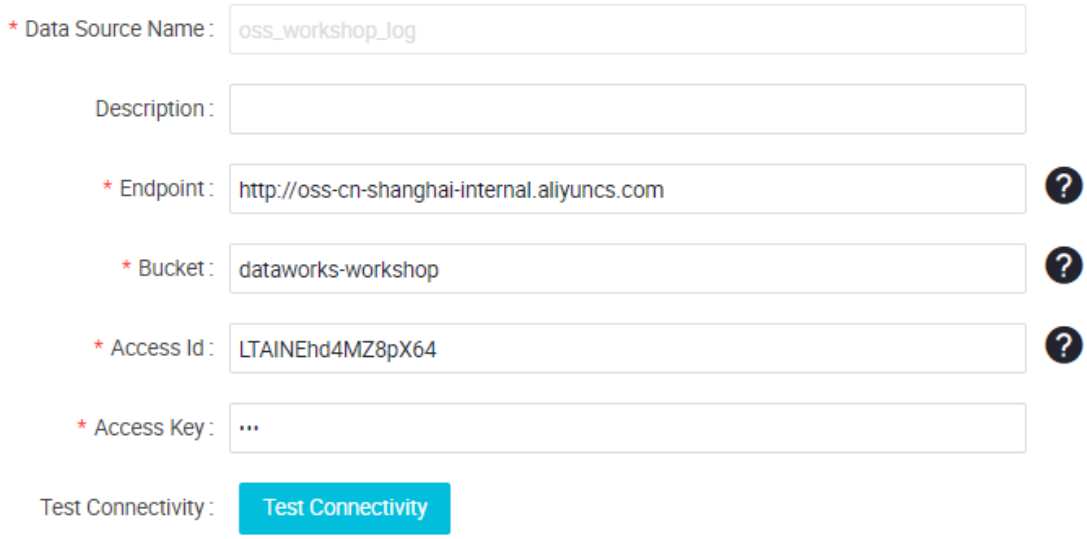
Based on the scenario simulated by this lab, you need to distribute to create both the OSS data source and the RDS data source.

- 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:

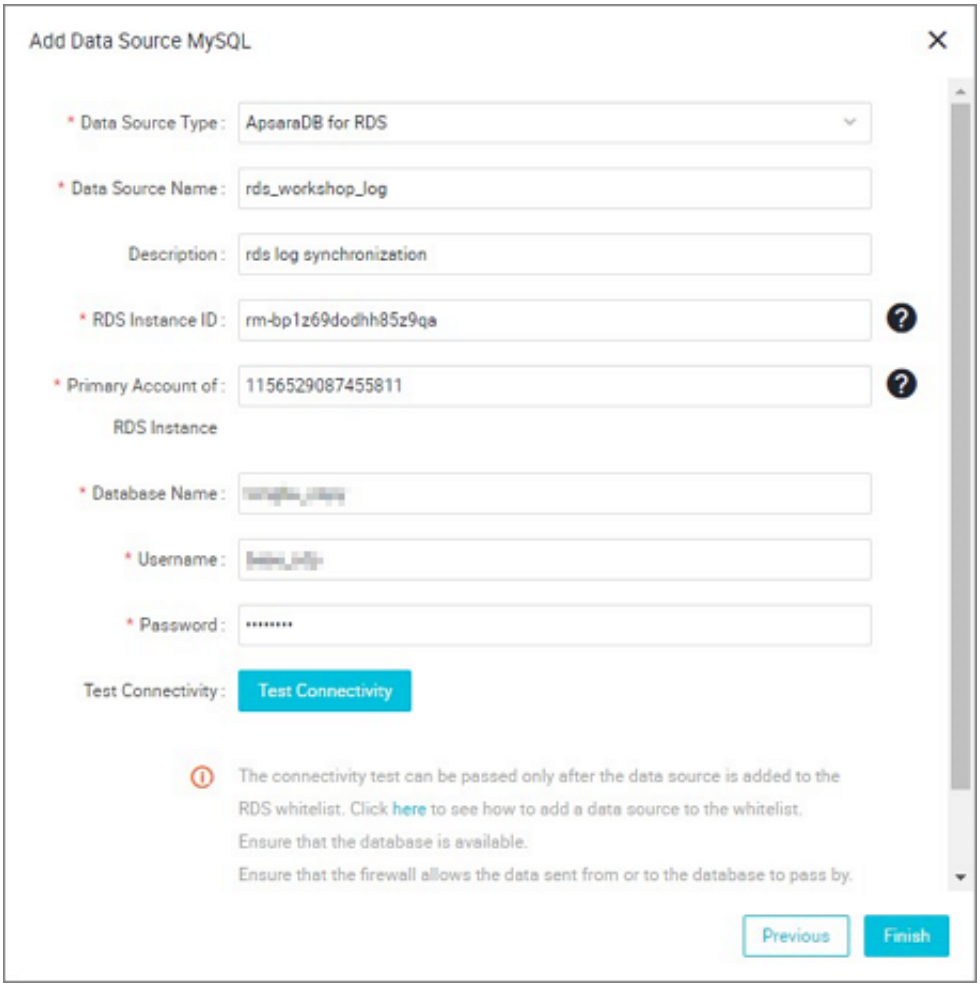
- Endpoint: http://oss-cn-shanghai-internal.aliyuncs.com
- bucket: dataworks-workshop
- AK ID: LTAINEhd4MZ8pX64
- AK Key: lXnzUngTSebt3SfLYxZxoSjGAK6IaF

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 SourcePage, and click Add Data Source.
2. Select the data source type as MySQL, and fill in the configuration information.



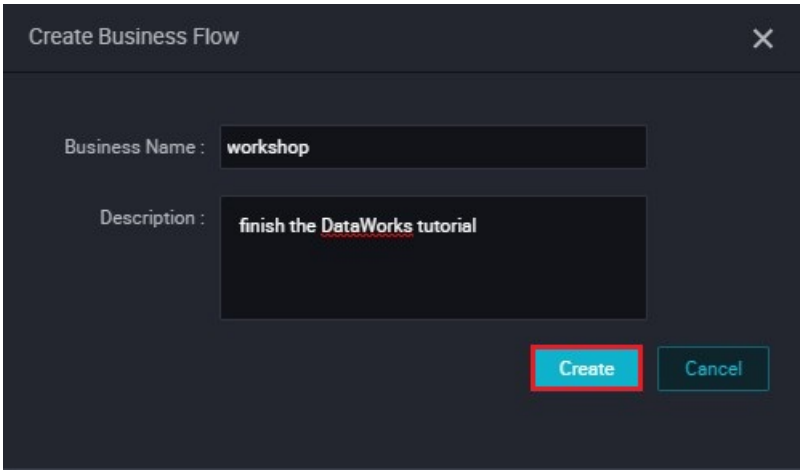

Parameters:

- 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

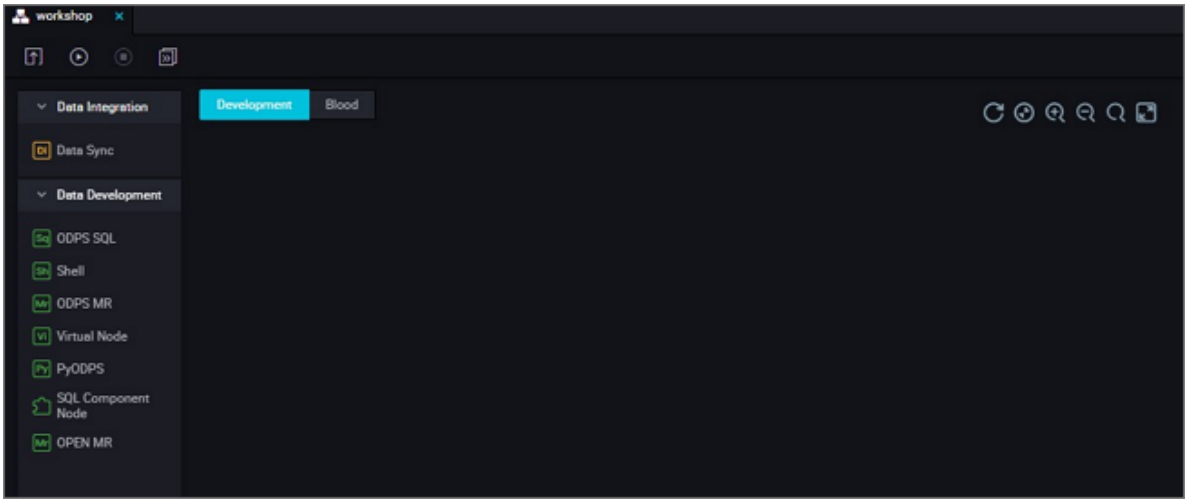
3. Click Test Connectivity, and after the connectivity test passes, click Finish to save the configuration.

## Create a Business Flow

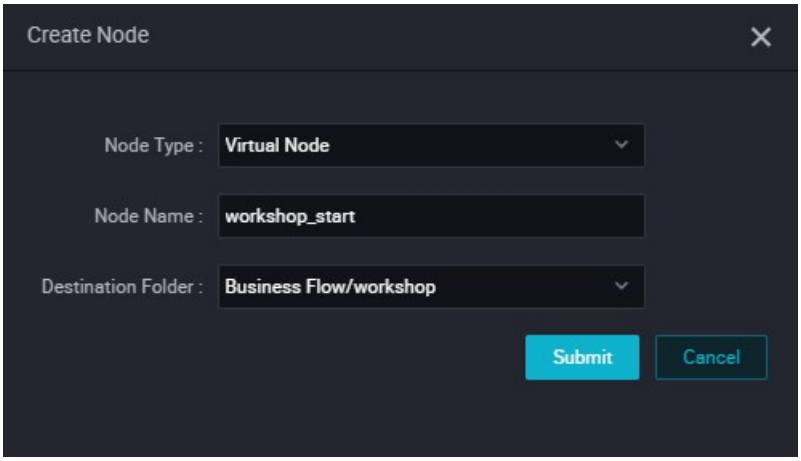
1. Right-click Business Flow under Data Analytics, select Create Business Flow.
2. Fill in the Business Flow name and description.

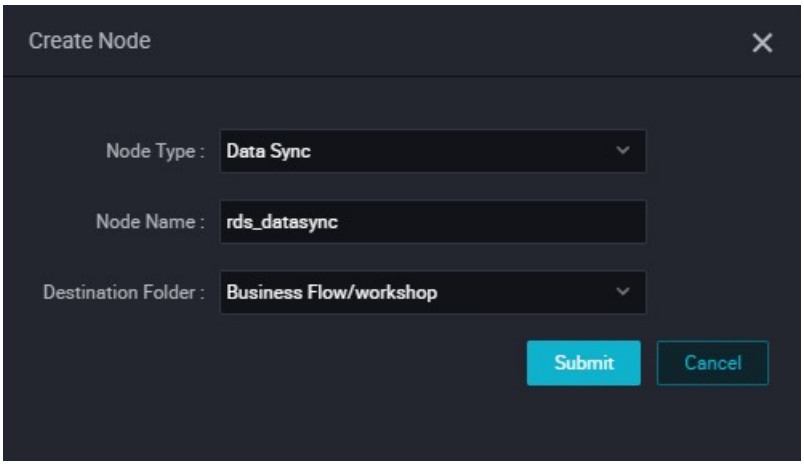


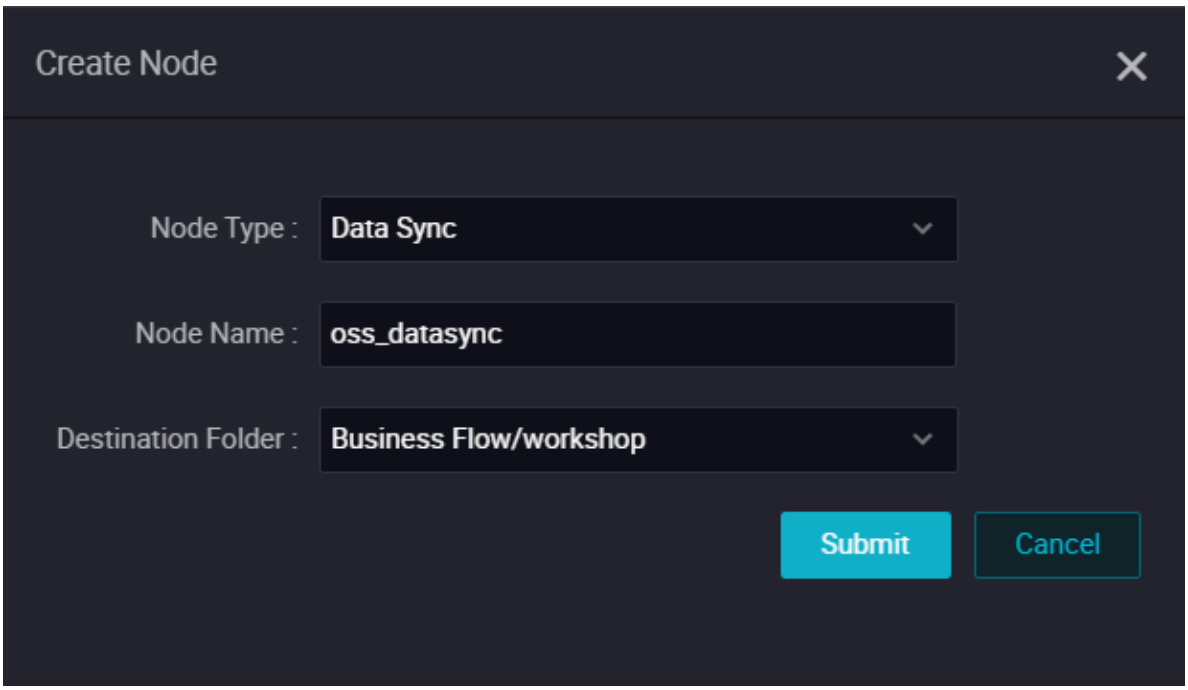

3. Click Create to complete the creation of the Business Flow.



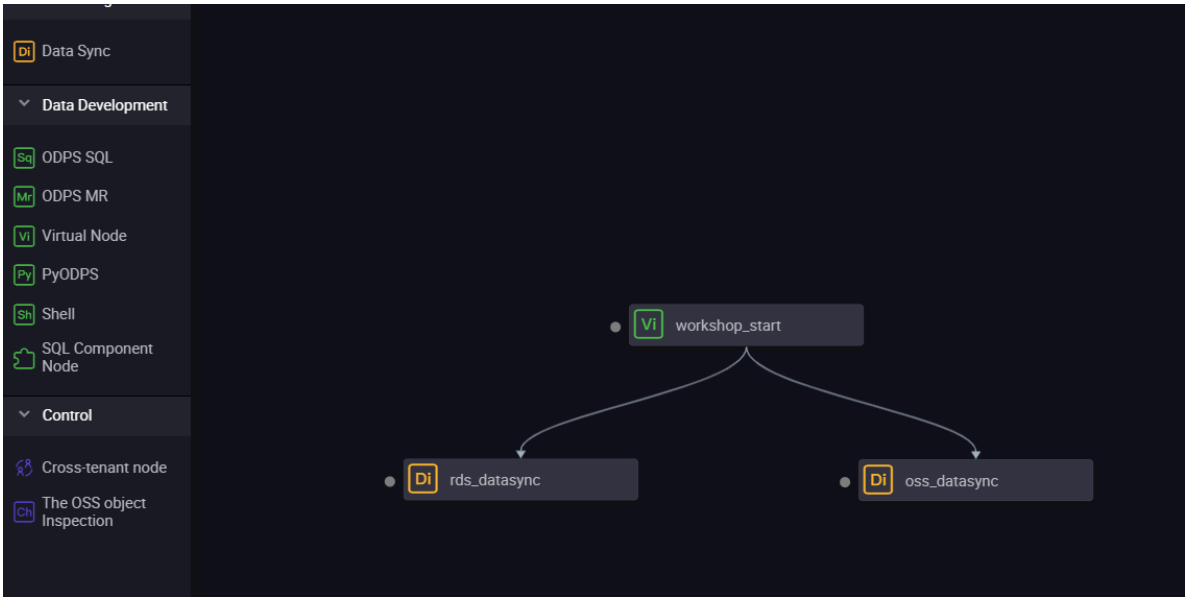
4. Enter the Business Flow Development Panel and drag a virtual node and two data sync nodes (oss_datasync and rds_datasync) into the Panel.



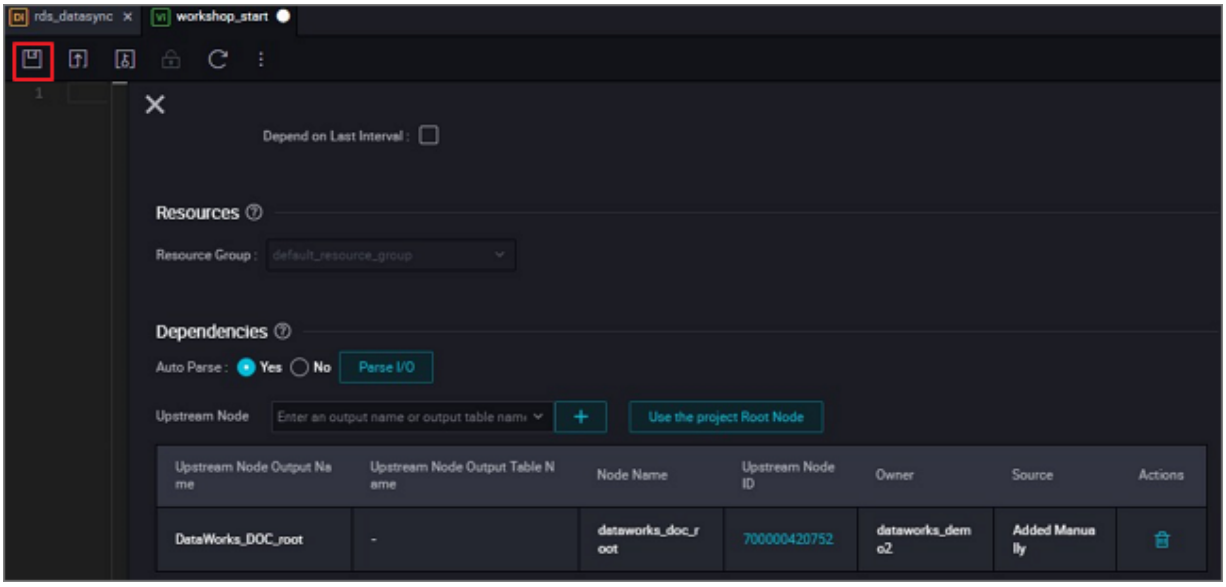
5. Drag the connection to set the workshop_start node to the upstream of both data synchronization nodes.




## Configure workshop_start task

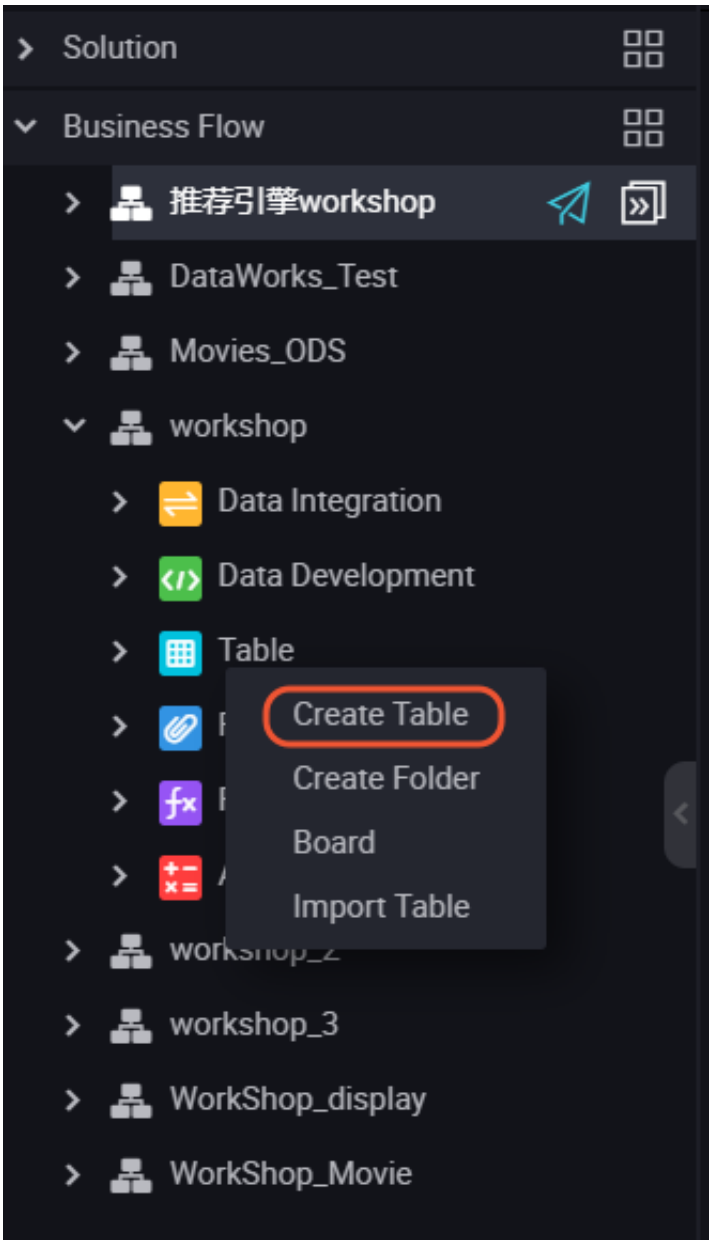
Since the new version sets the input and output nodes for each node, you need to set an input for the workshop_start node, the virtual node in the Business Flow can be set to the upstream node as the project root node, the project root node is generally named project name _ root.

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

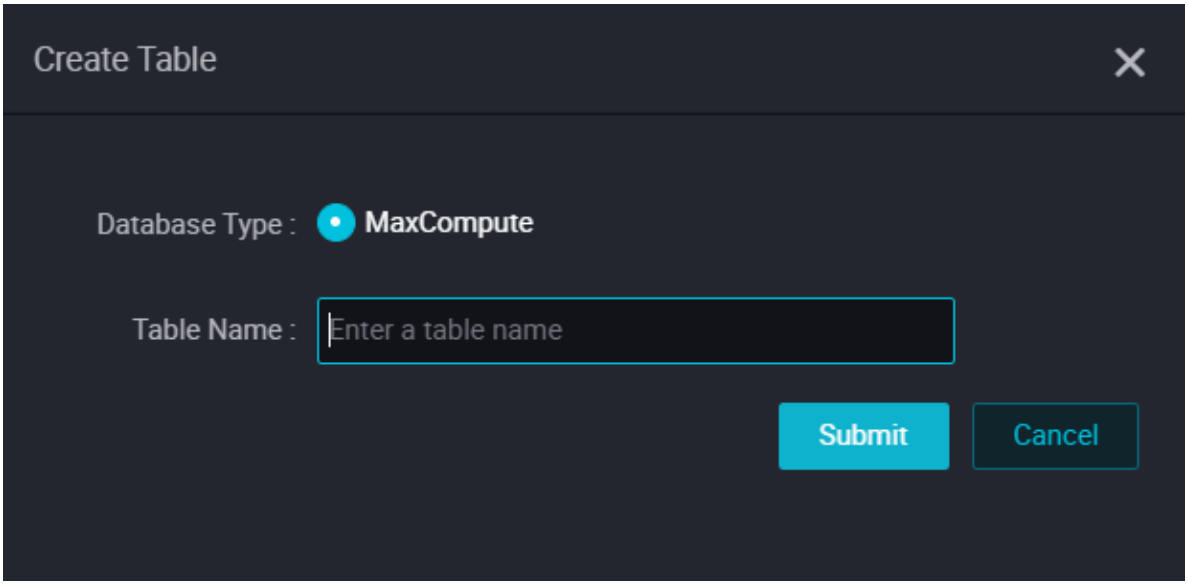


### Create Table

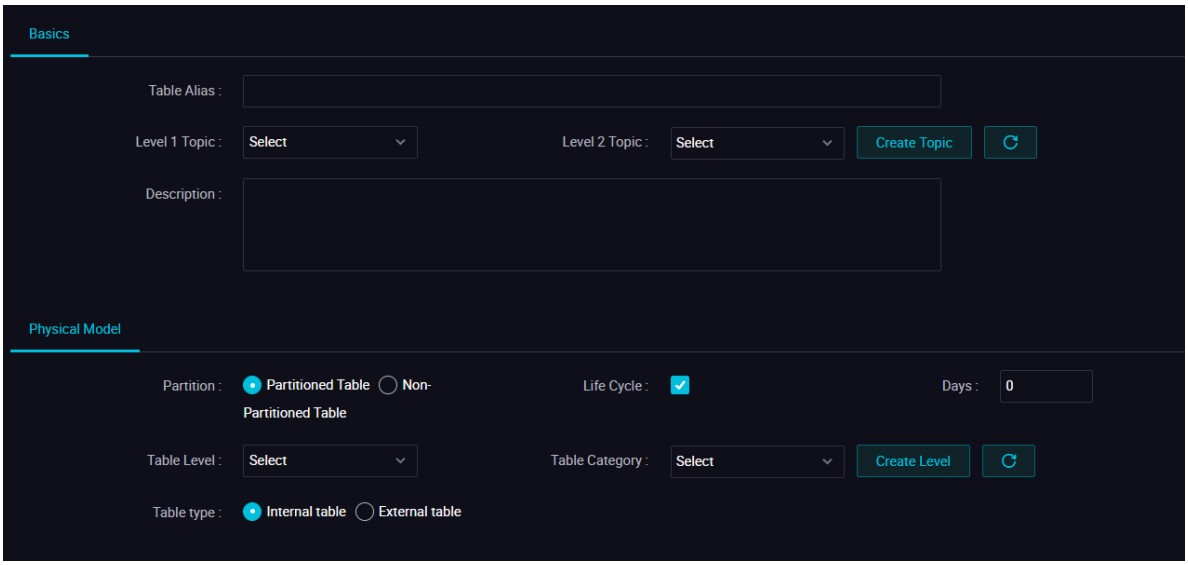
1. Right-click Table and choose Create Table.



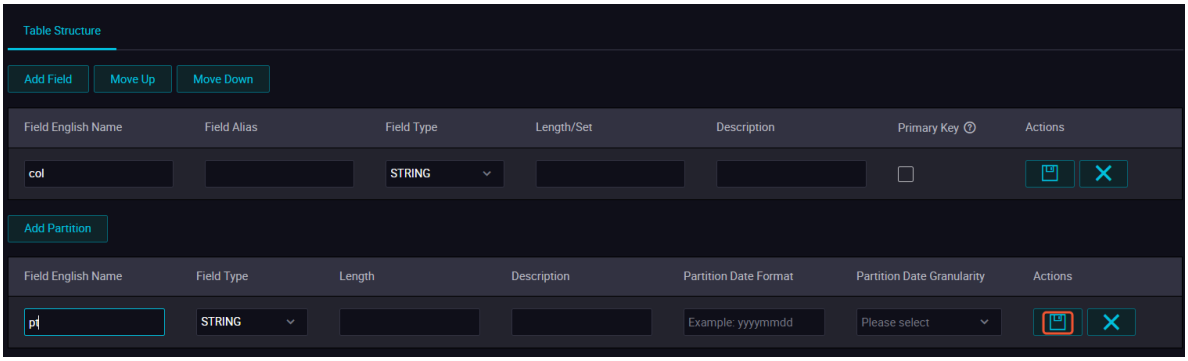
2. Type in Table Name(ods_raw_log_d and ods_user_info_d) for oss logs and RDS respectively.




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.

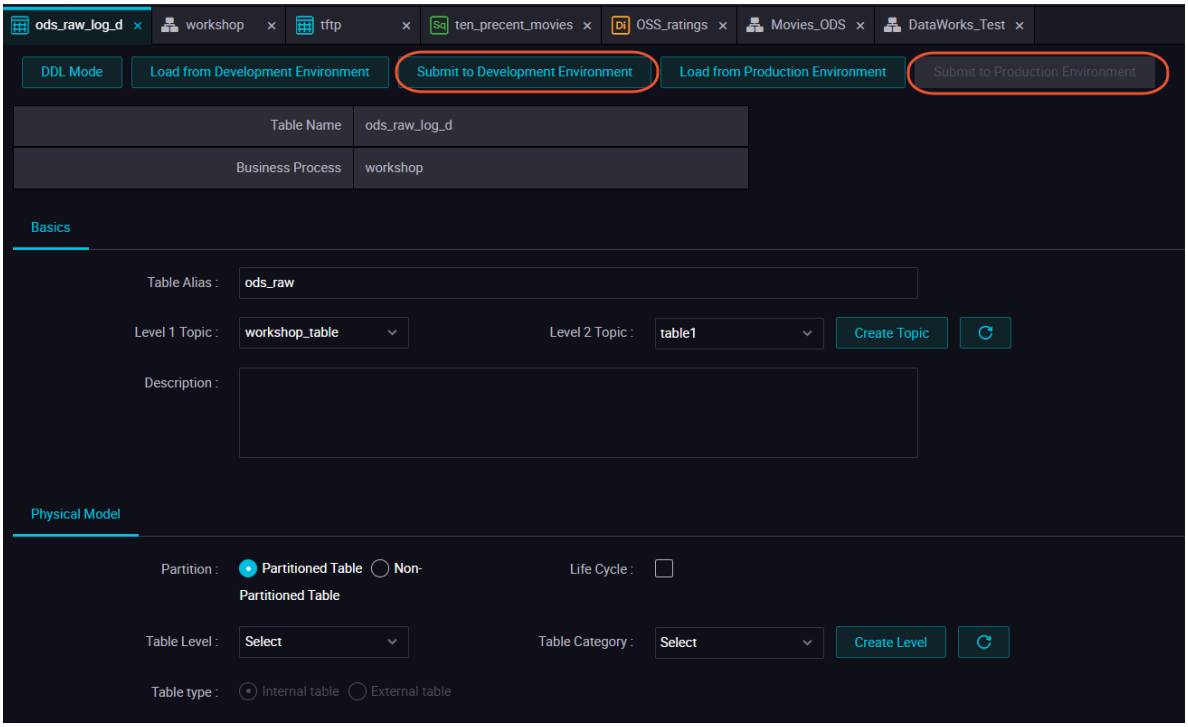



You can also click DDL Mode, use the following SQL statements to create tables.

```
// Create 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.

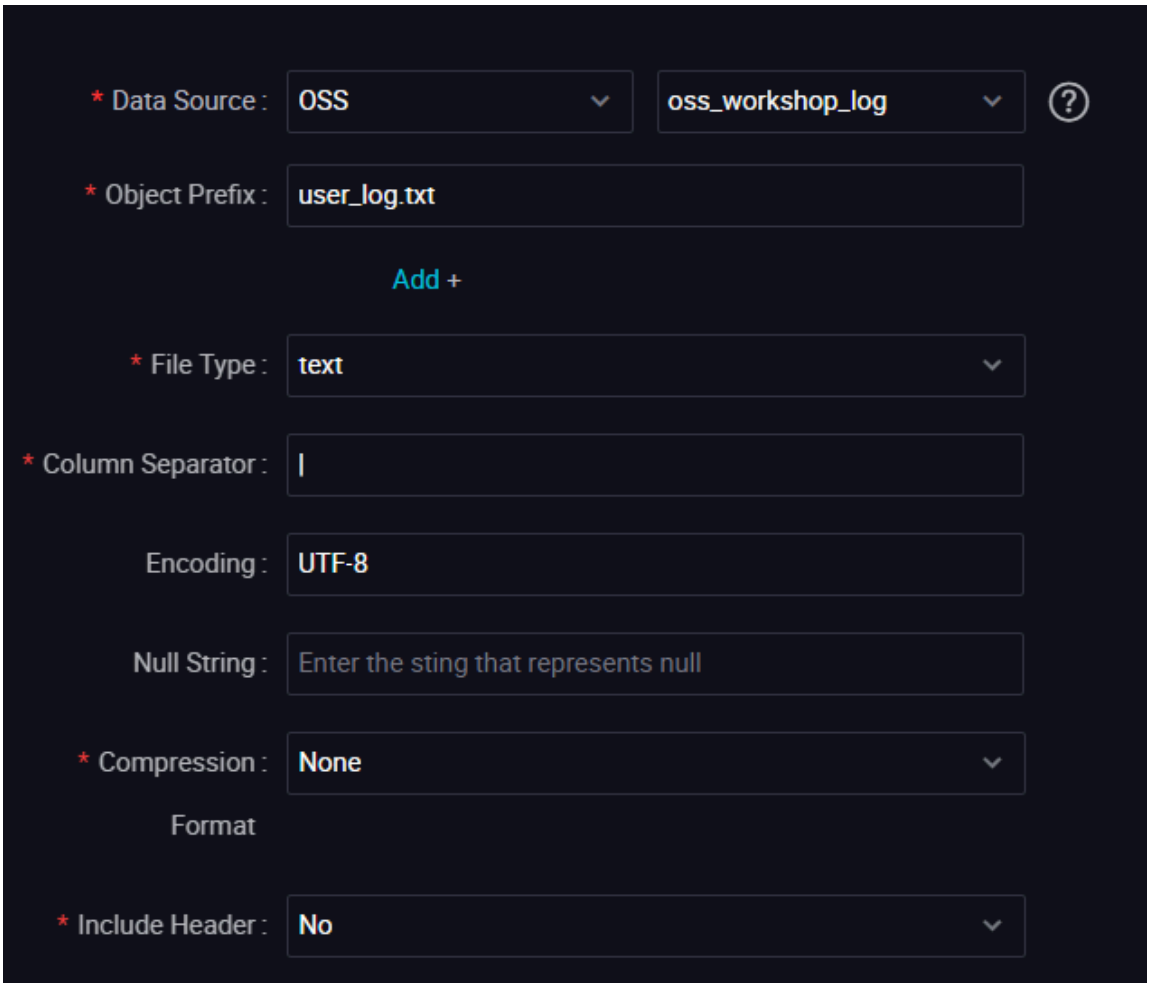


### Configure the data synchronization task

- 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.

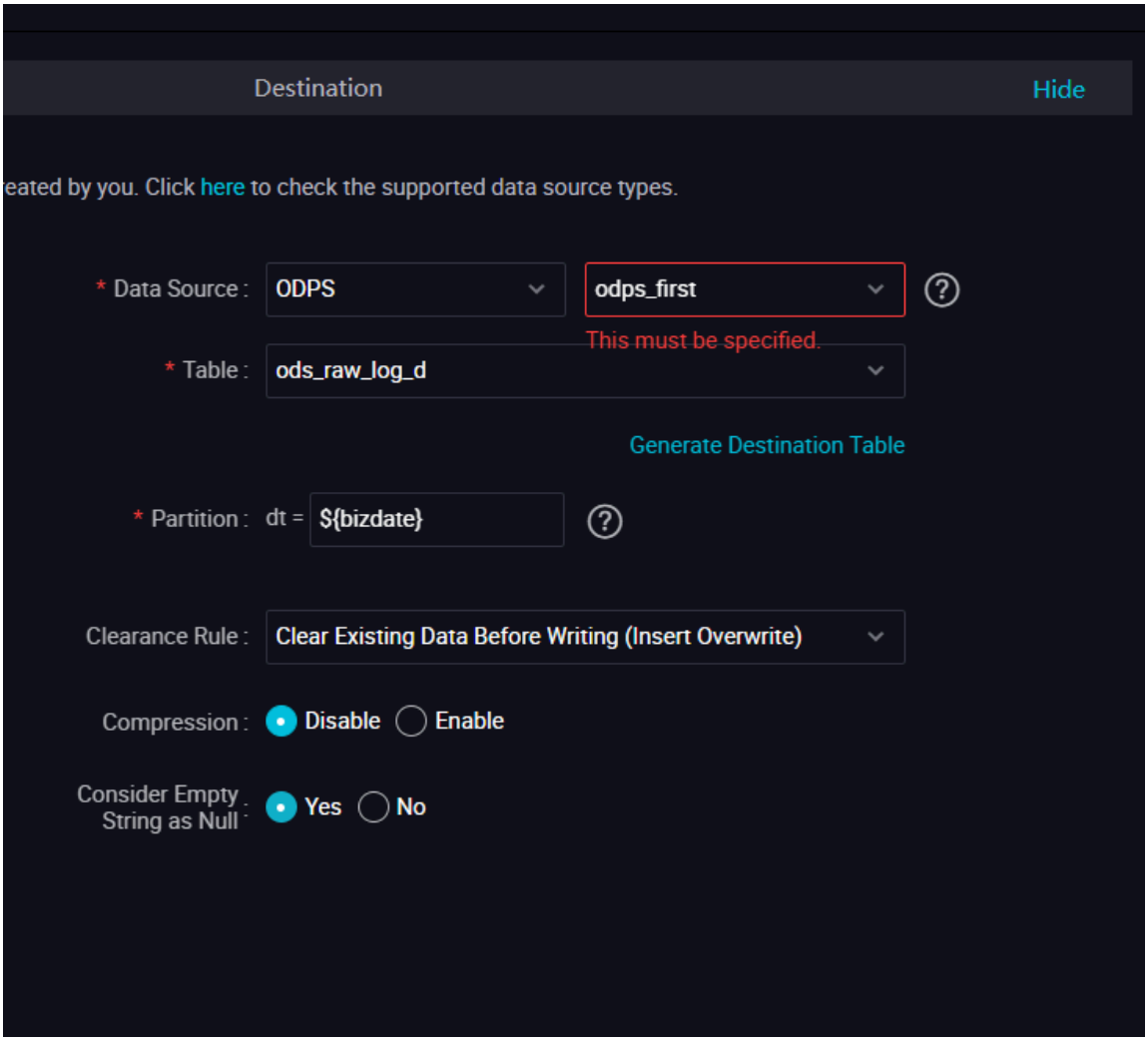



   Parameters:

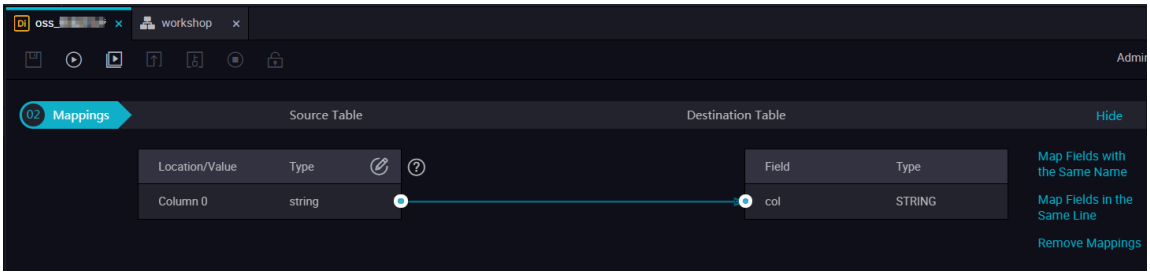
   - Data source: oss_workshop_log
   - Object Prefix: /user_log.txt
   - Column Separator: |

3. Select data destination

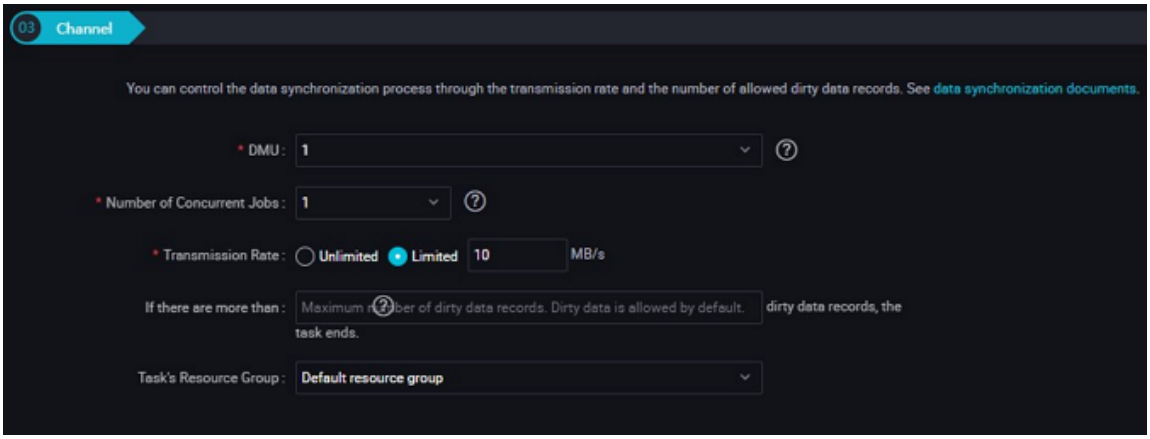
   Select the data destination is ods_raw_log_d in the odps_first data source. Both partition information and cleanup rules take the system default, the default configuration of the partition is ${bizdate}.




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.

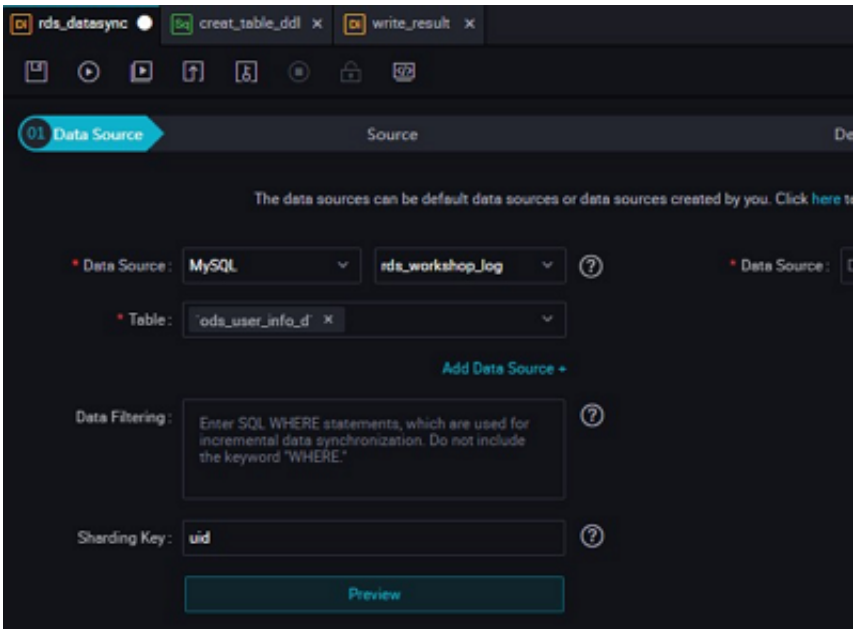



6. Verify that the current task is configured and can be modified. After the confirmation is correct, click Save in the upper left corner.
7. Closes the current task and returns to the Business Flow configuration panel.

- Configure the rds_datasync node Node

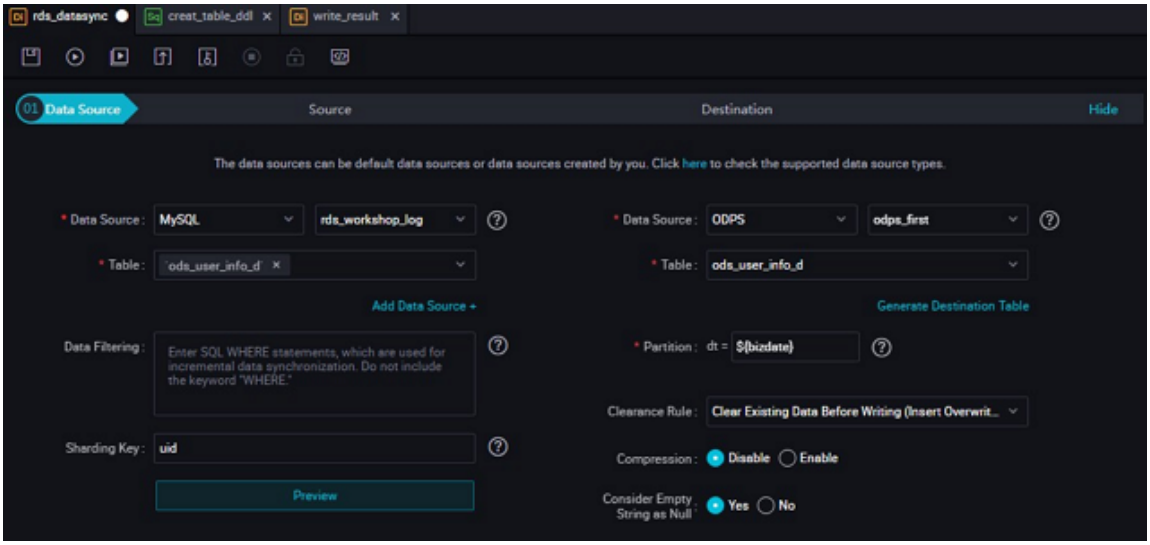
1. Double-click the rds_datasync node node to go to the node configuration page.
2. Select a data source.

   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.

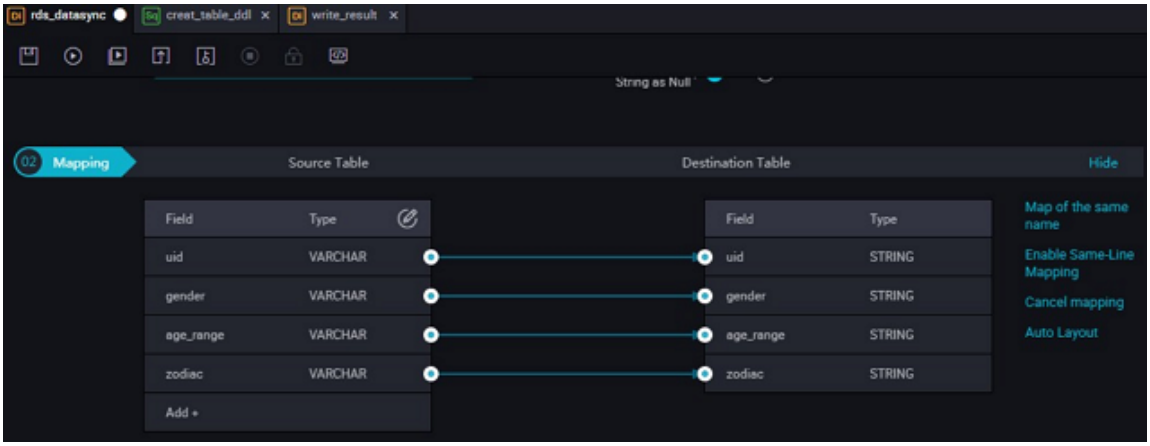



3. Select data destination

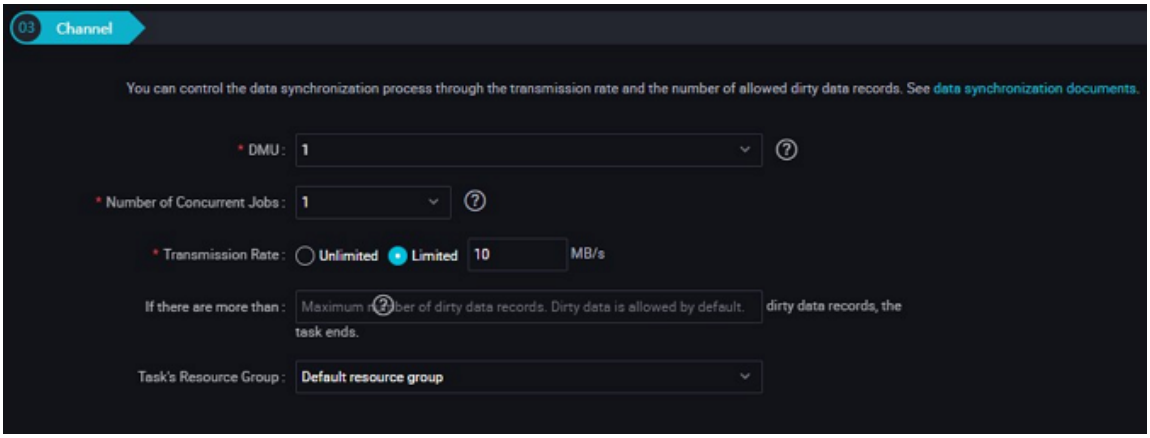
   Select the data destination ods_user_info_d in the data source named odps_first. Both partition information and cleanup rules take the system default, the default configuration of the partition is ${bizdate}.



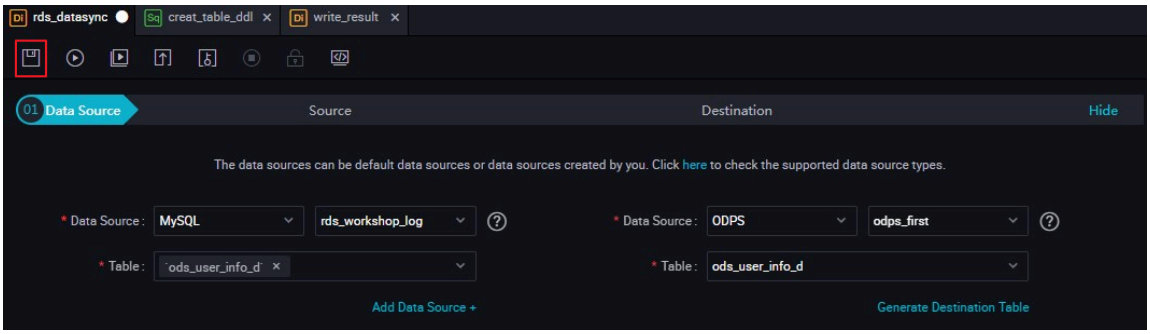

4. Configure the field mapping, default in association with the name mapping.




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.



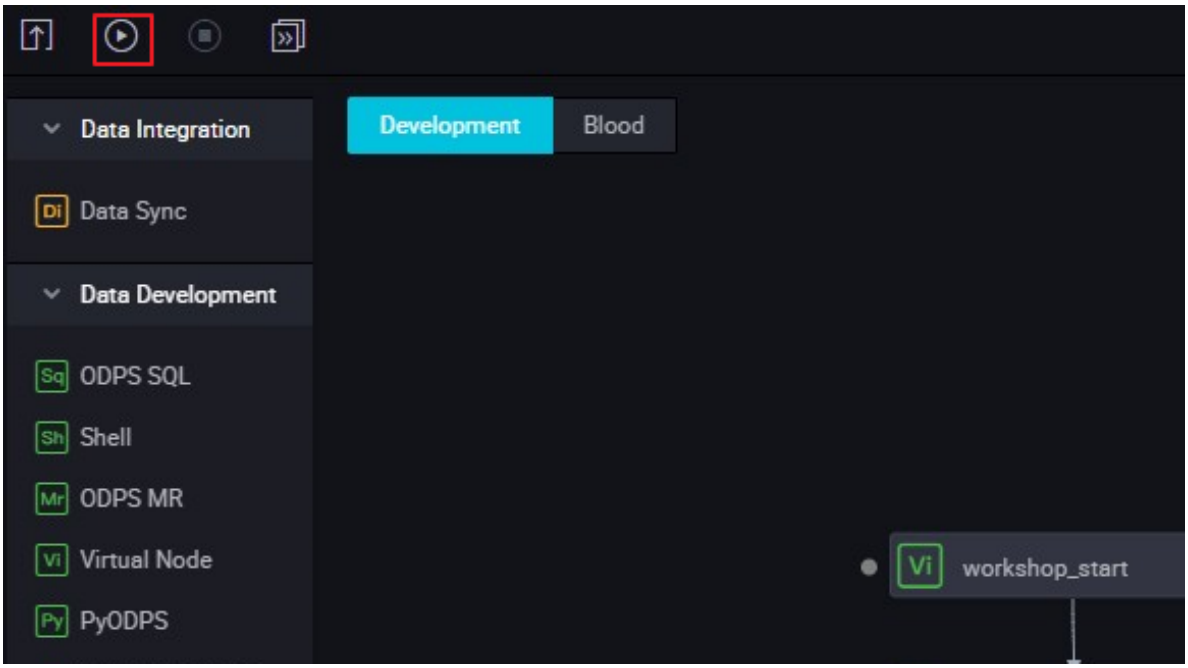

7. Closes the current task and returns to the Business Flow configuration panel.

## 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

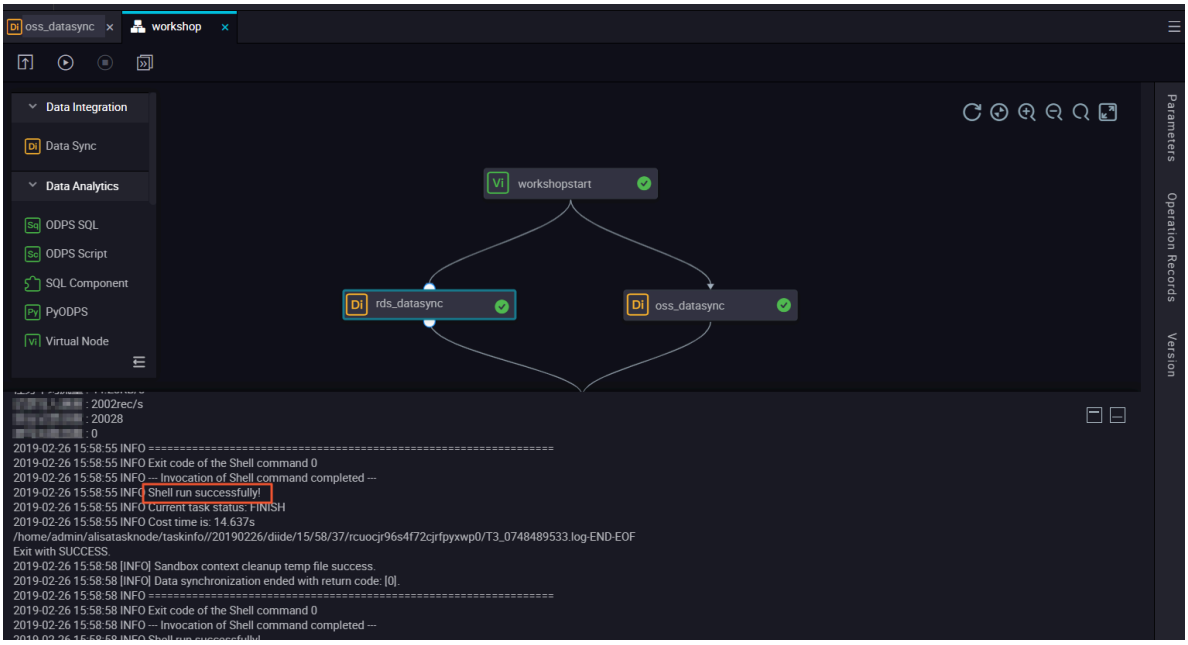
1. Click Run.




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

2. Right-click the rds_datasync task and select View Log.

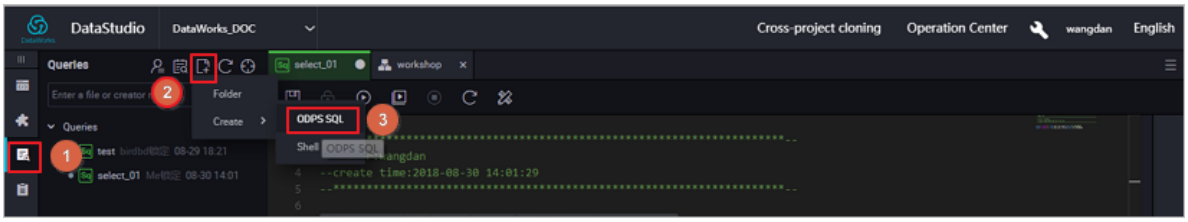
When the following words appear in the log, it indicates that the synchronization task runs successfully and synchronizes a batch of data.



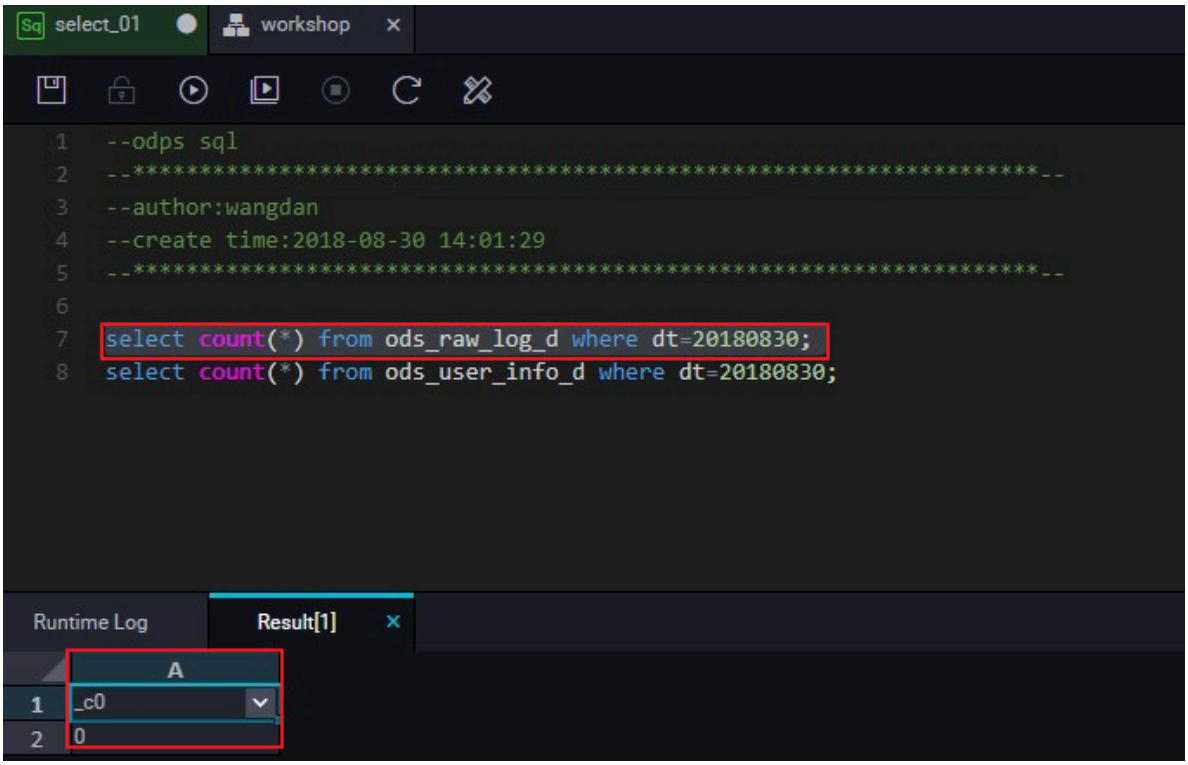

3. Right-click the oss_datasync task and select View Log. The confirmation method is consistent with the rds_datasync task.

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_lo g_d where dt = business
  date ;
```

```
select count (*) from ods_user_i nfo_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*.

## 1.3 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_in fo_d (
  ip string comment ' IP address ',
  uid STRING COMMENT ' User ID ',
  time string comment ' time : yyyymmddhh : mi : ss ',
  status string comment ' server return status code ',
  bytes string comment ' the number of bytes
returned to the Client ',
  region string comment ' region , get from IP ',
  method string comment ' HTTP request type ',
  url string comment ' url ',
  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 '
)
PARTITIONE D 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_in  fo_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  ' region ,  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 : yyyymmddhh : mi : ss '
)
 PARTITIONE  D   BY  (
   dt   string
);
```

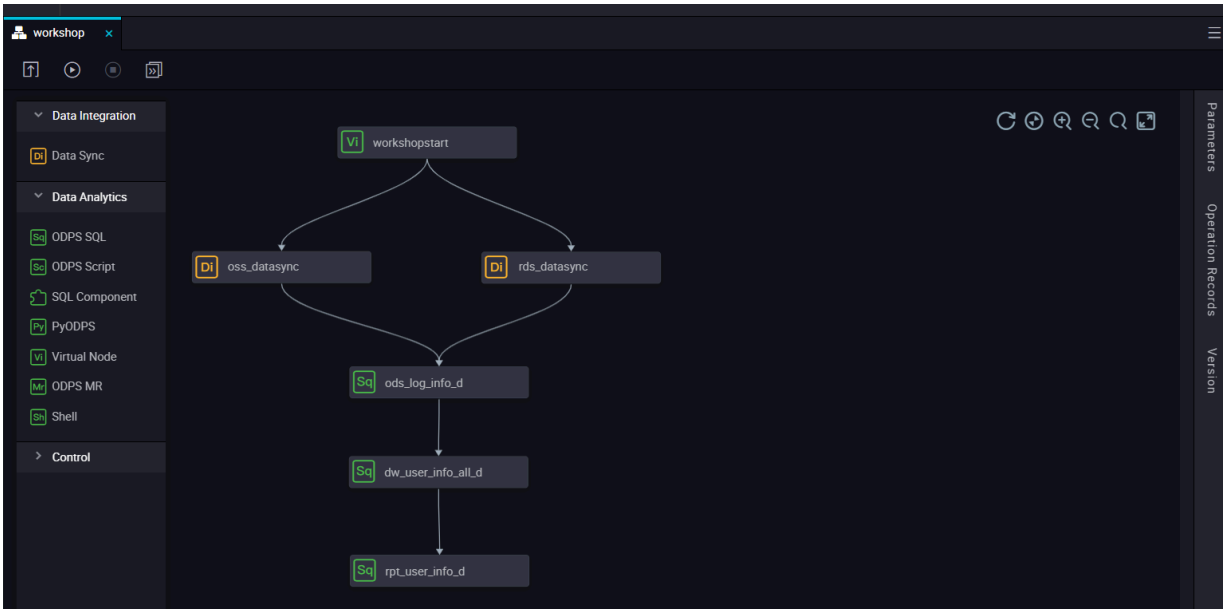
- Create rpt_user_info_d table

  The following are table creation statements:

```
--  Create   a   copy   table
 Create   Table   if   not   exists   rpt_user_i  nfo_d (
   uid   STRING   COMMENT  ' User   ID ',
   region   string   comment  ' region ,  get   from   IP ',
   device   string   comment  ' terminal   type ',
   pv   bigint   comment  ' pv ',
   gender   STRING   COMMENT  ' Gender ',
   age_range   STRING   COMMENT  ' Age   range ',
   zodiac   STRING   COMMENT  ' Zodiac   sign '
)
 PARTITIONE  D   BY  (
   dt   string
);
```

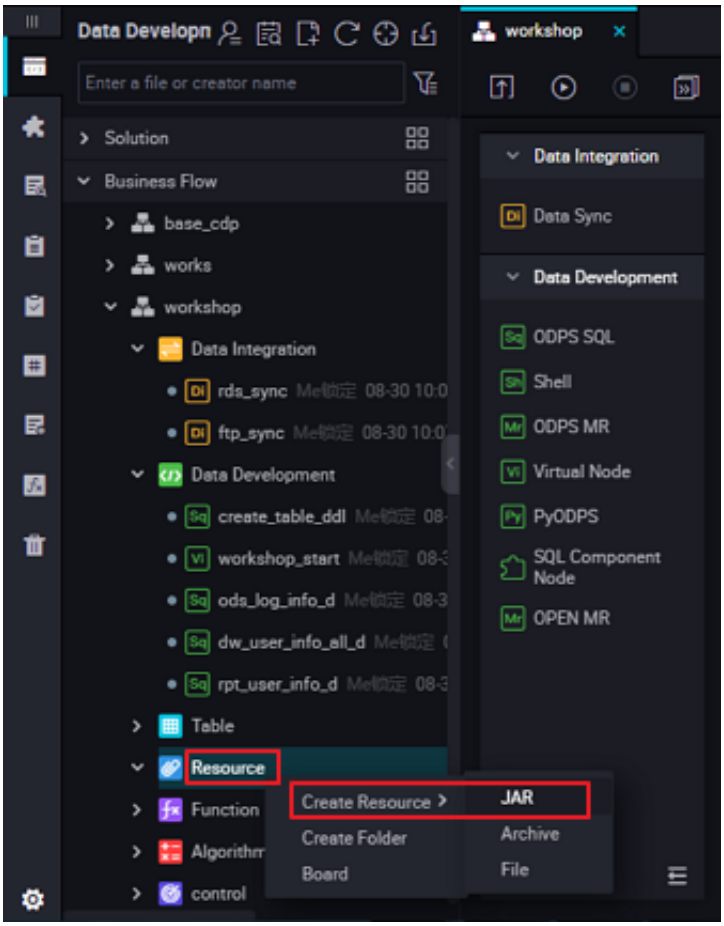
### Business Flow Design

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.

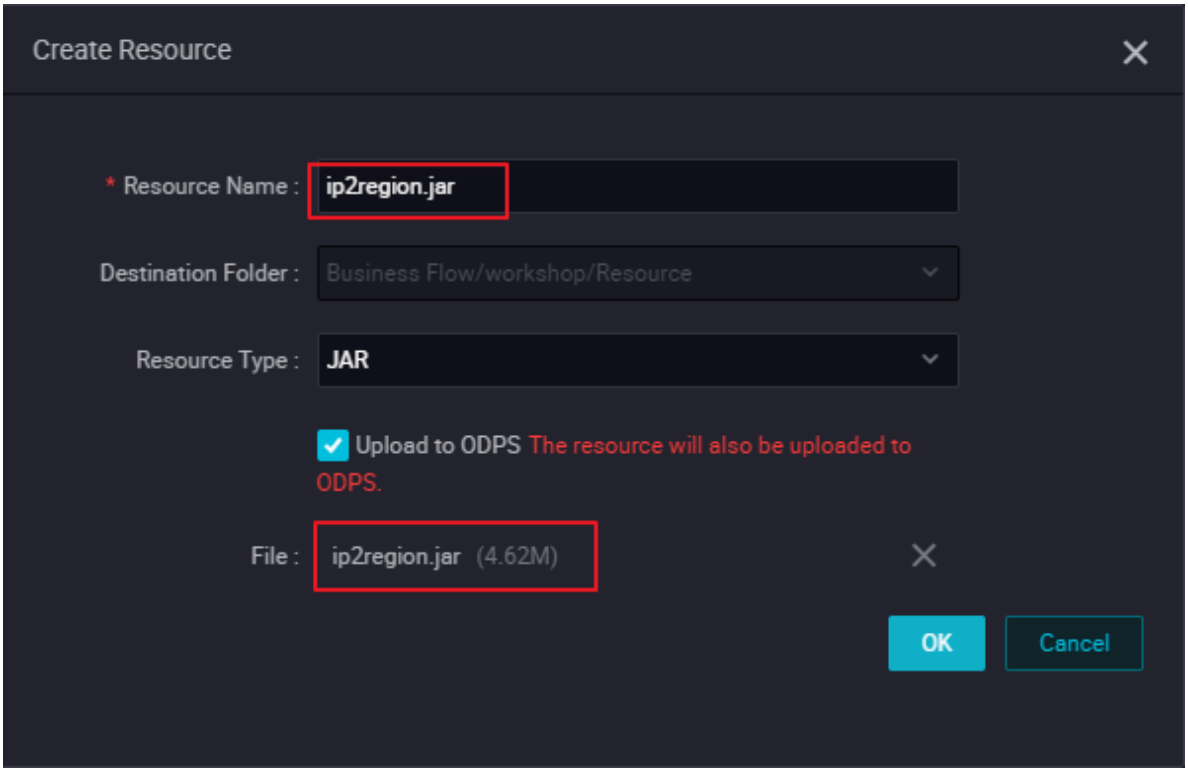



## Creating user-defined functions

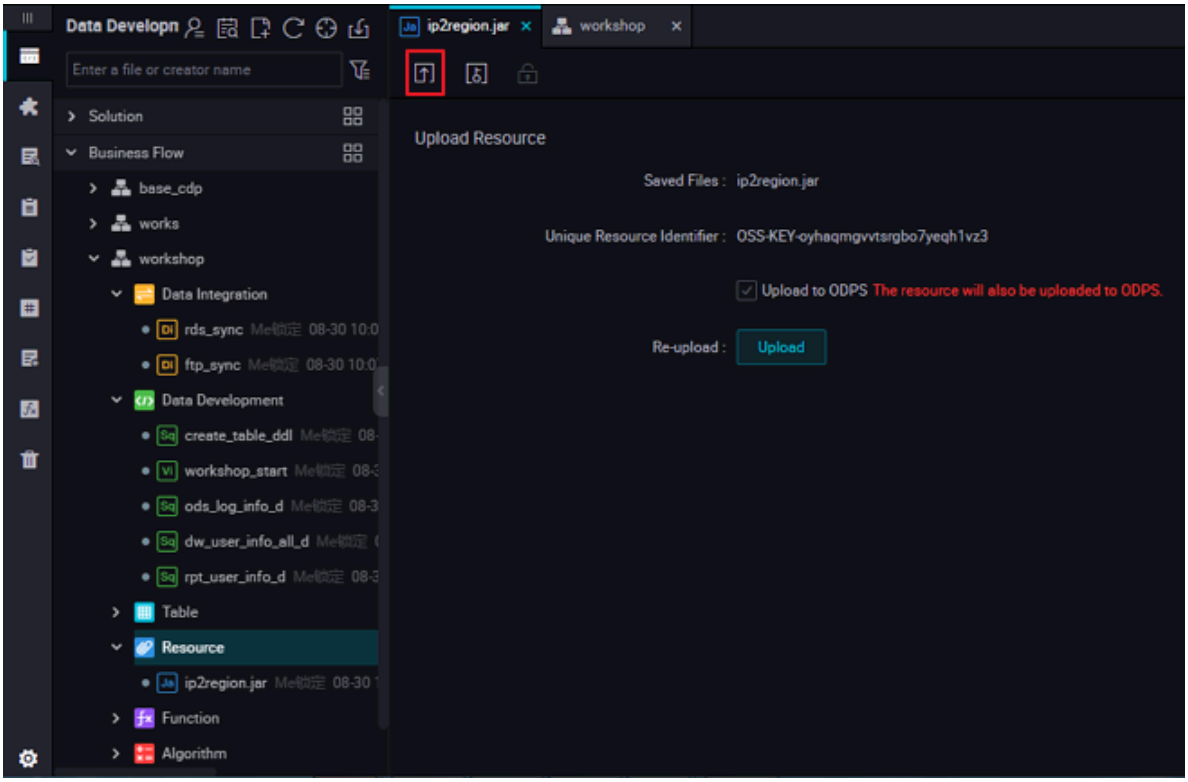
1. Download *ip2region.jar*.
2. Right-click Resource, and select Create Resource > jar.



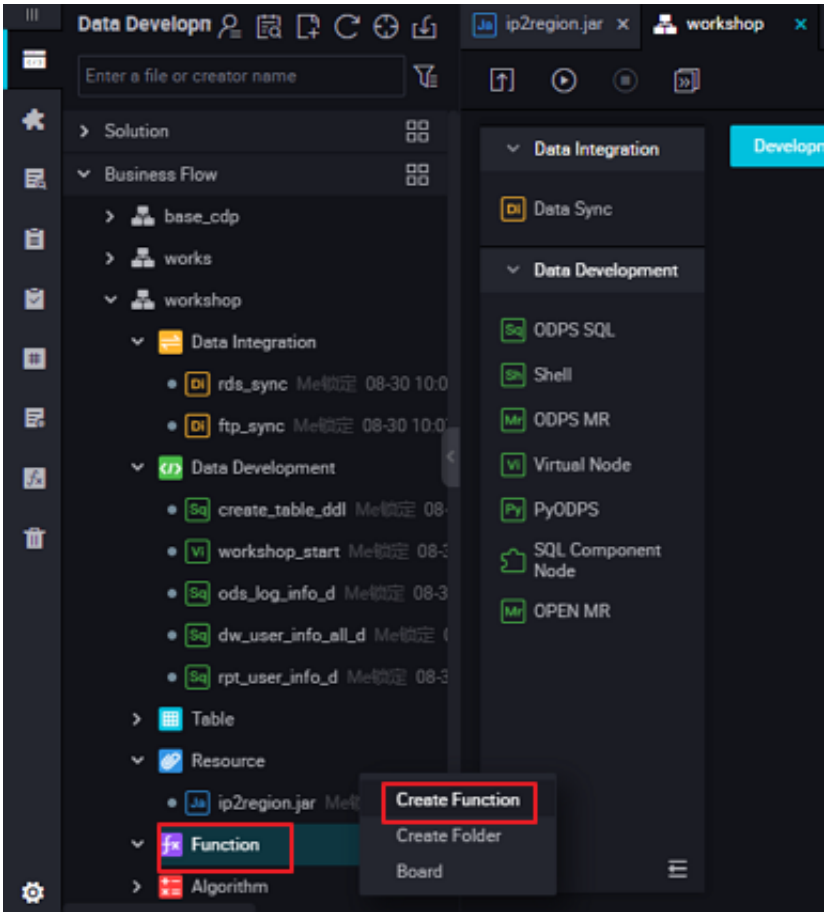
3. Click Select File, select ip2region. jar that has been downloaded locally, and click OK.



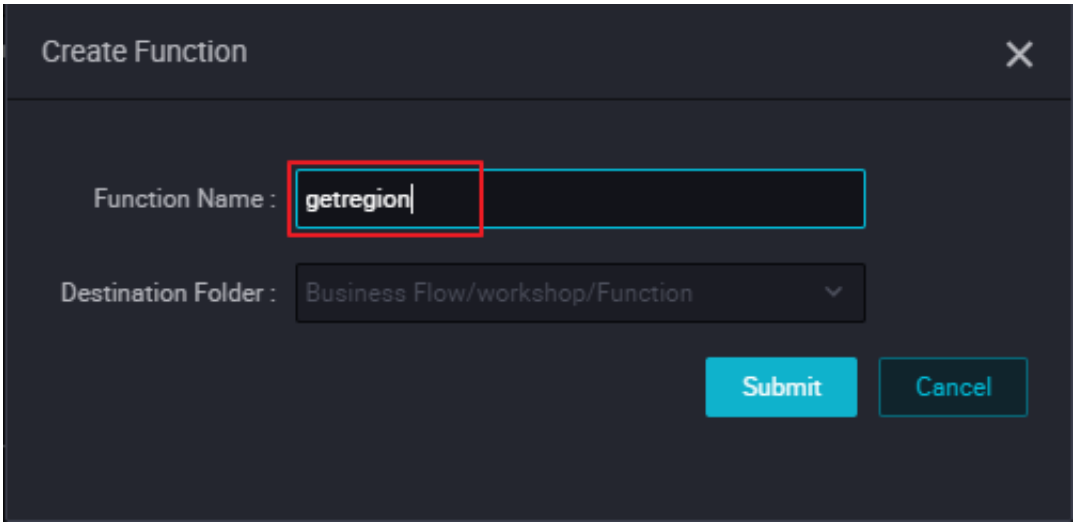

4. After the resource has been uploaded to dataworks, click Submit.



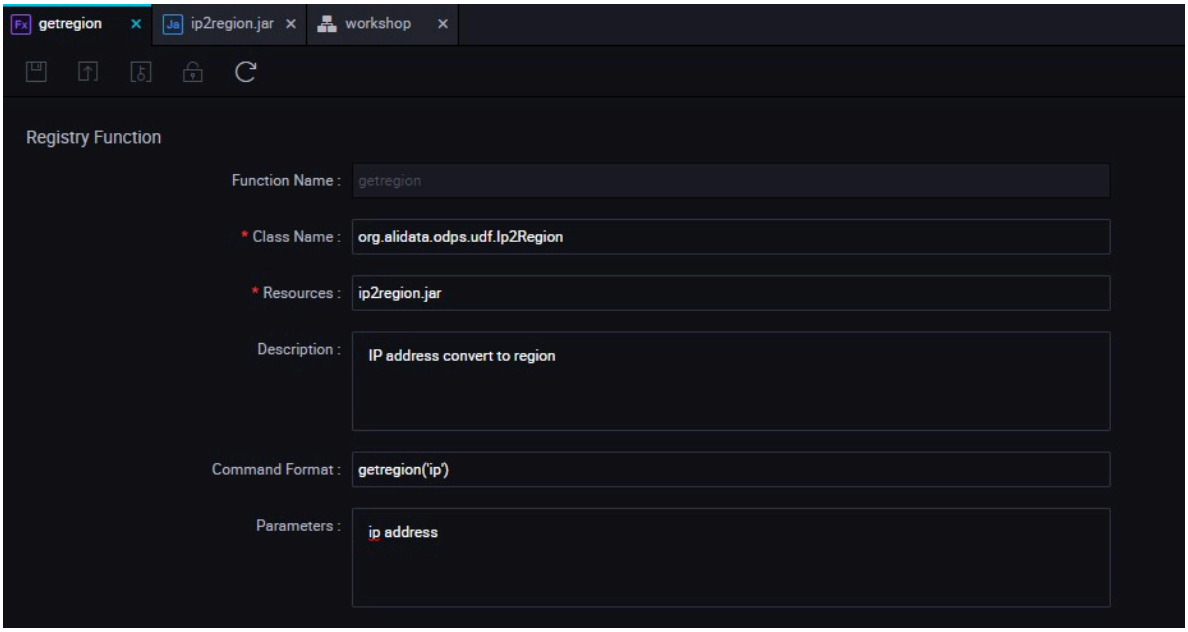
5. Right-click a function and select Create Function.




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:

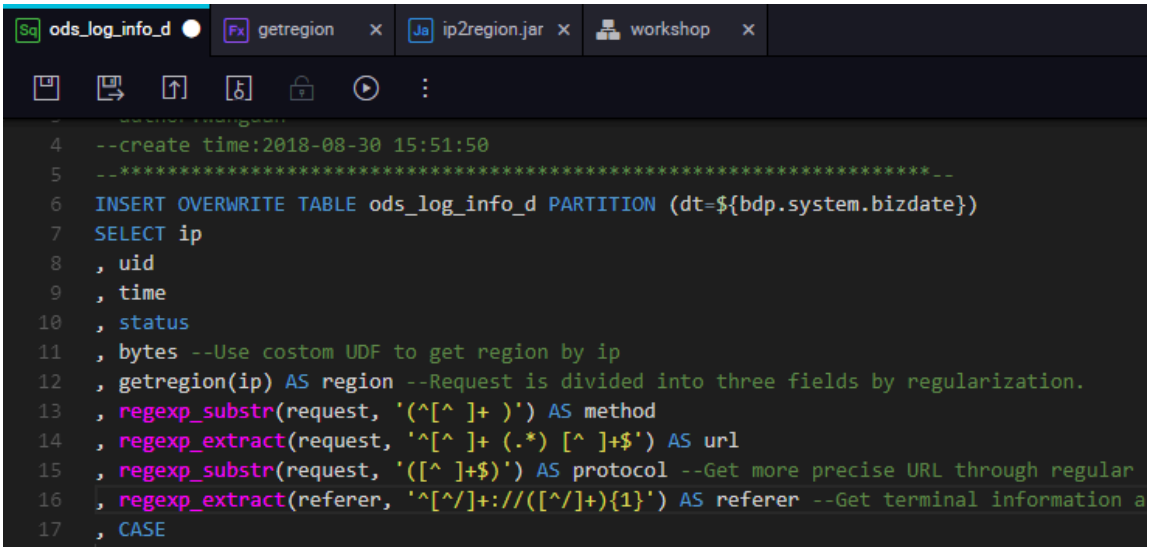
- Function Name: getregion
- Class Name: org.alidata.odps.udf.Ip2Region
- Resource list: ip2region. Jar
- Description: IP address translation area
- Command Format: getregion ('IP ')
- Parameter description: IP Address

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.

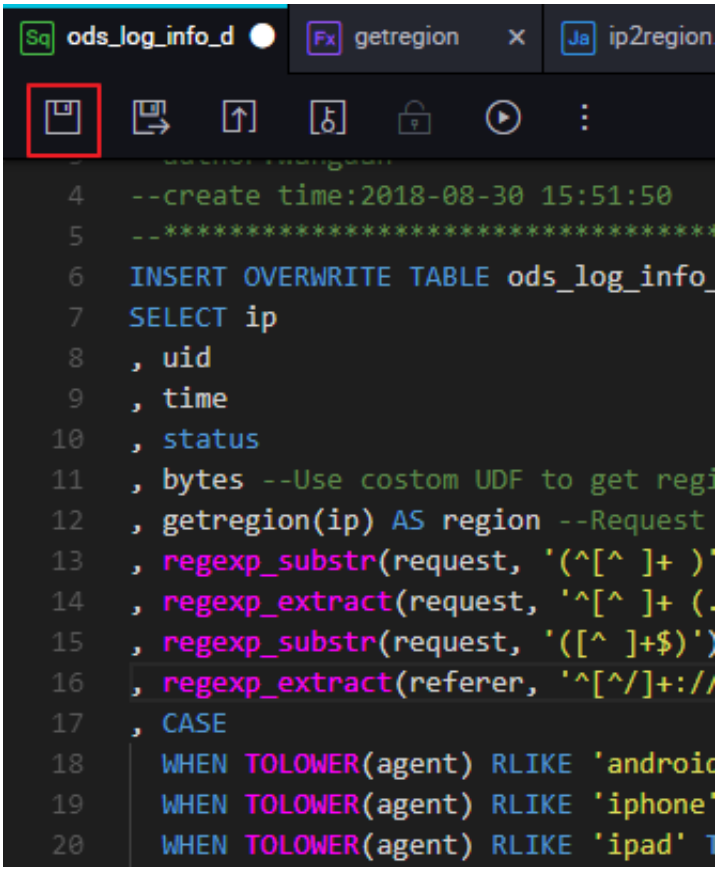



The SQL logic is as follows:

```
 INSERT   OVERWRITE   TABLE   ods_log_in  fo_d   PARTITION  ( dt =
${ bdp . system . bizdate })
 SELECT   ip
  ,  uid
  ,  time
  ,  status
  ,  bytes  --  use   a   custom   UDF   to   get   a   locale
 over   IP
  ,  getregion  ( ip )  as   region  --  the   request
 difference   is   divided   into   three   fields   through   the
   regular
  ,  regexp_sub  str ( request , '(^[^ ]+ )')  AS   method
  ,  regexp_ext  ract ( request , '^[^ ]+ (.*) [^ ]+$')  AS   url
  ,  regexp_sub  str ( request , '([^ ]+$)')  AS   protocol  --
 get   more   precise   urls   with   regular   clear   refer
  ,  regexp_ext  ract ( referer , '^[^/]+://([^/]+){ 1 }')  AS
 referer  --  Get   terminal   informatio  n   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_ph  one '
     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_ext  ract ( request , '^[^ ]+ (.*) [^ ]+$')
RLIKE  ' feed '  THEN  ' feed '
    WHEN   TOLOWER ( agent )  NOT   RLIKE  '( bot | spider |
crawler | feed | slurp )'
    AND   agent   RLIKE  '^[ Mozilla | Opera ]'
    AND   regexp_ext  ract ( request , '^[^ ]+ (.*) [^ ]+$')  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 ( col , '##@@')[ 6 ]  AS   referer
   ,  SPLIT ( col , '##@@')[ 7 ]  AS   agent
  FROM   ods_raw_lo  g_d
  WHERE   dt  = ${ bdp . system . bizdate }
)  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_in fo_all_d PARTITION (
dt ='${ bdp . system . bizdate }')
SELECT COALESCE ( a . uid , b . uid ) AS uid
 , b . gender
 , b . age_range
 , b . zodiac
 , a . region
 , a . device
 , a . identity
 , a . method
 , a . url
 , a . referer
 , a . time
FROM (
  SELECT *
  FROM ods_log_in fo_d
  WHERE dt = ${ bdp . system . bizdate }
) a
LEFT OUTER JOIN (
  SELECT *
  FROM ods_user_i nfo_d
  WHERE dt = ${ bdp . system . bizdate }
) b
ON a . uid = b . uid ;
```

  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.

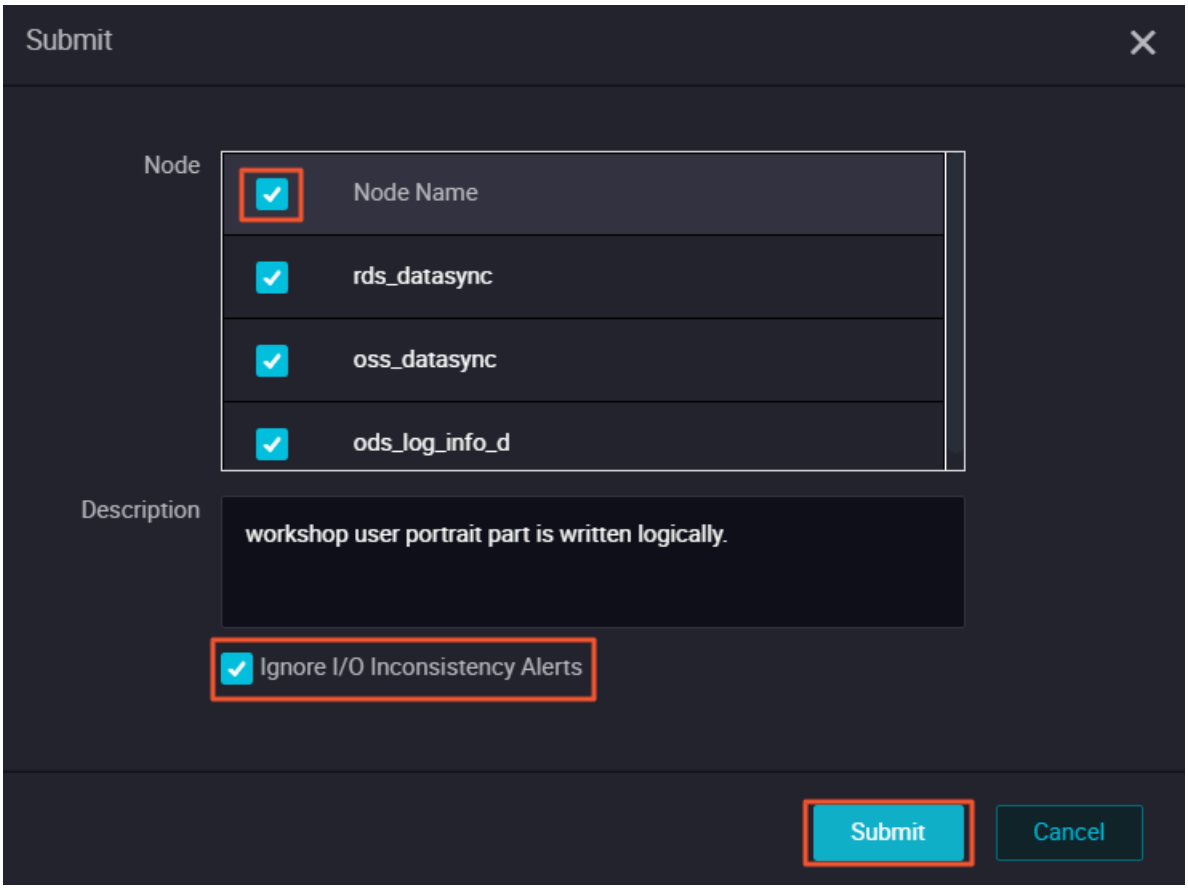
     The SQL logic is as follows:

```
INSERT OVERWRITE TABLE rpt_user_i nfo_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_in fo_all_d
WHERE dt = ${ bdp . system . bizdate }
GROUP BY uid ;
```

  2. Click Save.

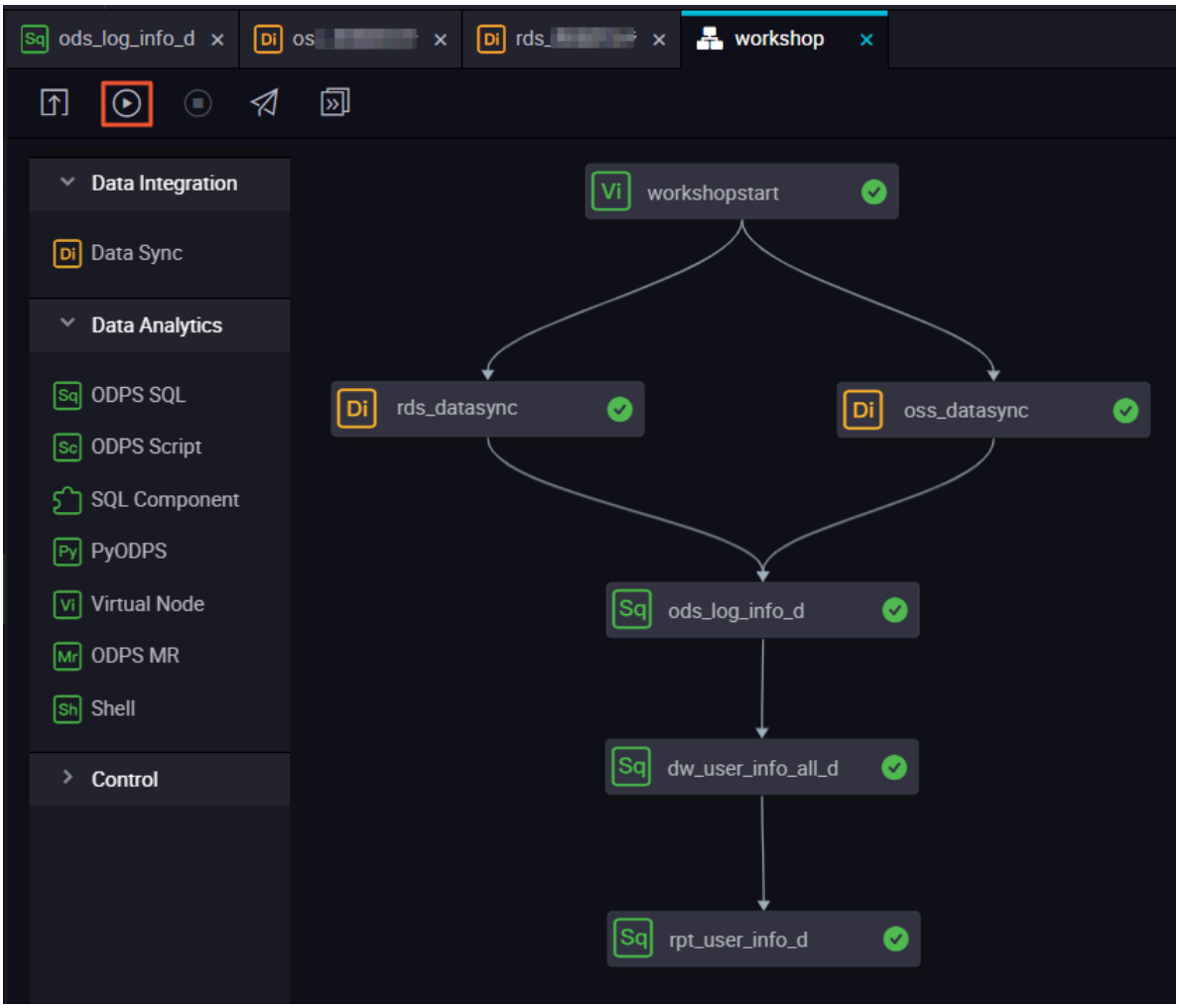
### Submitting Business Flows

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 Submitdialog box, and check the Ignore Warnings on I/O Inconsistency, click Submit.

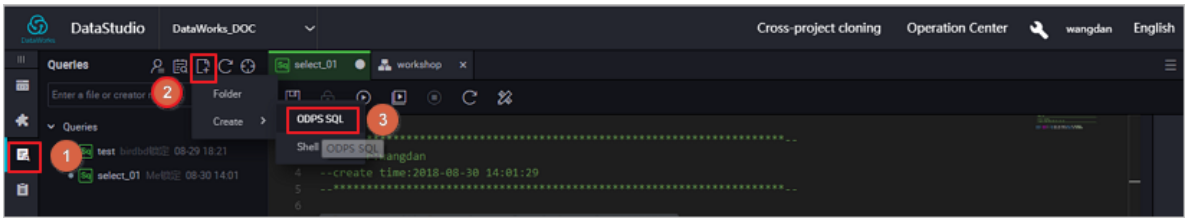


## Running Business Flows

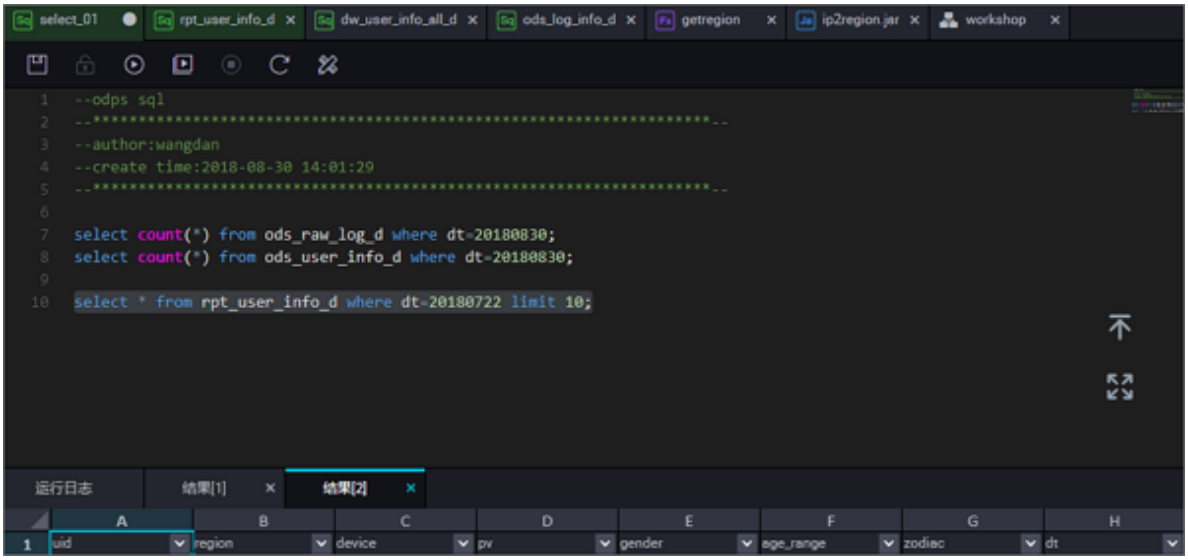
1. Click Run to verify the code logic.




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.



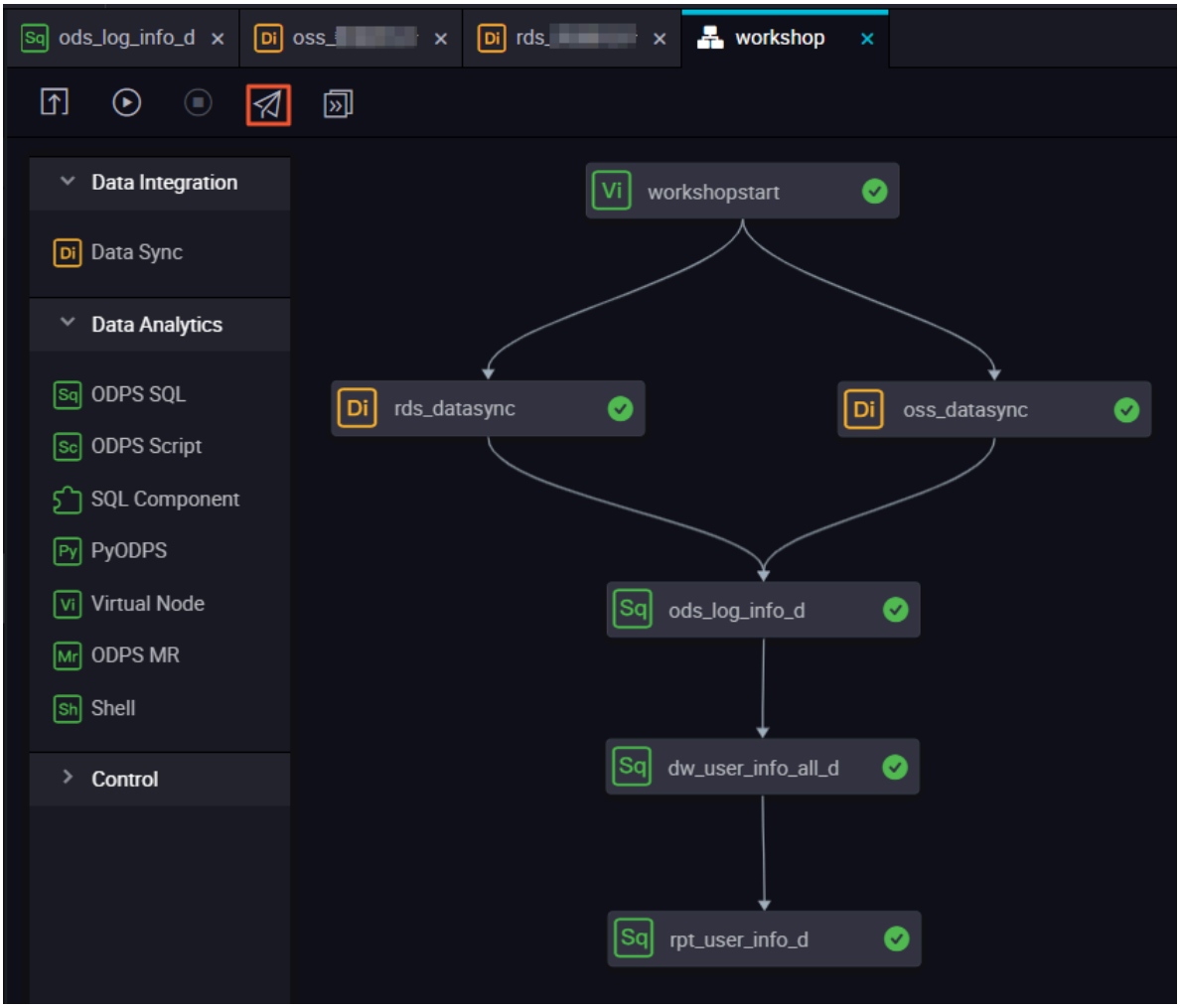

The query statement is as follows:

```
---  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.

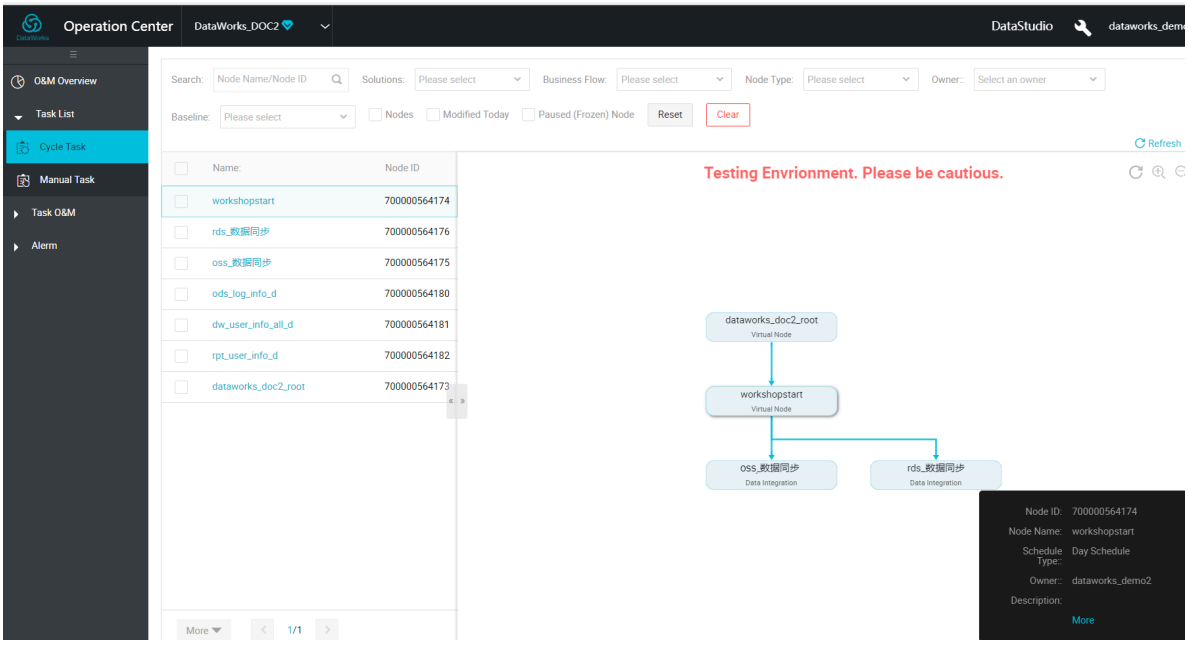



2. Select the task to publish and click Add To Be-Published List.
3. Enter the list of pending releases, and click Pack and publish all.
4. View published content on the Publish Package List page.

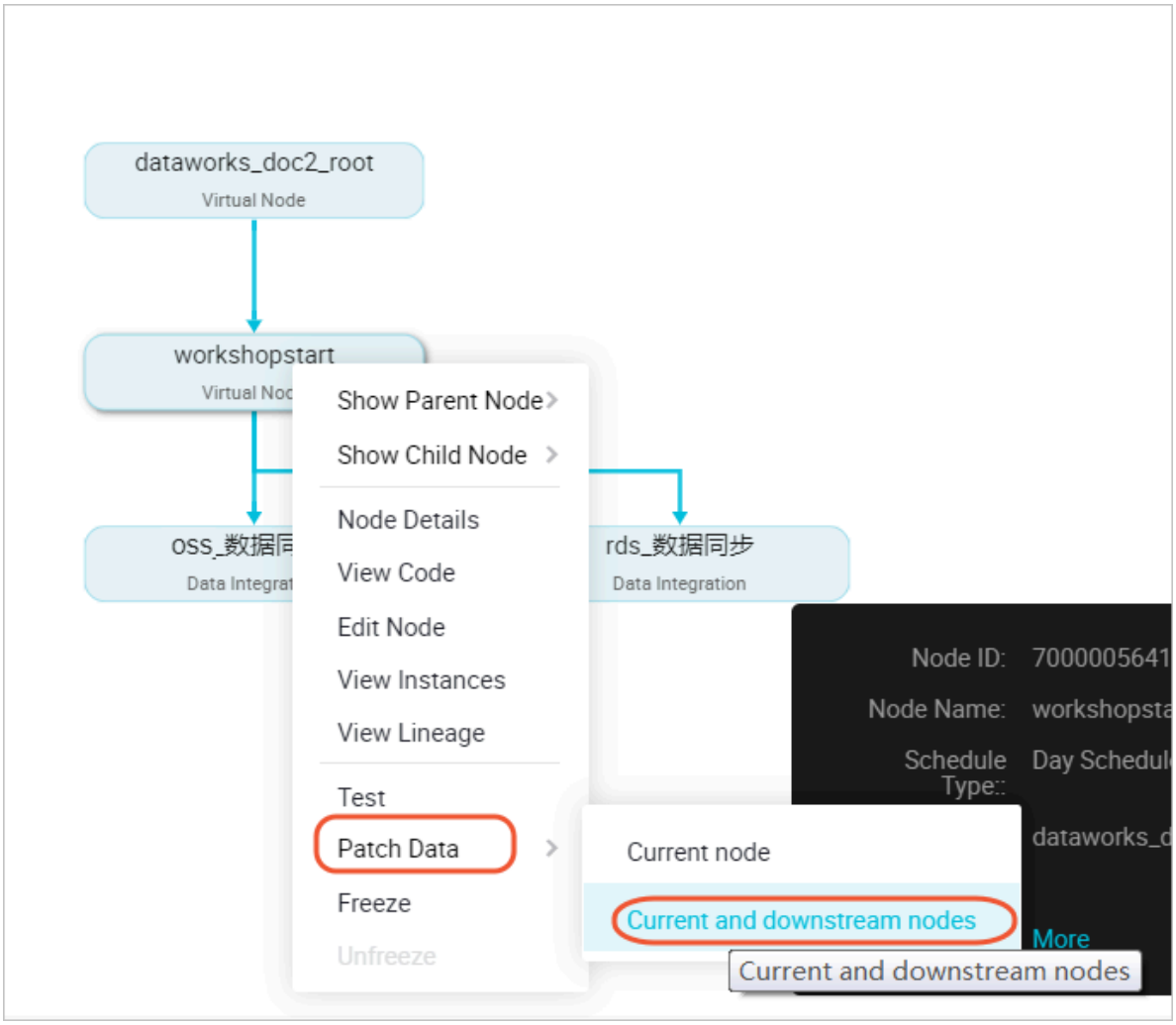
## Run tasks in production

1. After the task has been published successfully, click Operation center.

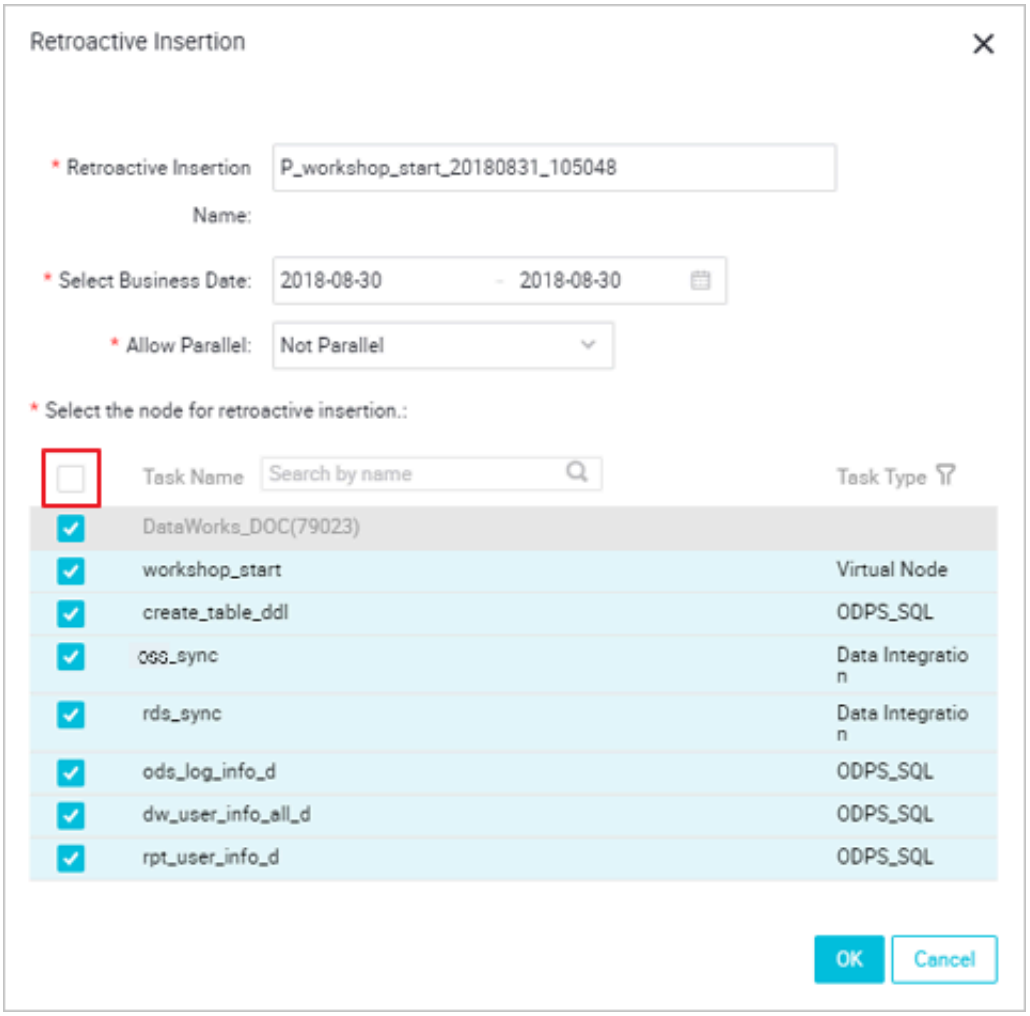
2. Select Workshop Business Flows in the Task List.




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

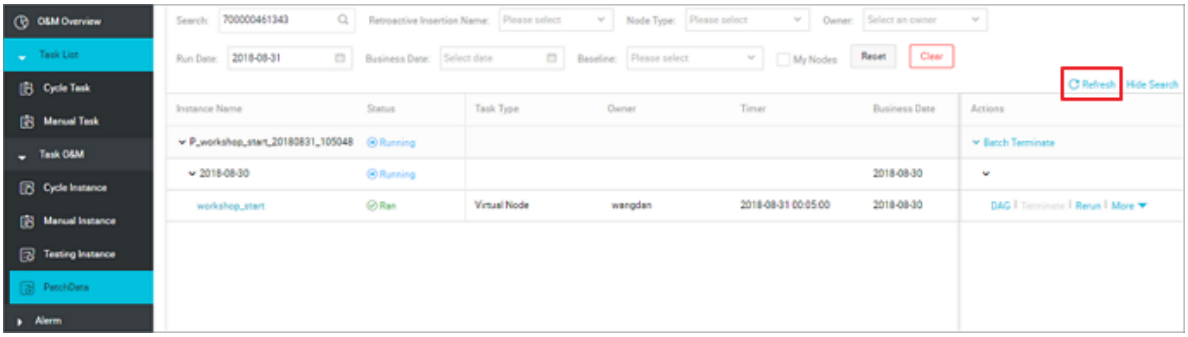


4. Check the task that needs to fill the data, enter the business date, and click OK.




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* .

# 1.4 Data quality monitoring

This topic 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.

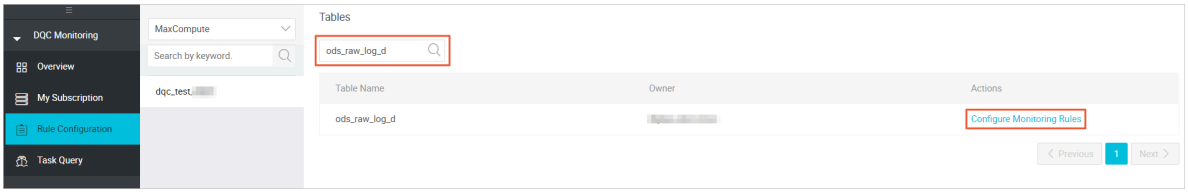
## Add Table Rule Configuration

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.

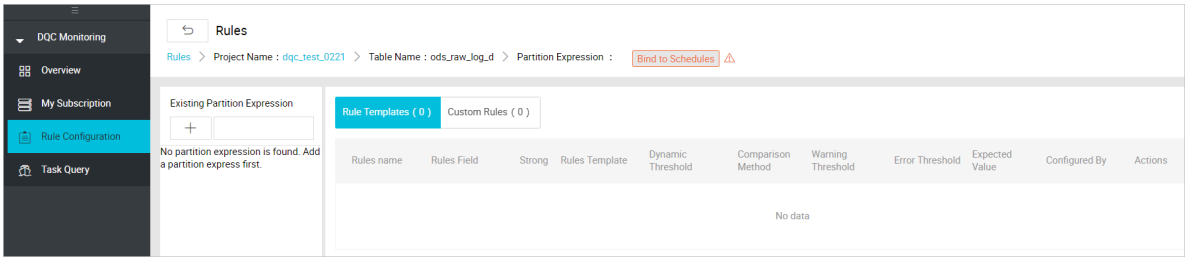
The most important thing in data quality is the configuration of table rules, so how to configure table rules is reasonable? Let's take a look at how the tables above be configured with table rules.

- 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.



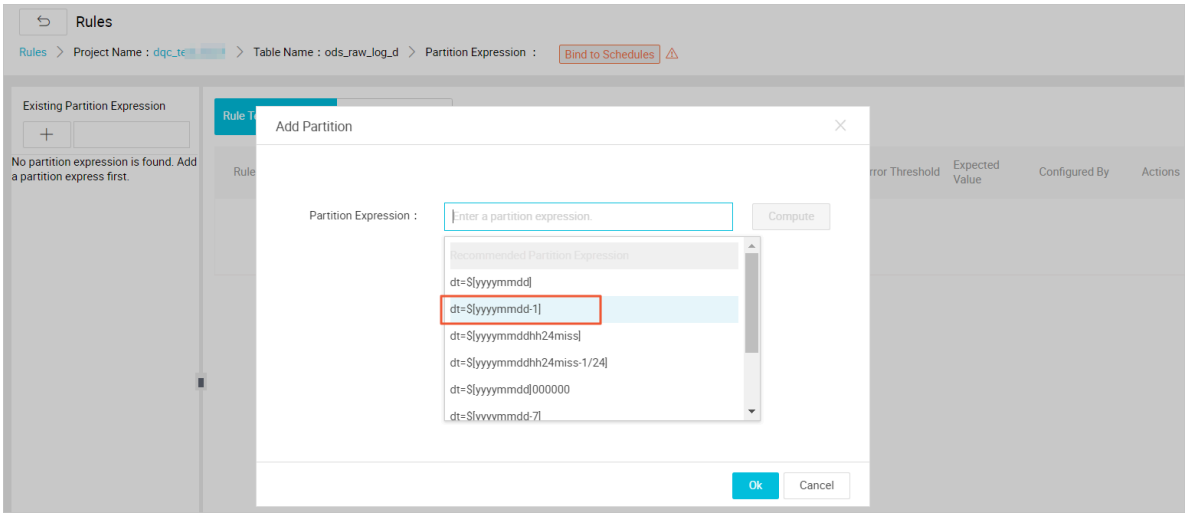

  Select the ods_raw_log_d table and click Rule Configuration to go to the following page.




  You can review the data sources for this ods_raw_log_d table. The data for ods_raw_log_d table is from OSS. 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.

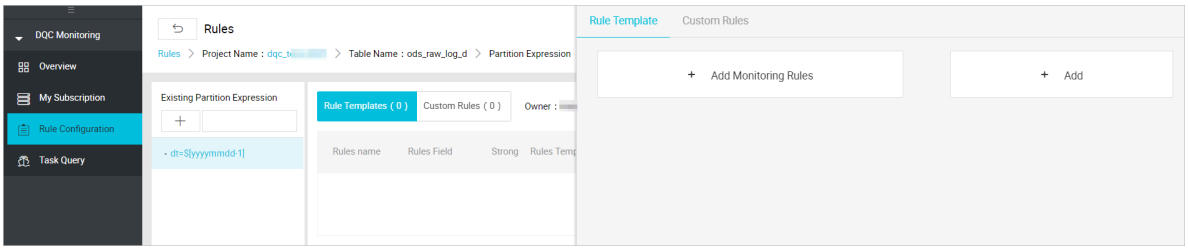



**Note:**

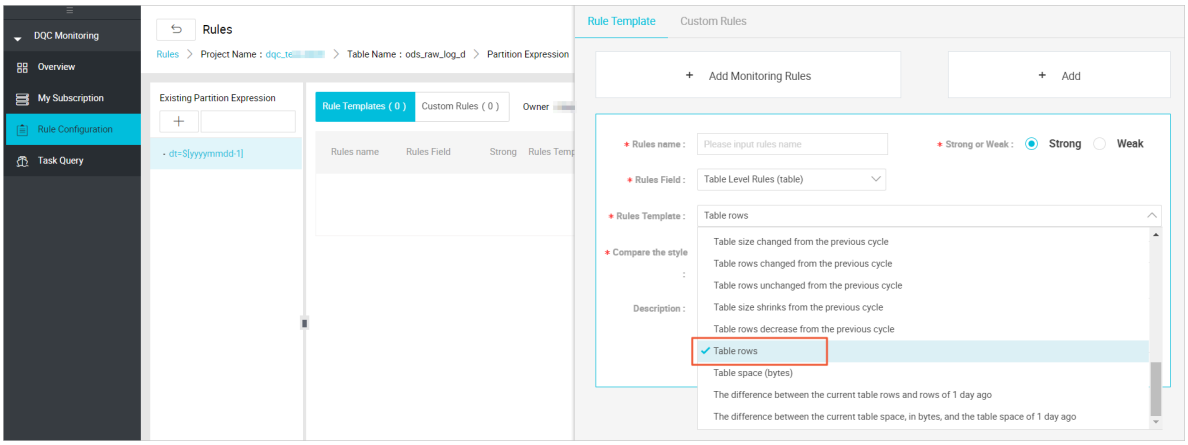
If there is no partition columns in the table, you can configure it as no partition . Depending on the real partition value, you can configure the corresponding partition expression.

After confirm, you can see the interface below and choose to Create Rule.


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




Click Add Monitoring Rules and a prompt window appears for you to configure the rule.

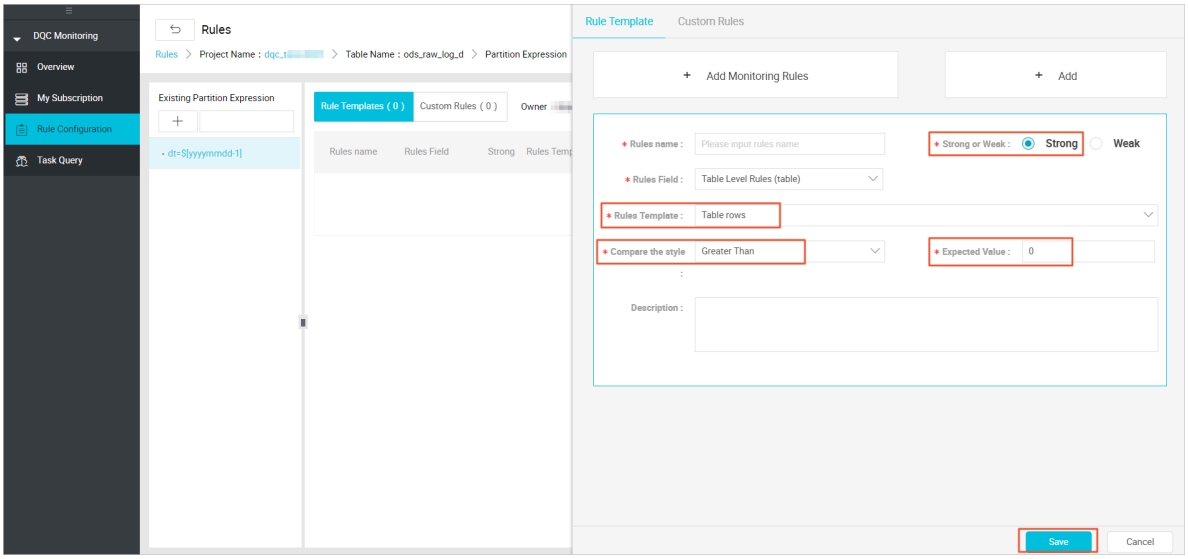



The data in this table comes from the log file that is uploaded by OSS as the source table. You need to determine whether there is data in this table partition as soon as possible. If there is no data in this table, you need to stop the subsequent tasks from running as if the source table does not have 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.**

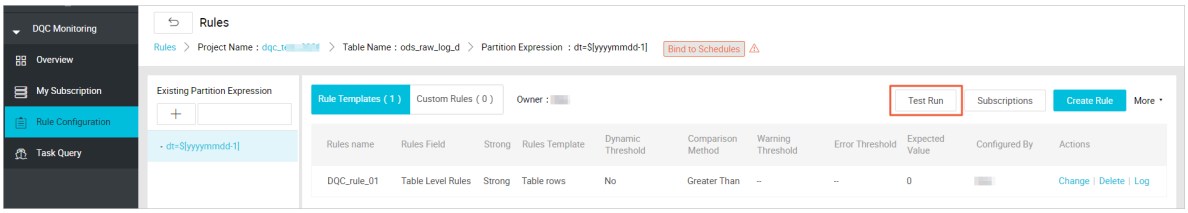



**Note:**

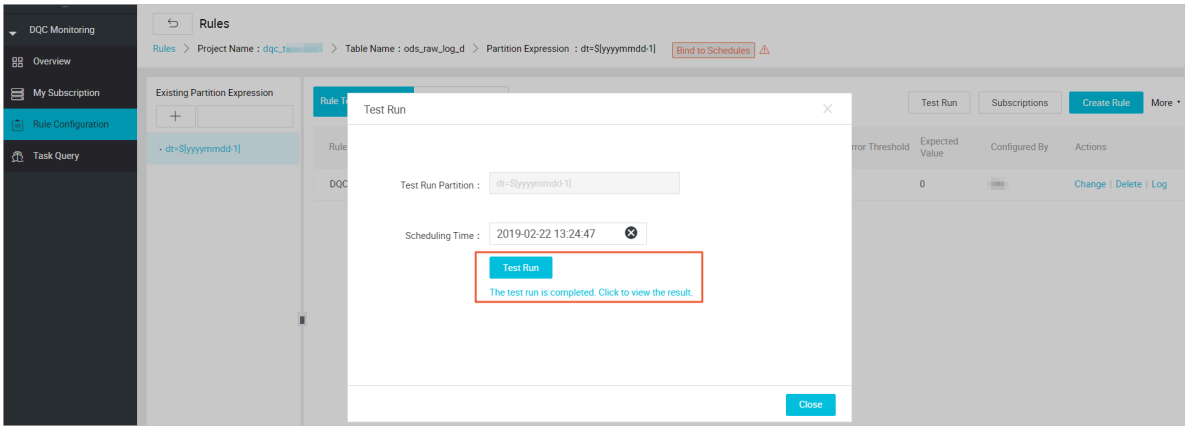
This configuration is primarily to avoid the situation that there is no data in the partition, which causes the data source for the downstream task to be empty.

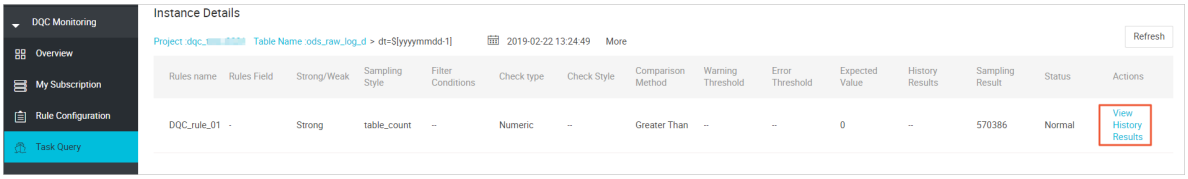
Rules test

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




When you click the Test Run button, you are prompted for a window to confirm the Scheduling Time. After a Test 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.

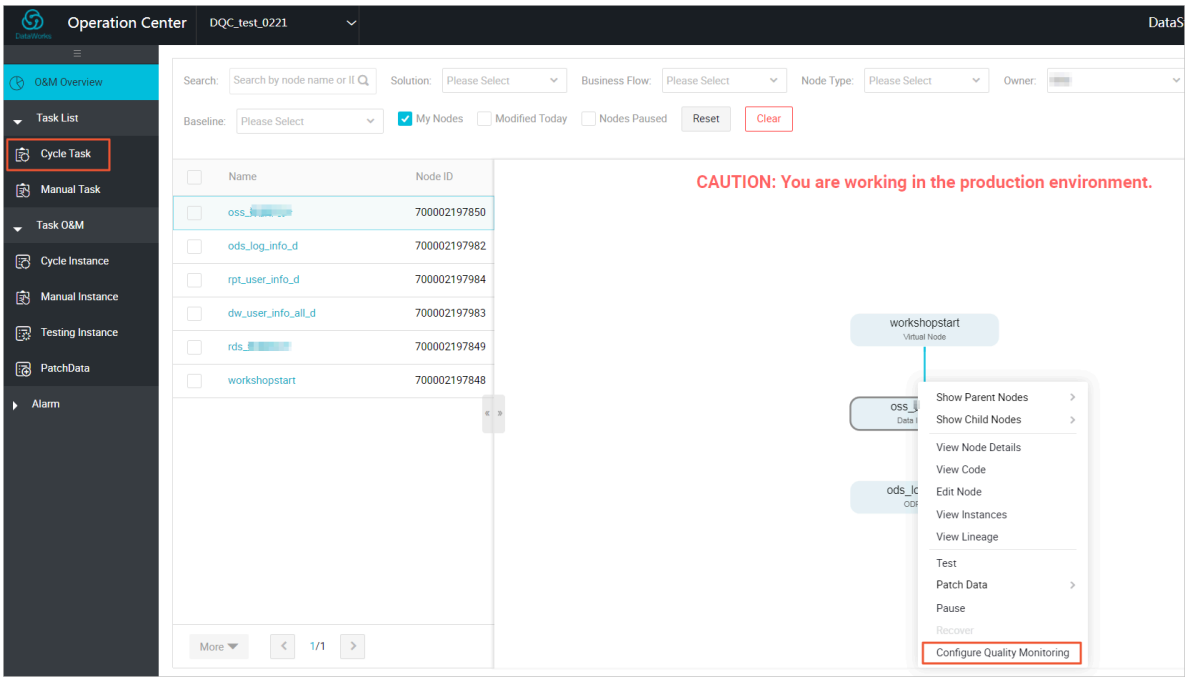
Bind to schedules

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.

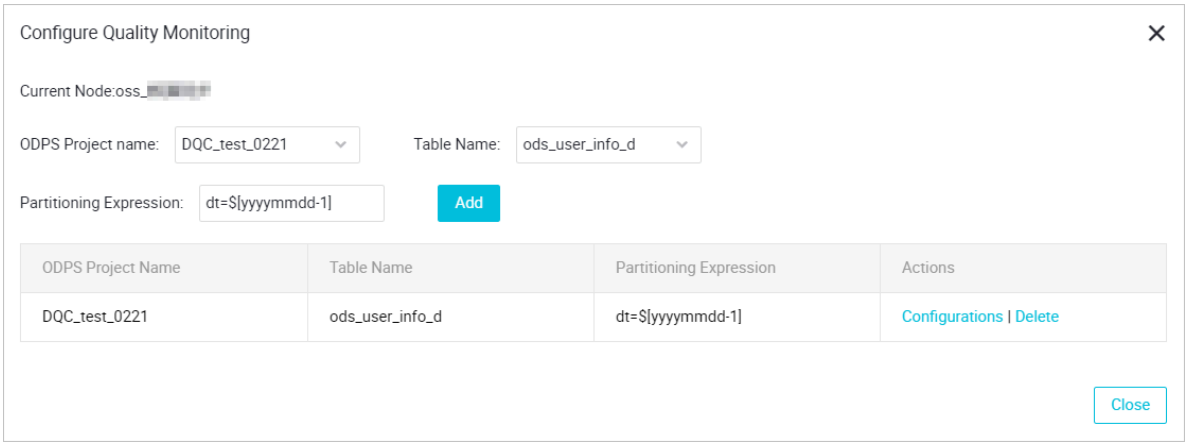
Associate table rules in operation center

In Operation Center, locate the OSS_datasync task in Cycle Task, and right-click to select Configure Quality Monitoring.

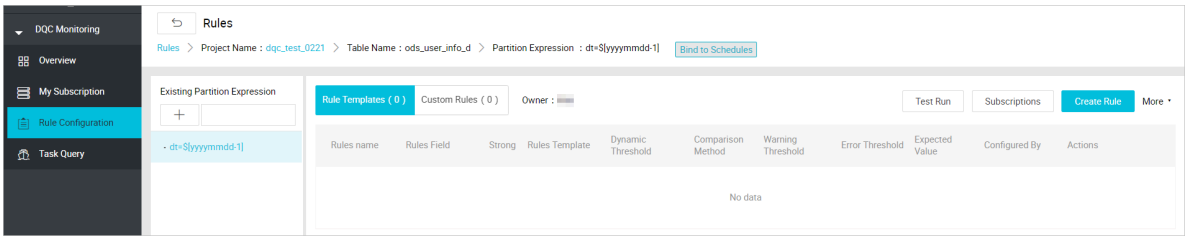



Enter the monitored table name in the burst window, as well as the partition expression. The table entered here is named as ods_user_info_d and the partition expression is dt = $ [yyyymmdd-1].

After the configuration is completed, as shown in the figure below.




Click Configurations to quickly go to the rule configuration interface.

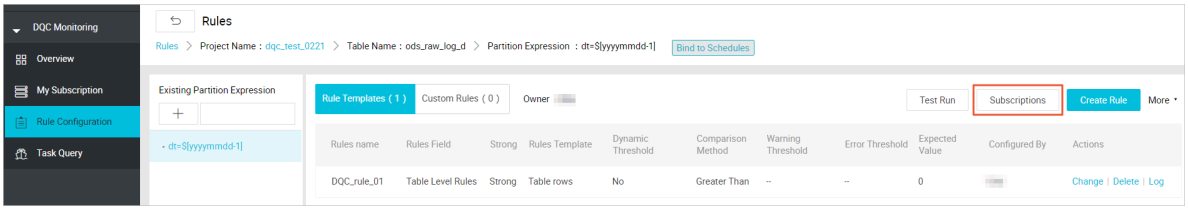


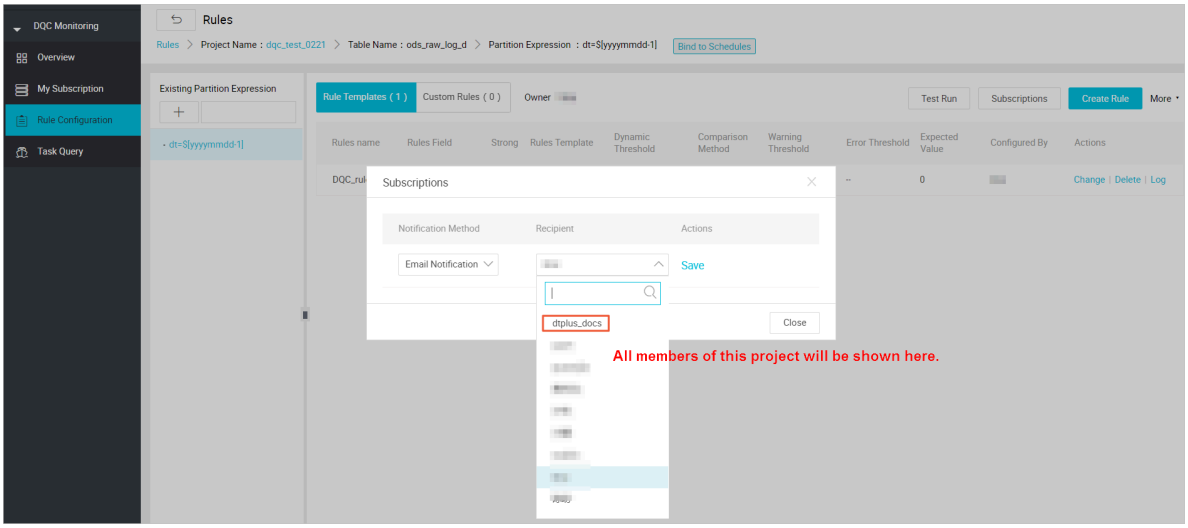

## Configure task subscriptions

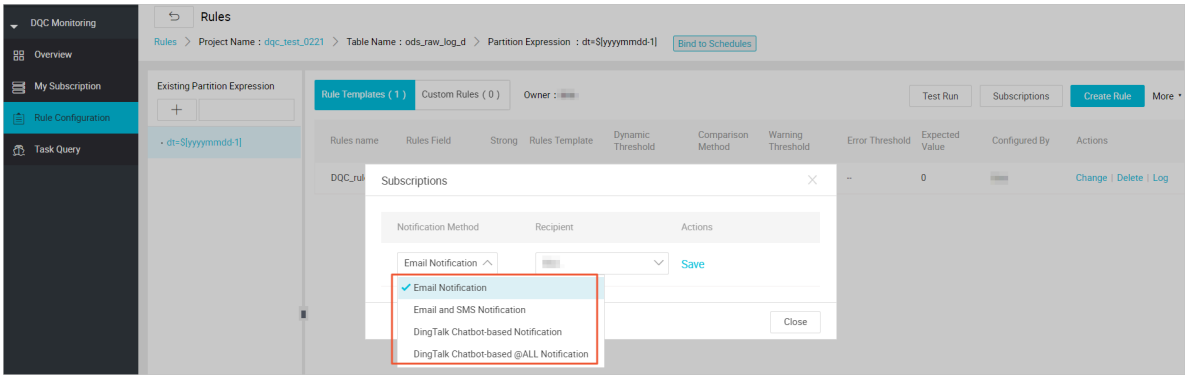
After the associated scheduling, every time the scheduling task is run, the data quality verification is triggered. Data quality supports setting up rule subscriptions , and you can set up subscriptions for important tables and their rules, set up your subscription to alert you based on the results of the data quality check. If the data

quality check results are abnormal, notifications are made based on the configured alarm policy.

Click Subscriptions to set up subscription methods. Email 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.

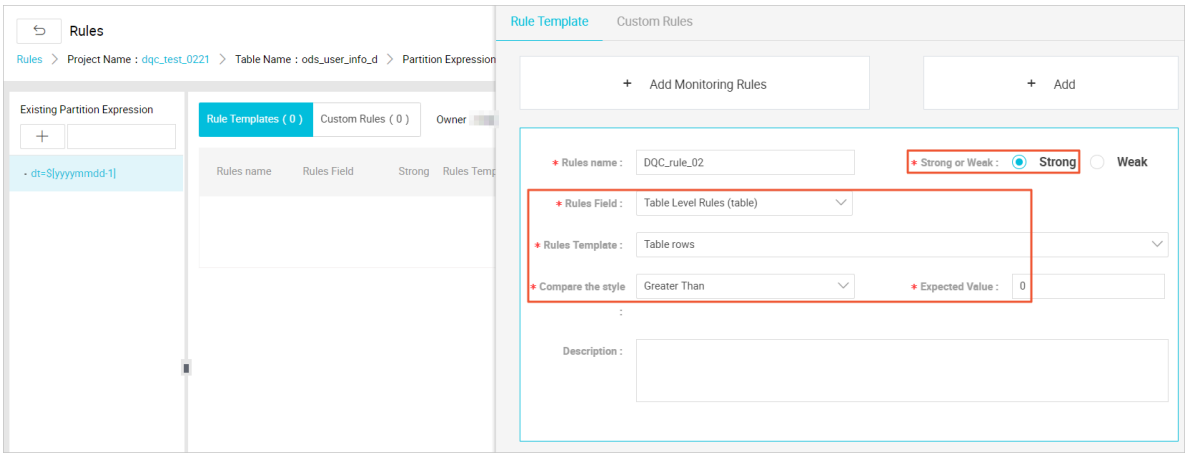

It is recommended that you subscribe to all rules so that the verification results are not notified in a timely manner.

- 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.

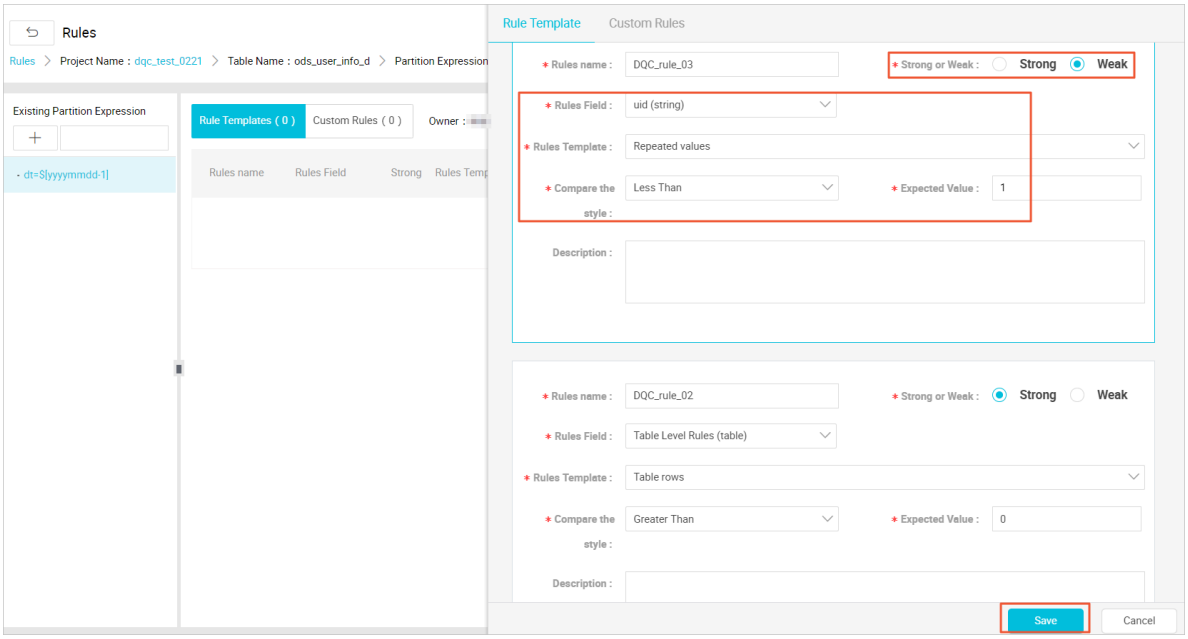
  Similarly, you need to configure a monitoring rule for a partition field first, and the monitoring time expression is: dt = $[yyyymmdd-1]. After successful configuration , you can see a successful partition configuration record in the partition expression that has been added.


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 Save button.




**Note:**

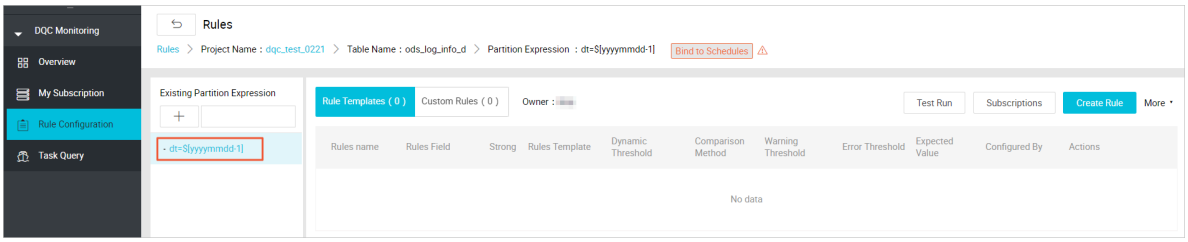
This configuration is primarily designed to avoid duplication of data which may result in contamination of downstream data.

Pay attention: don't forget to try Test Run -> Bind to Schedules -> Subscription.

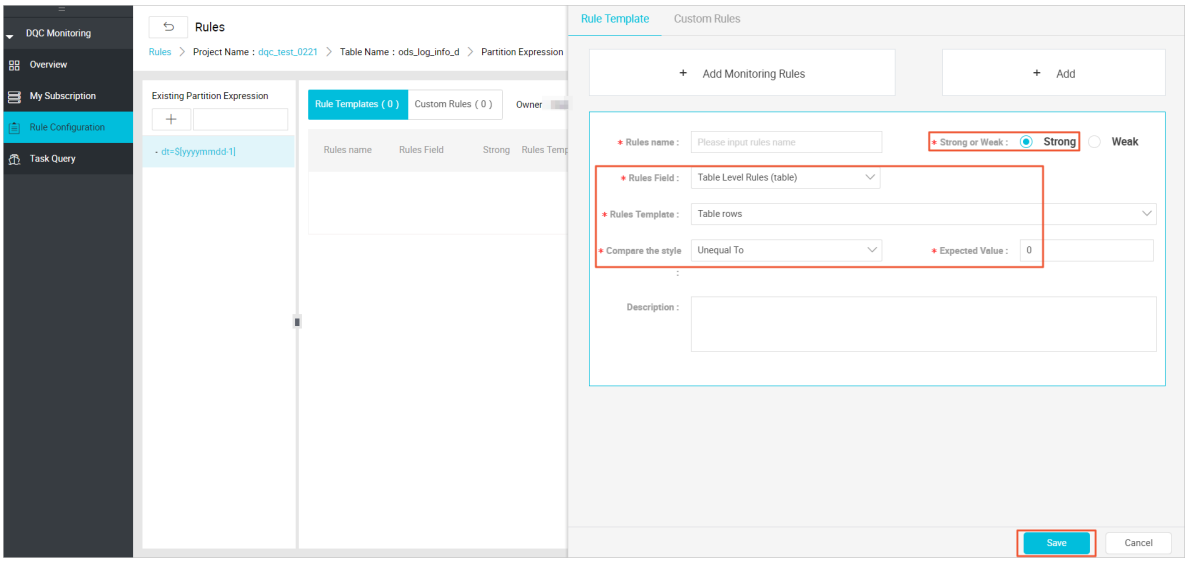
ods_log_info_d

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.



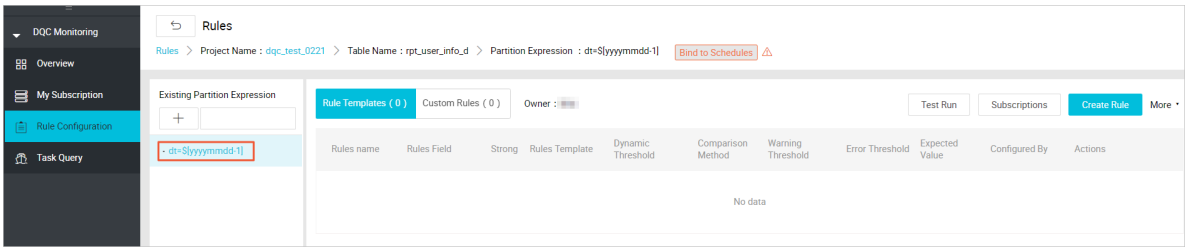

- dw_user_info_all_d

  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.

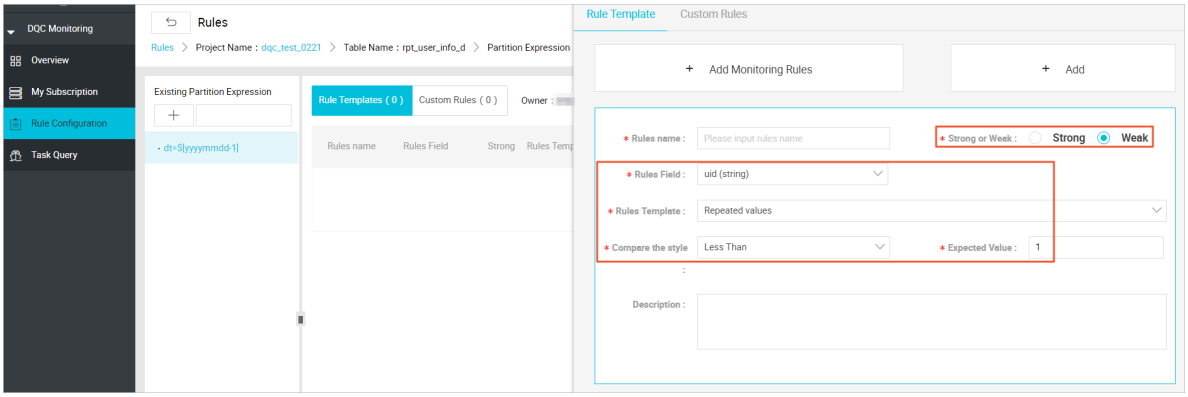
- rpt_user_info_d

  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

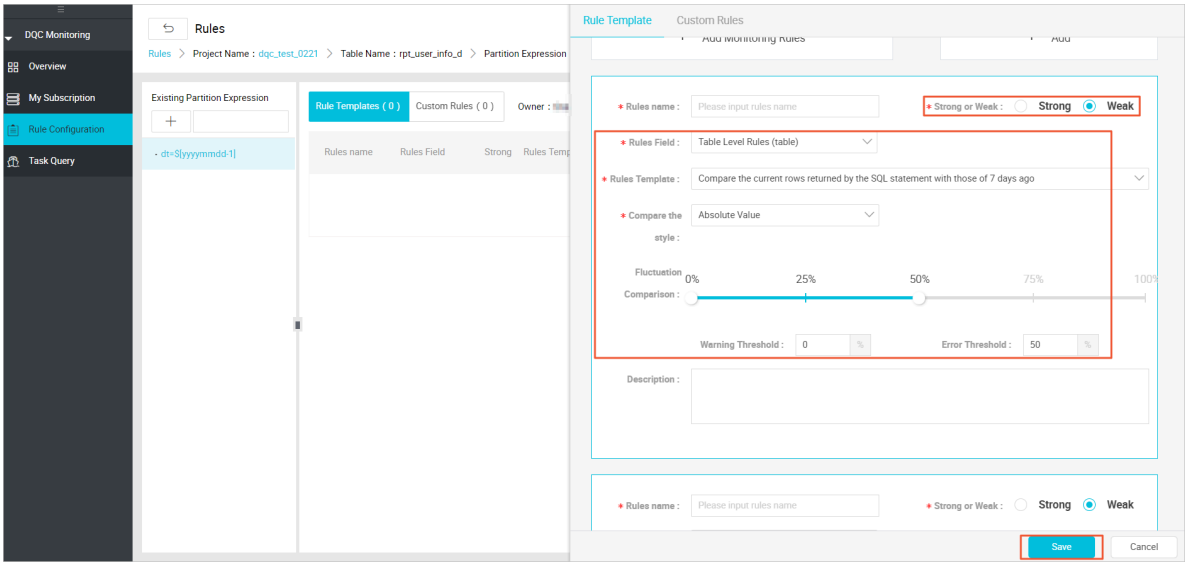
, and verify the unique values for primary keys. Partition expression for the first configuration table: dt = $[yyyymmdd-1]




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 style is set to field repeat values less than 1.




Continue to add monitoring rules.




**Note:**

**Here you monitor the number of table rows mainly to view the daily UV fluctuations in order to keep abreast of application dynamics.**

**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.**

# 2 Data migration

## 2.1 Migrate data from Hadoop to MaxCompute

**This topic describes how to use the data synchronization feature of DataWorks to migrate data from Hadoop to Alibaba Cloud MaxCompute.**

### Prepare the environment

1. **Build a Hadoop cluster.**

   **Before data migration, you must ensure that your Hadoop cluster works properly. You can use Alibaba Cloud E-MapReduce to automatically build a Hadoop cluster.**

   **The version information of E-MapReduce Hadoop is as follows:**

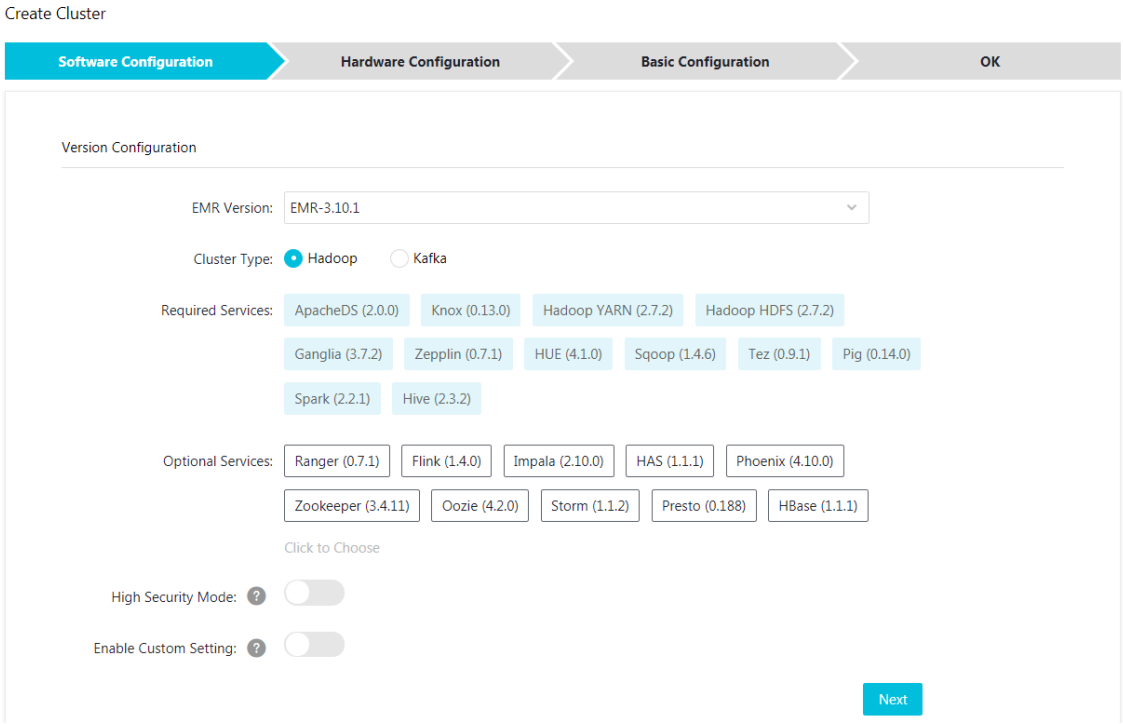
   **E-MapReduce version: EMR-3.10.1 or 3.11.0**

   **Cluster type: Hadoop**

   **Software(for EMR-3.11.0): HDFS2.7.2 / YARN2.7.2 / Hive2.3.3 / Ganglia3.7.2 / Spark2.2.1 / HUE4.1.0 / Zeppelin0.7.3 / Tez0.9.1 / Sqoop1.4.6 / Pig0.14.0 / ApacheDS2.0.0 / Knox0.13.0**

   **The network type of the Hadoop cluster is classic. The region is China East 1 ( Hangzhou). The ECS compute resource of the master instance group is configured with an Internet IP address and an intranet IP address. The high availability mode**

is set to No (a non-HA mode). The following figure shows the configuration for EMR -3.10.1.




2. MaxCompute

   For more information, see *Activate MaxCompute*.

   Activate MaxCompute and create a project. In this topic, create a project named bigdata_DOC in China East 1 (Hangzhou) and enable the related DataWorks services for this project.

## Prepare data

1. Create test data on the Hadoop cluster.

   In the E-MapReduce console, go to the Hadoop cluster page and use Notebook to create a notebook task. The table creation Hive statements in this example are as follows:

```
CREATE   TABLE   IF   NOT
EXISTS   hive_doc_g  ood_sale (

create_tim  e   timestamp ,

category   STRING ,

brand   STRING ,

buyer_id   STRING ,

trans_num   BIGINT ,
```

```
 trans_amou  nt   DOUBLE ,

 click_cnt   BIGINT

)

 PARTITIONE  D   BY  ( pt   string )  ROW   FORMAT
 DELIMITED   FIELDS   TERMINATED   BY  ','  lines   terminated   by
 '\ n '
```

Click run. The test table `hive_doc_g ood_sale` is then successfully created on the E-MapReduce Hadoop cluster.

Insert the test data. You can select data from OSS or other data sources, or manually insert a small amount of test data. The following data can be manually inserted:

```
 insert   into
 hive_doc_g  ood_sale   PARTITION ( pt  = 1  )  values (' 2018 - 08
 - 21 ',' Coat ',' Brand   A ',' lilei ', 3 , 500 . 6 , 7 ),(' 2018
 - 08 - 22 ',' Fresh   food ',' Brand   B ',' lilei ', 1 , 303 , 8
 ),(' 2018 - 08 - 22 ',' Coat ',' Brand   C ',' hanmeimei ', 2 , 510
 , 2 ),( 2018 - 08 - 22 ,' Toiletries ',' Brand   A ',' hanmeimei ',
 1 , 442 . 5 , 1 ),(' 2018 - 08 - 22 ',' Fresh   food ',' Brand   D
 ',' hanmeimei ', 2 , 234 , 3 ),(' 2018 - 08 - 23 ',' Coat ',' Brand
   B ',' jimmy ', 9 , 2000 , 7 ),(' 2018 - 08 - 23 ',' Fresh   food
 ',' Brand   A ',' jimmy ', 5 , 45 . 1 , 5 ),(' 2018 - 08 - 23 ','
 Coat ',' Brand   E ',' jimmy ', 5 , 100 . 2 , 4 ),(' 2018 - 08 - 24
 ',' Fresh   food ',' Brand   G ',' peiqi ', 10 , 5560 , 7 ),(' 2018
 - 08 - 24 ',' Sanitary   ware ',' Brand   F ',' peiqi ', 1 , 445 .
 6 , 2 ),(' 2018 - 08 - 24 ',' Coat ',' Brand   A ',' ray ', 3 , 777
 , 3 ),(' 2018 - 08 - 24 ',' Sanitary   ware ',' Brand   G ',' ray
 ', 3 , 122 , 3 ),(' 2018 - 08 - 24 ',' Coat ',' Brand   C ',' ray
 ', 1 , 62 , 7 ) ;
```

After inserting the data, you can use the `select * from hive_doc_g ood_sale where pt = 1 ;` statement to check whether the data exists in the Hadoop cluster table for migration.

2. Use DataWorks to create a destination table.

   In the DataWorks console, click the MaxCompute project, and choose Data Development > New > Create Table.

   In the displayed window, enter the following table creation SQL statements:

```
CREATE   TABLE   IF   NOT   EXISTS   hive_doc_g  ood_sale (
   create_tim  e   string ,
   category   STRING ,
   brand   STRING ,
   buyer_id   STRING ,
   trans_num   BIGINT ,
   trans_amou  nt   DOUBLE ,
   click_cnt   BIGINT
  )
```

```
    PARTITIONE  D   BY  ( pt   string ) ;
```

When creating a table, pay attention to the mapping between the Hive data type and the MaxCompute data type.

The data synchronization feature of DataWorks does not support `timestamp` data. Therefore, in the DataWorks table creation statements, create_time is set to a string value. You can also use the odpscmd Command Line (CLI) tool to create a table. For more information about how to install and configure the tool, see *Install and configure a client*. The table creation process is as follows.

```
odps@ bigdata_DOC>CREATE TABLE IF NOT EXISTS hive_doc_good_sale(
   create_time timestamp,
   category STRING,
   brand STRING,
   buyer_id STRING,
   trans_num BIGINT,
   trans_amount DOUBLE,
   click_cnt BIGINT
   )
   PARTITIONED BY (pt string) ;
                >               >               >               >
        >               >               >               >
 >
ID = 20180906110540873gev1bpim
OK
odps@ bigdata_DOC>drop table hive_doc_good_sale;
Confirm to "drop table hive_doc_good_sale;" (yes/no)? yes

ID = 20180906110825180gxh66292
```

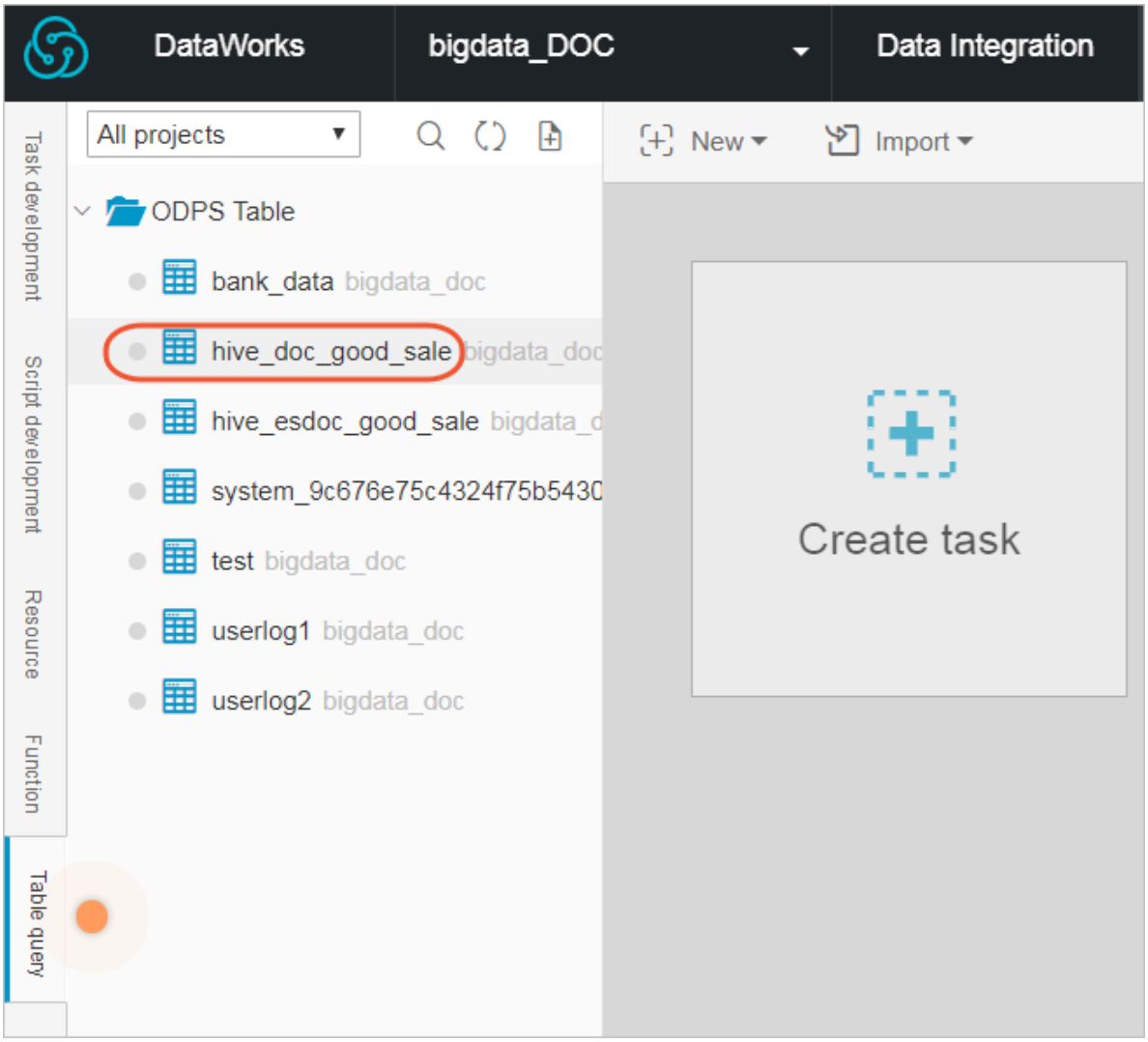
**Note:**

Considering the compatibility of the Hive and MaxCompute data types, we recommend that you run the following command on the odpscmd client:

```
  set   odps . sql . type . system . odps2 = true ; set
```

```
odps . sql . hive . compatible = true ;
```

After the table is created, you can choose Data Development > Table Query in the DataWorks console to view the table created in MaxCompute, as shown in the following figure.




### Synchronize data

1. Create a custom resource group.

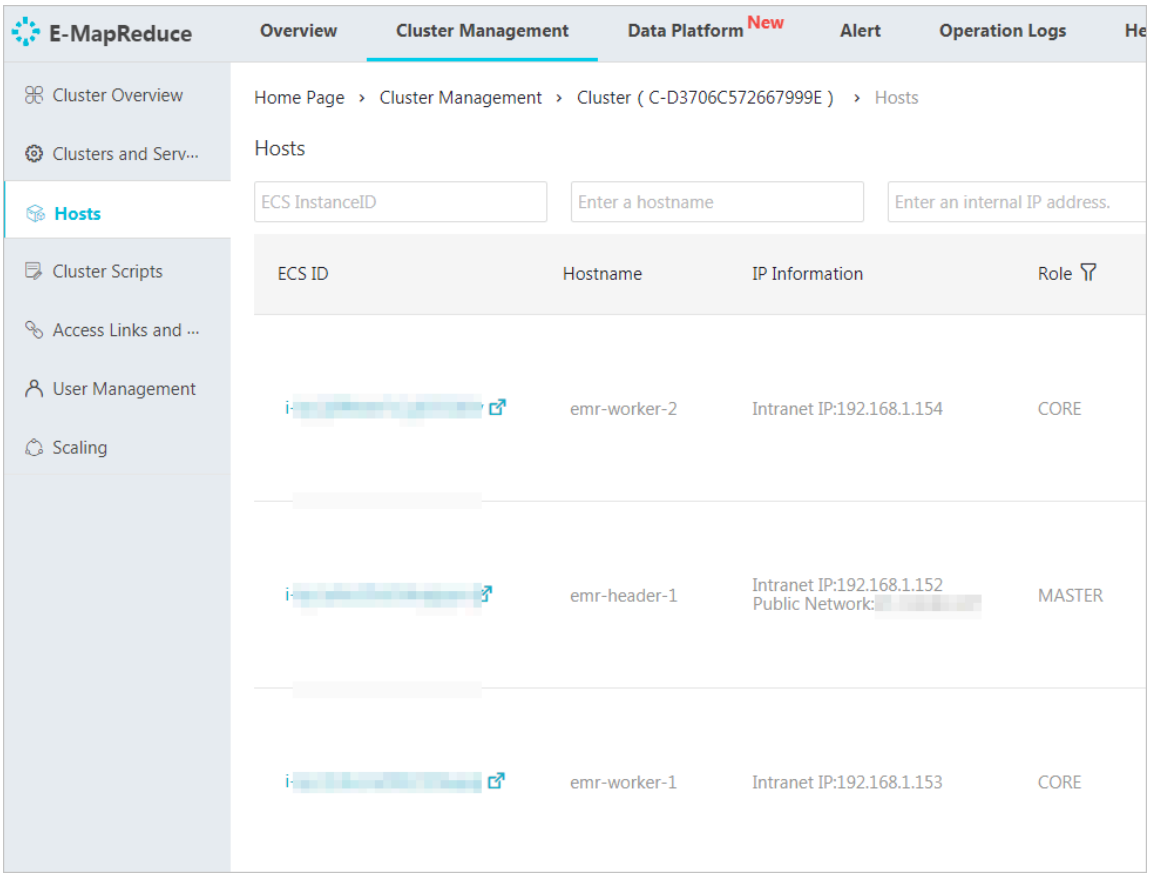
   In most cases, the network between the project data node of MaxCompute and the data node of the Hadoop cluster is not connected. You can customize a resource group to execute the synchronization task of DataWorks on the master node of the

Hadoop cluster. (In general, the network between the master node and the data node on the Hadoop cluster is connected).

a. View the data node of the Hadoop cluster.

   On the home page of the E-MapReduce console, choose Cluster Management > Cluster > Hosts. You can view the data node of the Hadoop cluster. As shown in the following figure, the host name of the master node on the E-MapReduce

Hadoop cluster (non-HA mode) is emr-header-1, and the host name of the data node is emr-worker-X.




You can also click the ECS ID of the master node, click Connect on the displayed ECS details page, and run the hadoop dfsadmin –report command to view the data node, as shown in the following figure.

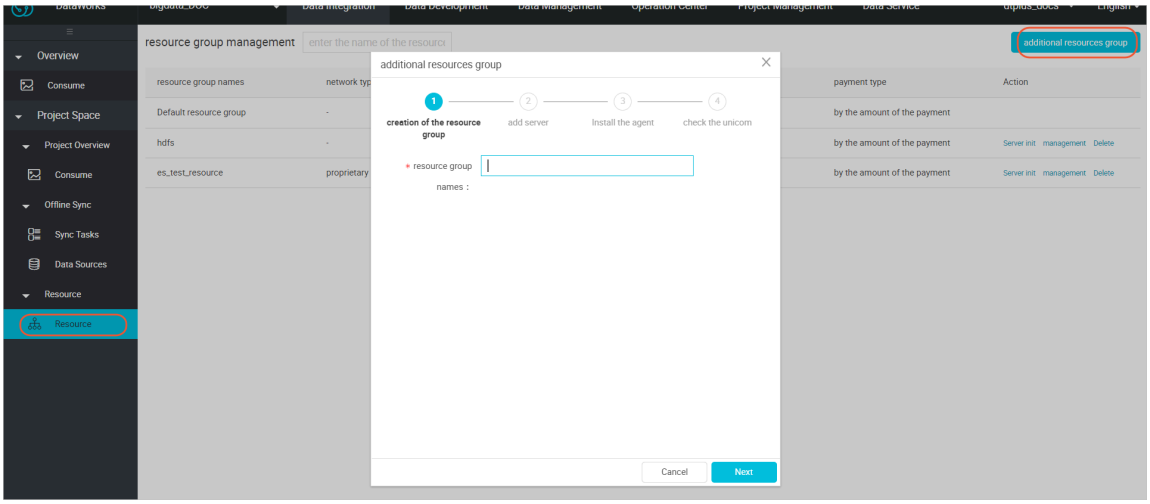
```
DFS Used%: 0.05%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0
Missing blocks (with replication factor 1): 0

-------------------------------------------------
Live datanodes (2):

Name: 10.31.122.189:50010 (emr-worker-1.cluster-74503)
Hostname: emr-worker-1.cluster-74503
Decommission Status : Normal
Configured Capacity: 333373341696 (310.48 GB)
DFS Used: 155725824 (148.51 MB)
Non DFS Used: 325541888 (310.46 MB)
DFS Remaining: 332892073984 (310.03 GB)
DFS Used%: 0.05%
DFS Remaining%: 99.86%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Thu Sep 06 19:41:01 CST 2018


Name: 10.81.78.209:50010 (emr-worker-2.cluster-74503)
Hostname: emr-worker-2.cluster-74503
Decommission Status : Normal
Configured Capacity: 333373341696 (310.48 GB)
DFS Used: 155725824 (148.51 MB)
Non DFS Used: 325451776 (310.38 MB)
DFS Remaining: 332892164096 (310.03 GB)
DFS Used%: 0.05%
DFS Remaining%: 99.86%
Configured Cache Capacity: 0 (0 B)
Cache Used: 0 (0 B)
Cache Remaining: 0 (0 B)
Cache Used%: 100.00%
Cache Remaining%: 0.00%
Xceivers: 1
Last contact: Thu Sep 06 19:41:02 CST 2018
```

As shown in the preceding figure, the data node has only an intranet address and cannot communicate with the default resource group of DataWorks. Therefore, you need to customize a resource group and set the master node to a node that executes the synchronization task of DataWorks.

b. Create a custom resource group.

In the DataWorks console, go to the Data Integration page, select Resource Group, and click New Resource Groups, as shown in the following figure.

For more information about how to customize a resource group, see *Add task resources*.




When you add a server, you need to enter the information such as the ECS UUID and machine IP address. (For a classic network, enter the sever name. For a VPC network, enter the server UUID. You can add scheduling resources only for classic networks in China East 2 (Shanghai) in DataWorks V2.0. In other regions, select the VPC network type when you add a scheduling resource group regardless of whether your network type is classic or VPC.) Set the machine IP address to the Internet IP address of the master node because the intranet IP address may be inaccessible. The ECS UUID must be connected to the master node management terminal, which can be obtained by running the dmidecode | grep UUID command, as shown in the following figure. (You can use the same method if your Hadoop cluster is not built on E-MapReduce.)


After adding the server, make sure that the network between the master node and DataWorks is connected. If you are using an ECS server, you must set the server security group. If you are using an intranet IP address for

communication, set the server security group. For more information, see *Adding security groups*.

If you are using an Internet IP address, you can directly set the Internet ingress and egress under Security Group Rules(In practical application scenarios, we recommend that you set detailed bypass rules for your data security.)

After completing the preceding steps, install the custom resource group agent as prompted. If the state is available, the custom resource group is added successfully.

If the state is unavailable, you can log on to the master node, and run the `tail - f / home / admin / alisataskn ode / logs / heartbeat . log` command to check whether the heartbeat message between DataWorks and the master node has timed out, as shown in the following figure.

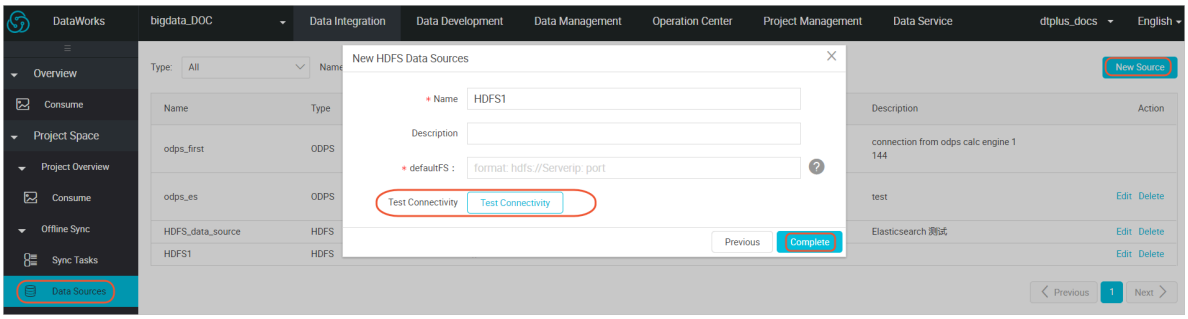
2. Create a data source.

   For more information about how to create a data source in DataWorks, see *Configuring Data Source*.

   After you create a project in DataWorks, the data source is set to `odps_first` by default. Therefore, you only need to add a Hadoop cluster data source. To do so, perform the following steps: On the Data Integration page of DataWorks, choose Data Source > New Source, and select HDFS.

   In the displayed window, enter the data source name and defaultFS. If the E-MapReduce Hadoop cluster is an HA cluster, the address is IP:8020 of hdfs://emr-header-1. If the E-MapReduce Hadoop cluster is a non-HA cluster, the address is IP:9000 of hdfs://emr-header-1. In this topic, emr-header-1 is connected to

DataWorks through the Internet. Therefore, enter the Internet IP address and open the security group.

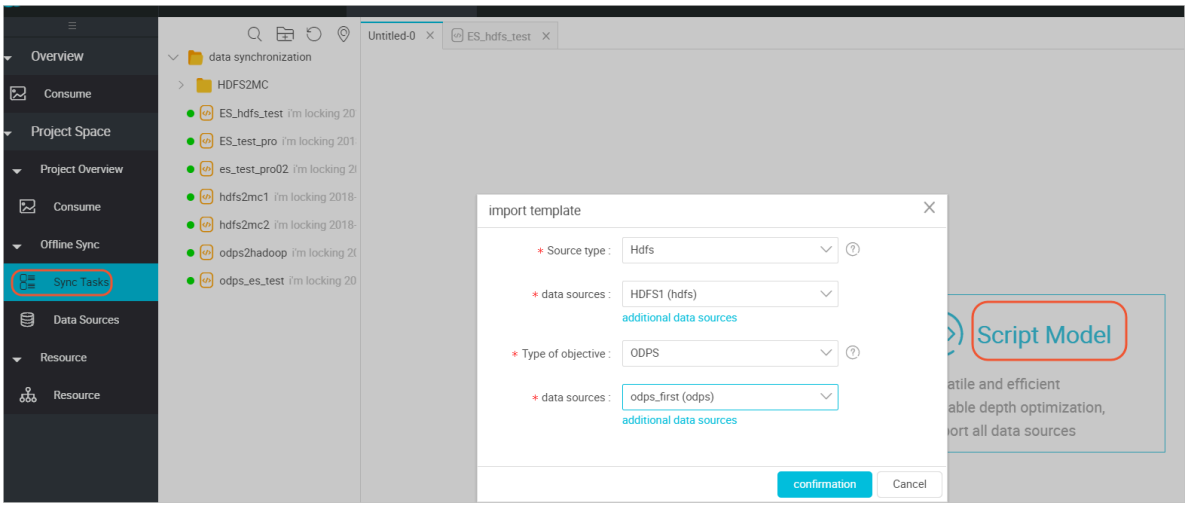



After the configuration is completed, click Test Connectivity. If Test connectivity successfully is displayed, the data source is added successfully.

> **Note:**
> If the network type of the E-MapReduce Hadoop cluster is VPC, the connectivity test is not supported.

3. Configure the data synchronization task.

   On the Data Integration page of DataWorks, click Sync Tasks and create a script mode. In the displayed window, select a data source, as shown in the following figure.




After the template is imported, the synchronization task is converted to the script mode. For more information, see *Script Mode*.

When you configure the data synchronization task script, the data types of the DataWorks synchronization task and the Hive table are as follows.

| Data type in the Hivetable | Data type in DataX / DataWorks |
| --- | --- |

| TINYINT,SMALLINT,INT,BIGINT | Long |
| --- | --- |
| FLOAT,DOUBLE,DECIMAL | Double |
| String,CHAR,VARCHAR | String |
| BOOLEAN | Boolean |
| Date,TIMESTAMP | Date |
| Binary | Binary |

The code details are as follows:

```
{
  " configurat  ion ": {
    " reader ": {
      " plugin ": " hdfs ",
      " parameter ": {
        " path ": "/ user / hive / warehouse / hive_doc_g  ood_sale
 /",
        " datasource ": " HDFS1 ",
        " column ": [
          {
            " index ":  0 ,
            " type ": " string "
          },
          {
            " index ":  1 ,
            " type ": " string "
          },
          {
            " index ":  2 ,
            " type ": " string "
          },
          {
            " index ":  3 ,
            " type ": " string "
          },
          {
            " index ":  4 ,
            " type ": " long "
          },
          {
            " index ":  5 ,
            " type ": " double "
          },
          {
            " index ":  6 ,
            " type ": " long "
          }
        ],
        " defaultFS ": " hdfs :// 121 . 199 . 11 . 138 : 9000 ",
        " fieldDelim  iter ": ",",
        " encoding ": " UTF - 8 ",
        " fileType ": " text "
      }
    },
    " writer ": {
      " plugin ": " odps ",
      " parameter ": {
        " partition ": " pt = 1 ",
```

```
        " truncate ":  false ,
        " datasource ": " odps_first ",
        " column ": [
          " create_tim  e ",
          " category ",
          " brand ",
          " buyer_id ",
          " trans_num ",
          " trans_amou  nt ",
          " click_cnt "
        ],
        " table ": " hive_doc_g  ood_sale "
      }
    },
    " setting ": {
      " errorLimit ": {
        " record ": " 1000 "
      },
      " speed ": {
        " throttle ":  false ,
        " concurrent ":  1 ,
        " mbps ": " 1 ",
        " dmu ":  1
      }
    }
  },
  " type ": " job ",
  " version ": " 1 . 0 "
}
```

The path parameter indicates the place where the data is stored in the Hadoop cluster. You can log on to the master node and run the `hdfs  dfs  - ls  / user / hive / warehouse / hive_doc_g  ood_sale` command to confirm the place. For a partition table, you do not need to specify the partitions. The data synchronization feature of DataWorks can automatically recurse to the partition path, as shown in the following figure.

```
[root@emr-header-1 logs]# hdfs dfs -ls /user/hive/warehouse/hive_doc_good_sale/
Found 1 items
drwxr-x--x   - hive hadoop          0 2018-09-03 17:46 /user/hive/warehouse/hive_doc_good_sale/pt=1
```

After the configuration is completed, click Run. If a message is displayed indicating that the task is executed successfully, the synchronization task is completed. If a message is displayed indicating that the task failed to be executed, copy the logs for further troubleshooting.

### Verify the results

In the DataWorks console, choose Data Development > Table Query and select the `hive_doc_g  ood_sal e` table. You can check whether the Hive data has been synchronized to MaxCompute. You can also create a table query task, enter the

`select * FROM hive_doc_good_sale where pt = 1;` script in the task, and click Run to query the results.

You can also enter `select * FROM hive_doc_good_sale where pt = 1;` in the odpscmd CLI tool to query the table results.

### Migrate data from MaxCompute to Hadoop

To migrate data from MaxCompute to Hadoop, perform the preceding steps but exchange the reader and writer objects in the synchronization script. The following is an example:

```
{
  " configurat ion ": {
    " reader ": {
      " plugin ": " odps ",
      " parameter ": {
      " partition ": " pt = 1 ",
      " isCompress ": false ,
      " datasource ": " odps_first ",
      " column ": [
        " create_tim e ",
        " category ",
        " brand ",
      " buyer_id ",
      " trans_num ",
      " trans_amou nt ",
      " click_cnt "
    ],
    " table ": " hive_doc_good sale "
    }
  },
  " writer ": {
    " plugin ": " hdfs ",
    " parameter ": {
    " path ": "/ user / hive / warehouse / hive_doc_good sale ",
    " fileName ": " pt = 1 ",
    " datasource ": " HDFS_data_ source ",
    " column ": [
      {
        " name ": " create_tim e ",
        " type ": " string "
      },
      {
        " name ": " category ",
        " type ": " string "
      },
      {
        " name ": " brand ",
        " type ": " string "
      },
      {
        " name ": " buyer_id ",
        " type ": " string "
      },
      {
        " name ": " trans_num ",
```

```
        " type ": " BIGINT "
      },
      {
        " name ": " trans_amou  nt ",
        " type ": " DOUBLE "
      },
      {
        " name ": " click_cnt ",
        " type ": " BIGINT "
      }
    ],
    " defaultFS ": " hdfs :// 47 . 99 . 162 . 100 : 9000 ",
    " writeMode ": " append ",
    " fieldDelim  iter ": ",",
    " encoding ": " UTF - 8 ",
    " fileType ": " text "
    }
  },
  " setting ": {
    " errorLimit ": {
      " record ": " 1000 "
  },
  " speed ": {
    " throttle ":  false ,
    " concurrent ":  1 ,
    " mbps ": " 1 ",
    " dmu ":  1
  }
  }
},
" type ": " job ",
" version ": " 1 . 0 "
}
```

Before executing the preceding synchronization task, you must set the Hadoop cluster. For more information, see *Configure HDFS Writer*. After executing the synchronization task, you need to manually copy the synchronized files.

## 2.2 Migrate JSON data from OSS to MaxCompute

This topic describes how to use the data integration feature of DataWorks to migrate JSON data from OSS to MaxCompute and use the built-in string function GET_JSON_OBJECT of MaxCompute to extract JSON information.

### Preparations

- Upload data to OSS.

  Convert your JSON file to a TXT file and upload it to OSS. The following is a JSON file example:

```
{
    " store ": {
        " book ": [
```

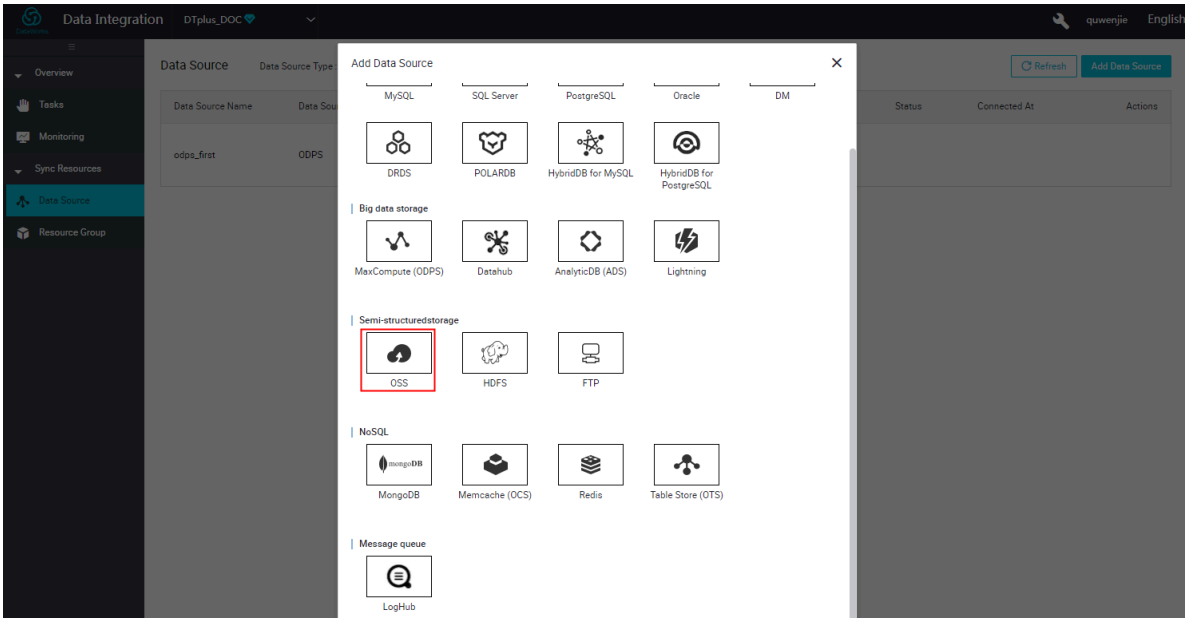
```
            {
               " category ": " reference ",
               " author ": " Nigel   Rees ",
               " title ": " Sayings   of   the   Century ",
               " price ":  8 . 95
            },
            {
               " category ": " fiction ",
               " author ": " Evelyn   Waugh ",
               " title ": " Sword   of   Honour ",
               " price ":  12 . 99
            },
            {
                " category ": " fiction ",
                " author ": " J .  R .  R .  Tolkien ",
                " title ": " The   Lord   of   the   Rings ",
                " isbn ": " 0 - 395 - 19395 - 8 ",
                " price ":  22 . 99
            }
         ],
         " bicycle ": {
             " color ": " red ",
             " price ":  19 . 95
         }
    },
    " expensive ":  10
}
```

**Upload the** *applog . txt* **file to OSS. In this example, the OSS bucket is located in China (Shanghai).**

## Use DataWorks to migrate JSON data from OSS to MaxCompute

- 1. Add an OSS data source.

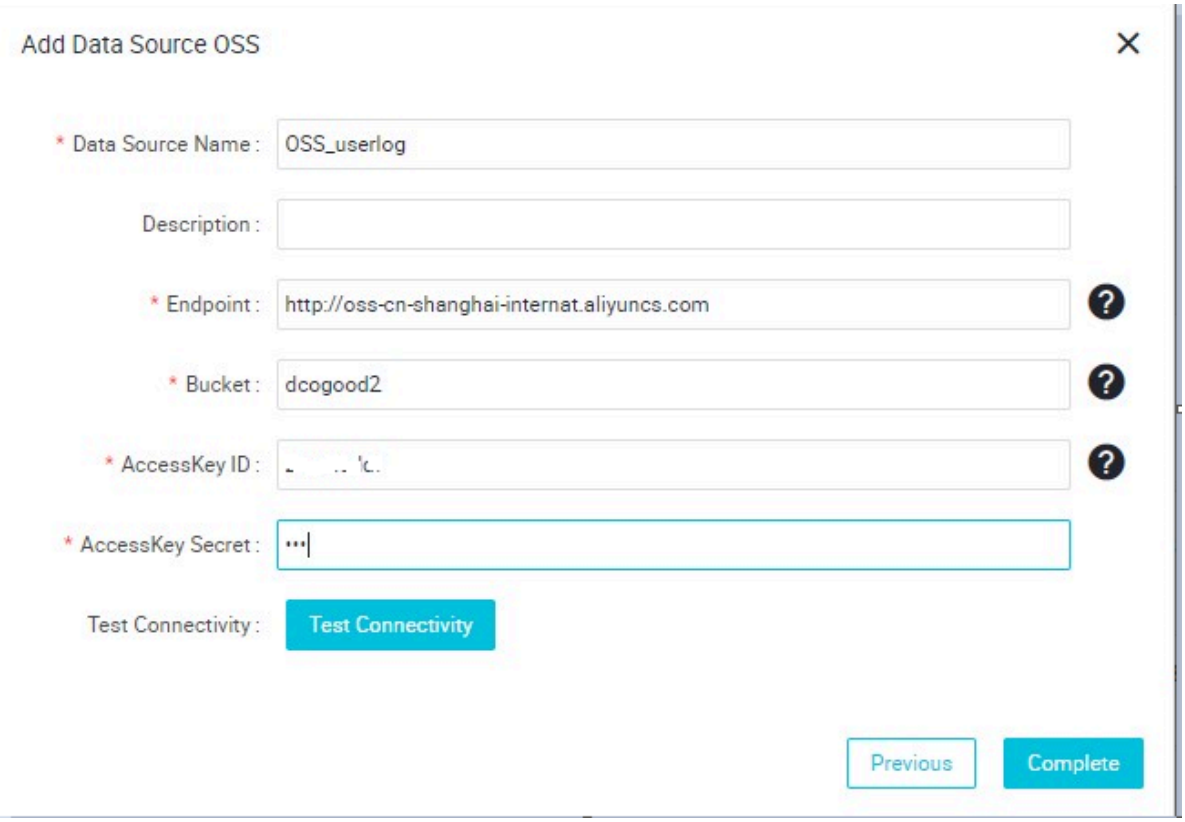
  In the DataWorks console, go to the *Data Integration* page and add an OSS data source. For more information, see *Configure OSS data source*.




  The parameters are shown in the following figure. Click Complete after the connectivity test is successful. The endpoints in this topic include `http://oss-cn-shanghai.aliyuncs.com` and `http://oss-cn-shanghai-internal.aliyuncs.com`.

  **Note:**

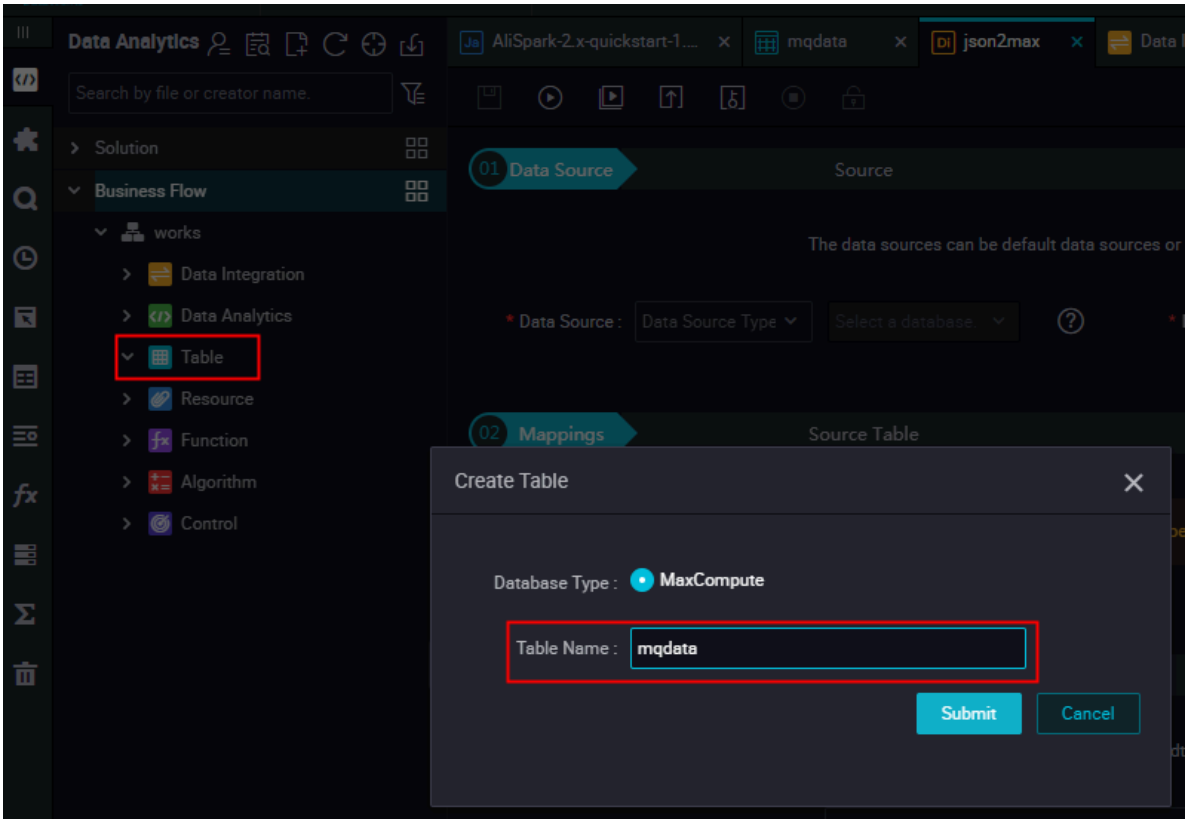
**Because the OSS and DataWorks projects are located in the same region, the intranet endpoint `http :// oss - cn - shanghai - internal . aliyuncs . com` is used.**



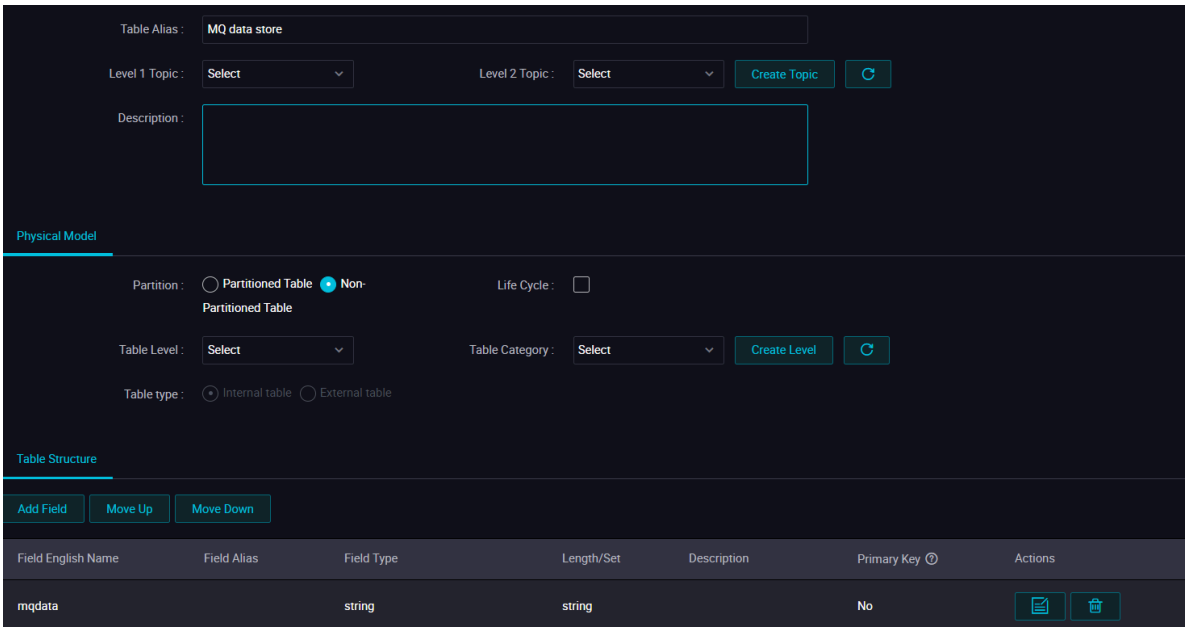

- **2. Create a data synchronization task.**

  **In the DataWorks console, create a data synchronization node. For more information, see *Configure OSS Reader*. At the same time, create a table named**

mqdata in DataWorks to store the JSON data. For more information, see *Create a table*.



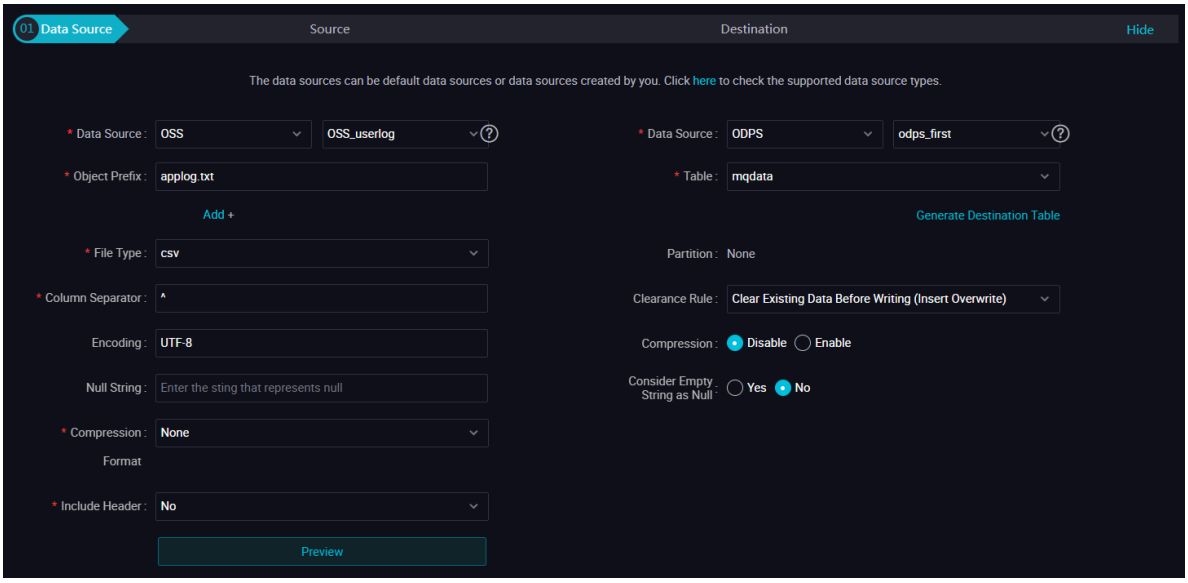

You can set the table parameters on the graphical interface. The mqdata table has only one column, which is named MQ data. The data type is string.




- 3. Set the parameters.

  After creating a table, you can set the data synchronization task parameters on the graphical interface, as shown in the following figure. First, set the destination data

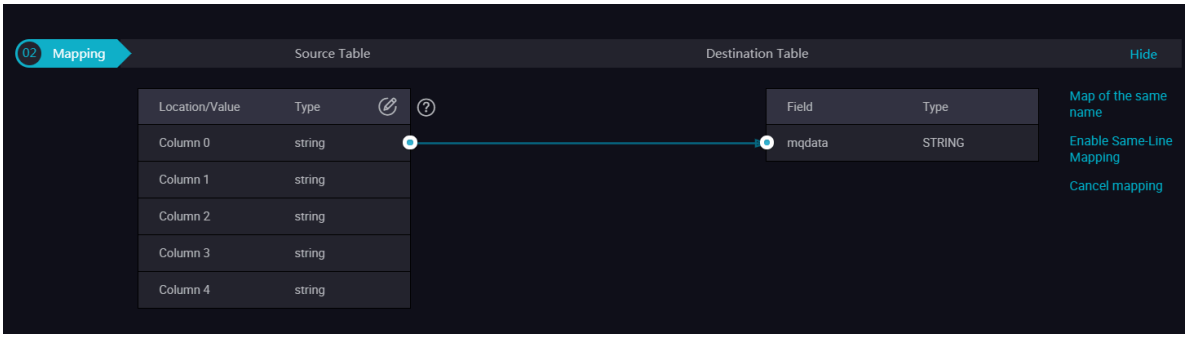
source to odps_first and the destination table to mqdata. Then, set the original data source to OSS and enter the file path and name as the object prefix.




**Note:**

You can set the column delimiter to caret (^) or any other character that is not contained in the TXT file. DataWorks supports multiple column delimiters for the TXT data sources in OSS. Therefore, you can use characters such as %&%#^$$^% to separate the data into a column.

Select Enable Same Line Mapping.




Click the script switching button in the upper-left corner to switch to the script mode. Set fileFormat to `" fileFormat ":" binary "`. The following is an example of the code in script mode:

```
{
    " type ": " job ",
    " steps ": [
        {
            " stepType ": " oss ",
            " parameter ": {
                " fieldDelim  iterOrigin ": "^",
```

```
                " nullFormat ": "",
                " compress ": "",
                " datasource ": " OSS_userlo  g ",
                " column ": [
                    {
                        " name ":  0 ,
                        " type ": " string ",
                        " index ":  0
                    }
                ],
                " skipHeader ": " false ",
                " encoding ": " UTF - 8 ",
                " fieldDelim  iter ": "^",
                " fileFormat ": " binary ",
                " object ": [
                    " applog . txt "
                ]
            },
            " name ": " Reader ",
            " category ": " reader "
        },
        {
            " stepType ": " odps ",
            " parameter ": {
                " partition ": "",
                " isCompress ":  false ,
                " truncate ":  true ,
                " datasource ": " odps_first ",
                " column ": [
                    " mqdata "
                ],
                " emptyAsNul  l ":  false ,
                " table ": " mqdata "
            },
            " name ": " Writer ",
            " category ": " writer "
        }
    ],
    " version ": " 2 . 0 ",
    " order ": {
        " hops ": [
            {
                " from ": " Reader ",
                " to ": " Writer "
            }
        ]
    },
    " setting ": {
        " errorLimit ": {
            " record ": ""
        },
        " speed ": {
            " concurrent ":  2 ,
            " throttle ":  false ,
            " dmu ":  1
        }
    }
}
```

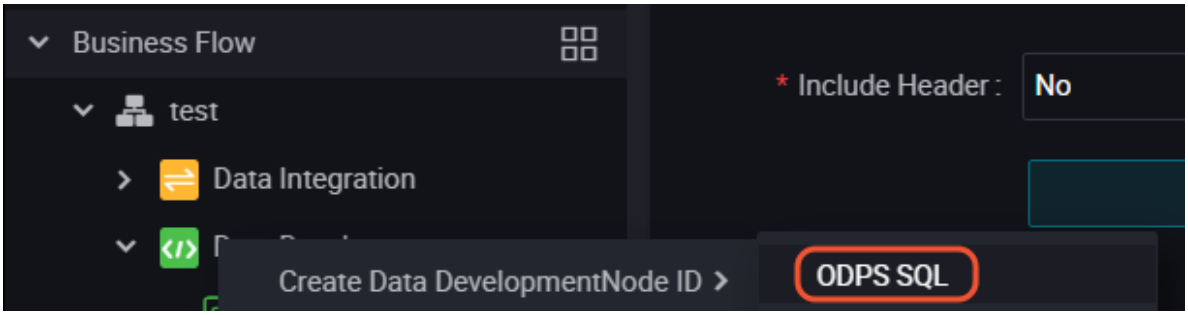
**Note:**

In this step, after the JSON file is synchronized from OSS to MaxCompute, data in the file is saved in the same row. That is, data in the JSON file shares the same field. You can use the default values for other parameters.

After completing the preceding settings, click run.

### Verify the result

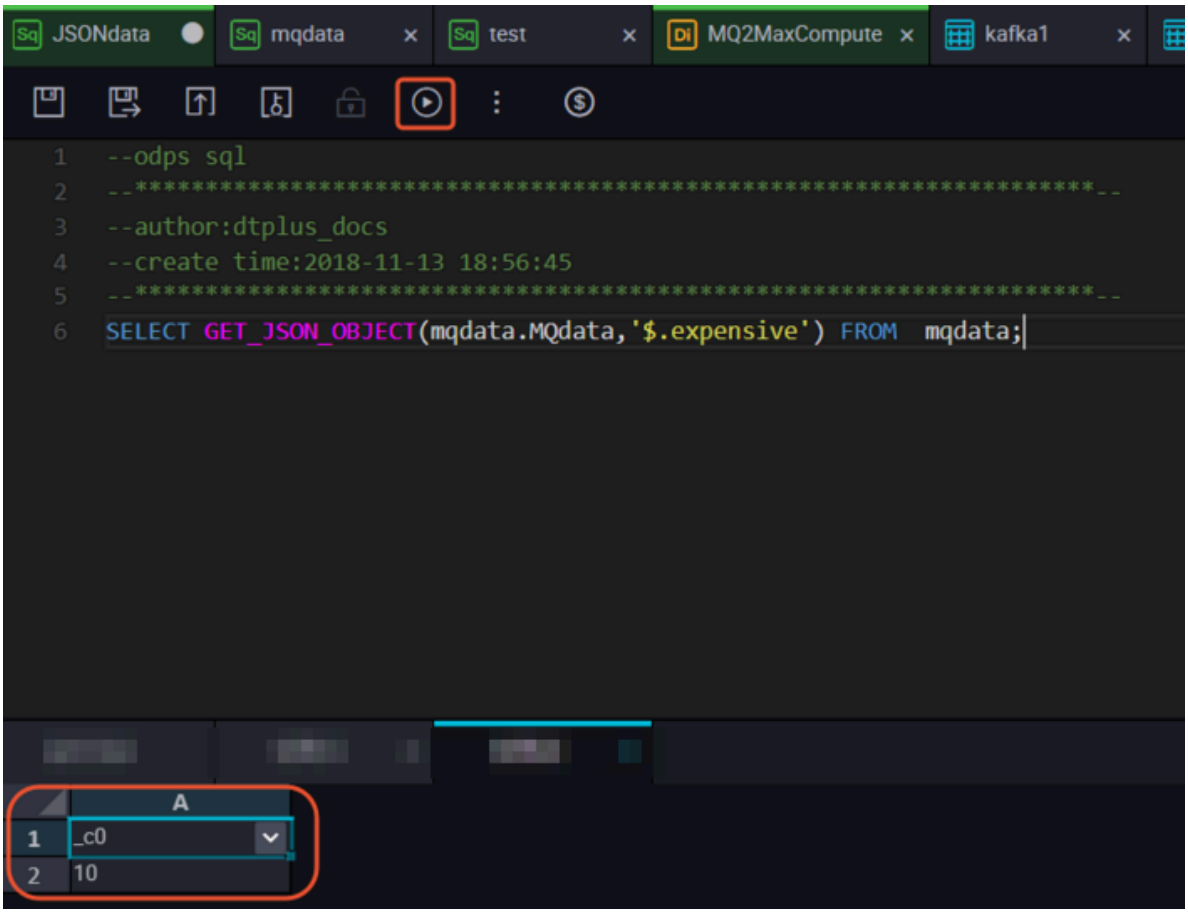
1. Create an ODPS SQL node in your *Business Flow*.




2. Enter the `SELECT * from mqdata ;` statement to view the data in the mqdata table.

Note:
You can also run the `SELECT * from mqdata ;` command on the *MaxCompute client* to view the data and perform subsequent steps.

3. Verify that the data imported to the table is correct and use `SELECT GET_JSON_OBJECT(mqdata.MQdata,'$.expensive') FROM mqdata;` to obtain the value of `expensive` in the JSON file.




### Additional information

To verify the result, you can also use the built-in string function *GET_JSON_OBJECT* in MaxCompute to obtain the JSON data as needed.

## 2.3 Migrate JSON data from MongoDB to MaxCompute

This topic describes how to use the data integration feature of DataWorks to extract JSON fields from MongoDB to MaxCompute.

### Preparations

1. Prepare an account.

   Create a user in the database in advance to add data sources in DataWorks. In this example, you can run the `db . createUser ({ user :" bookuser ", pwd :"`

123456 ", roles :[" root "]}) command to create a user named bookuser . The password of the user is 123456 , and the permission is root .

2. Prepare data.

   Upload data to your MongoDB. In this example, Alibaba Cloud ApsaraDB for MongoDB is used. The network type is VPC. (An Internet IP address is required for MongoDB to communicate with the default resource group of DataWorks.) The test data is as follows:

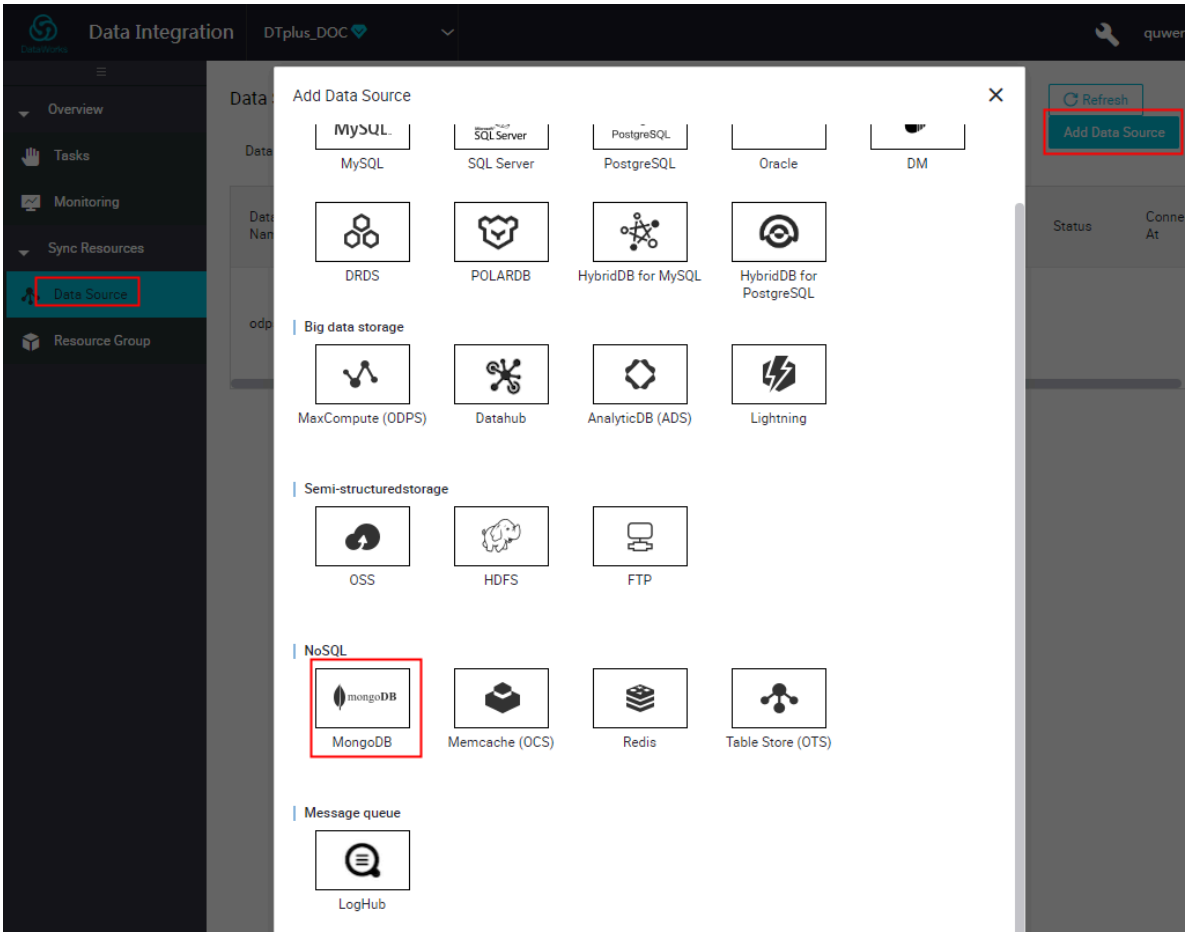
```
{
    " store ": {
        " book ": [
            {
               " category ": " reference ",
               " author ": " Nigel   Rees ",
               " title ": " Sayings   of   the   Century ",
               " price ":  8 . 95
            },
            {
               " category ": " fiction ",
               " author ": " Evelyn   Waugh ",
               " title ": " Sword   of   Honour ",
               " price ":  12 . 99
            },
            {
                " category ": " fiction ",
                " author ": " J .  R .  R .  Tolkien ",
                " title ": " The   Lord   of   the   Rings ",
                " isbn ": " 0 - 395 - 19395 - 8 ",
                " price ":  22 . 99
            }
        ],
        " bicycle ": {
            " color ": " red ",
            " price ":  19 . 95
        }
    },
    " expensive ":  10
}
```

   Log on to the DMS console of MongoDB. In this example, the database name is admin and the collection is userlog . You can run the db . userlog . find (). limit ( 10 ) command in the query window to view the uploaded data.

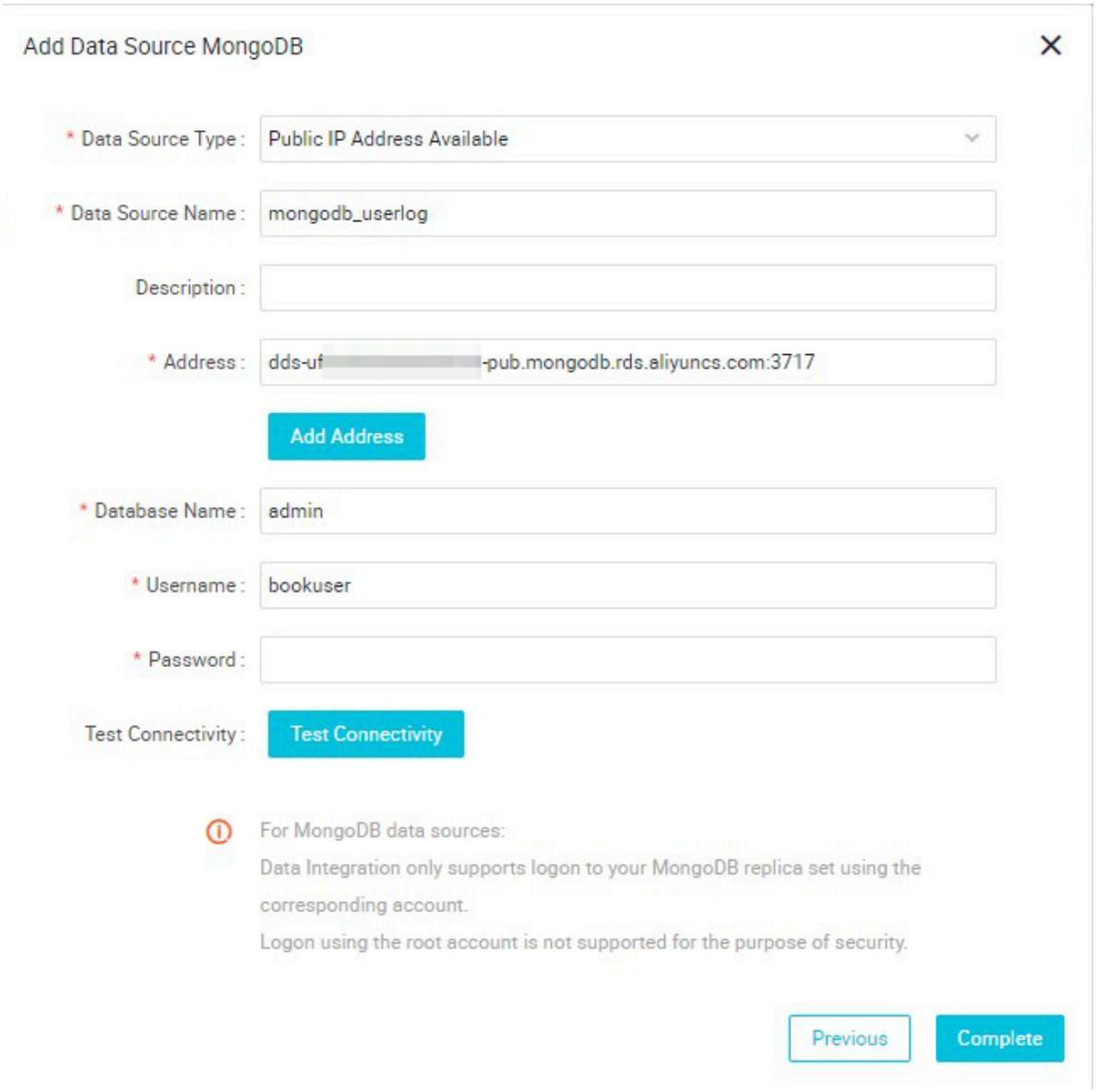
## Use DataWorks to extract data to MaxCompute

- 1. Add a MongoDB data source.

  In the DataWorks console, go to the *Data Integration* page and add a *MongoDB* data source.



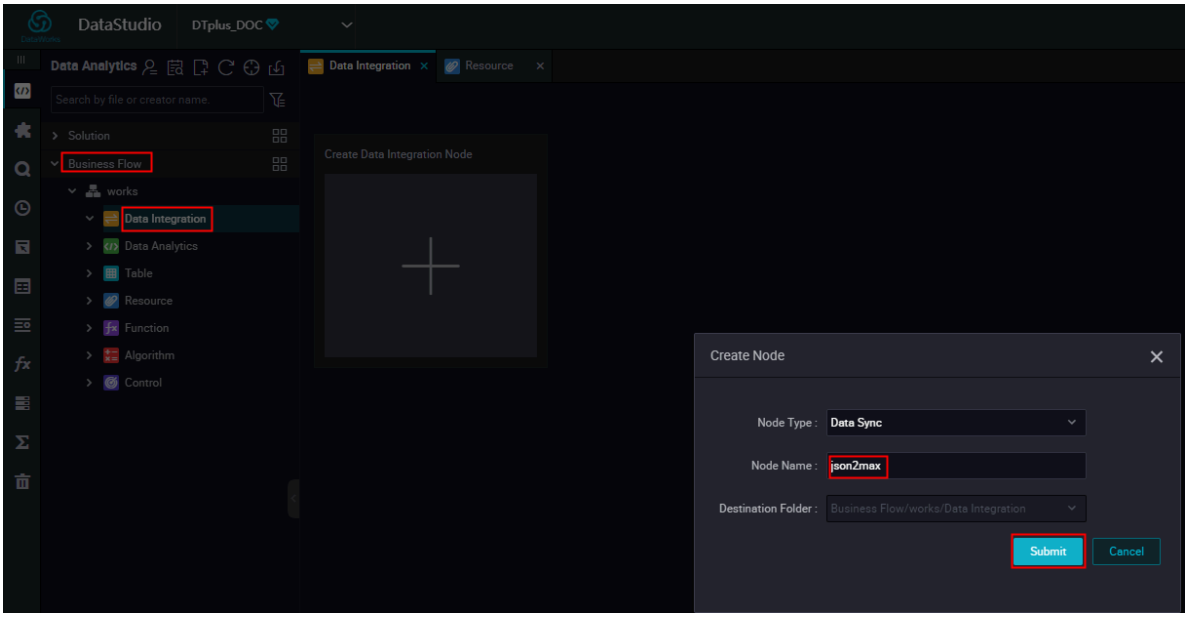

  The parameters are shown in the following figure. Click Complete after the connectivity test is successful. In this example, the network type of MongoDB is VPC. Therefore, the Data Source Type must be set to Public IP Address Available.



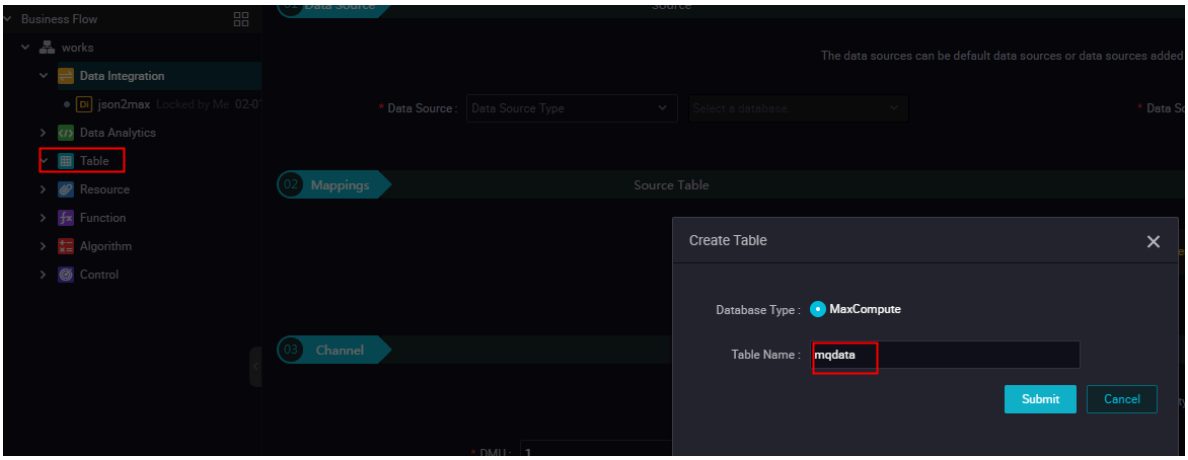

**To obtain the IP address and the port number, log on to the and click the target instance. Example parameters are shown in the following figure.**

- 2. Create a data synchronization task.

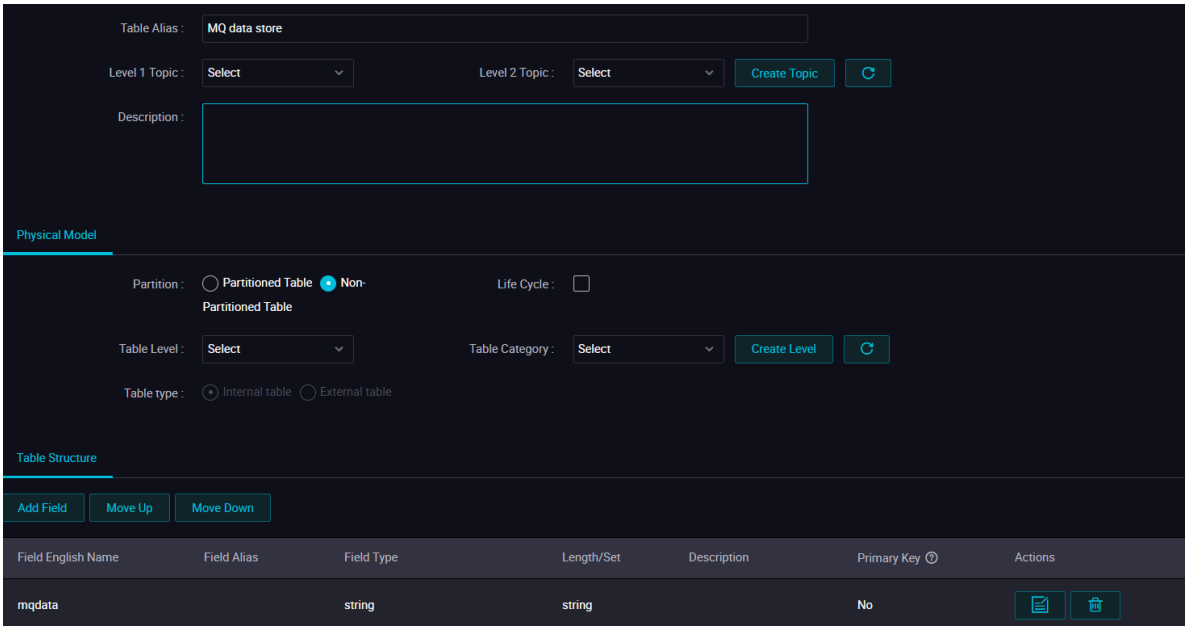
  In the DataWorks console, create a data synchronization node. For more information, see *Configure OSS Reader*.




At the same time, create a table named mqdata in DataWorks to store JSON data. For more information, see *Create a table*.




You can set the table parameters on the graphical interface. The mqdata table has only one column, which is named MQ data. The data type is string.




- 3. Set the parameters.

  After creating a table, you can set the data synchronization task parameters on the graphical interface. First, set the destination data source to odps_first and the destination table to mqdata. Then, set the original data source to MongoDB and select mongodb_userlog. After completing the preceding settings, click Switch to script mode. The following is an example of the code in script mode:

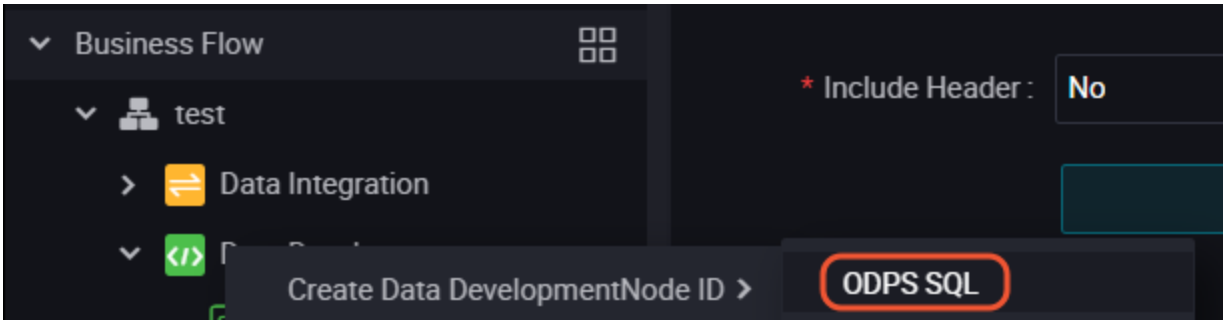
```
{
    " type ": " job ",
    " steps ": [
        {
            " stepType ": " mongodb ",
            " parameter ": {
                " datasource ": " mongodb_us  erlog ",
// Data   source   name
                " column ": [
                    {
                        " name ": " store . bicycle . color ", //
JSON   field   path .  In   this   example ,  the   value   of
color   is   extracted .
                        " type ": " document . document . string
" // The   number   of   fields   in   this   line   must   be
the   same   as   that   in   the   preceding   line  ( the   name
  line ).  If   the   JSON   field   is   a   level - 1   field
,  for   example ,  the   expensive   field   in   this   topic ,
enter   the   string .
                    }
                ],
                " collection  Name  // Collection   name ": "
userlog "
            },
            " name ": " Reader ",
            " category ": " reader "
        },
        {
```

```
            " stepType ": " odps ",
            " parameter ": {
                " partition ": "",
                " isCompress ":  false ,
                " truncate ":  true ,
                " datasource ": " odps_first ",
                " column ": [
                        " mqdata " // Table   column   name   in
 MaxCompute
                ],
                " emptyAsNul  l ":  false ,
                " table ": " mqdata "
            },
            " name ": " Writer ",
            " category ": " writer "
        }
    ],
    " version ": " 2 . 0 ",
    " order ": {
        " hops ": [
            {
                " from ": " Reader ",
                " to ": " Writer "
            }
        ]
    },
    " setting ": {
        " errorLimit ": {
            " record ": ""
        },
        " speed ": {
            " concurrent ":  2 ,
            " throttle ":  false ,
            " dmu ":  1
        }
    }
}
```

After completing the preceding settings, click Run. If the following information is displayed, the code has run successfully.

### Verify the result

Create an ODPS SQL node in your *Business Flow*.




Enter the `SELECT * from mqdata ;` statement to view the data in the mqdata table. You can also run the `SELECT * from mqdata ;` command on the *MaxCompute client* to view the data.

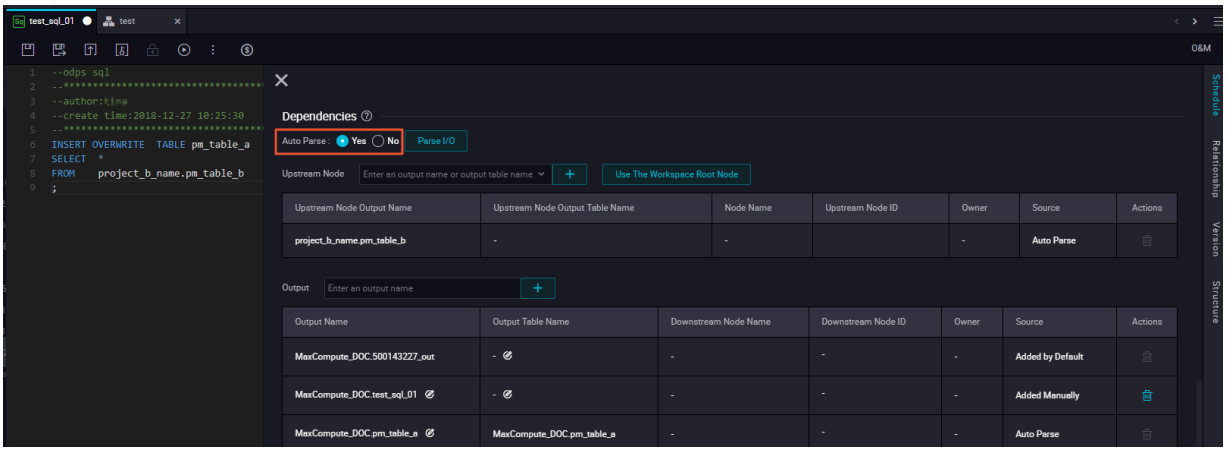
# 3 Data development

## 3.1 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.

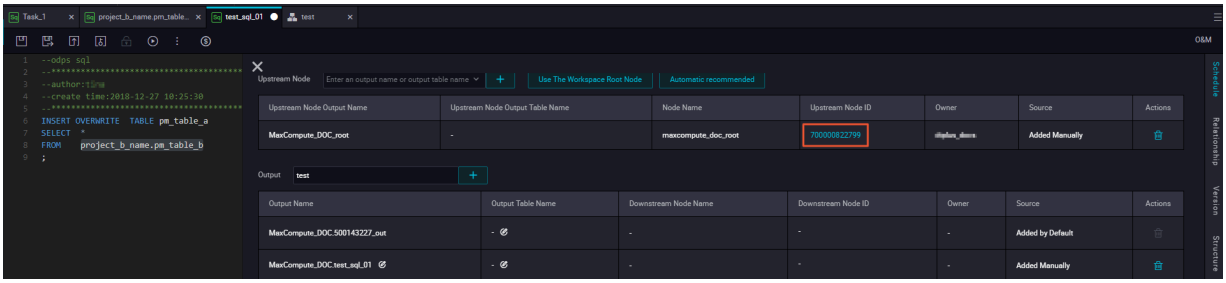
### How to configure the node input of a task

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.

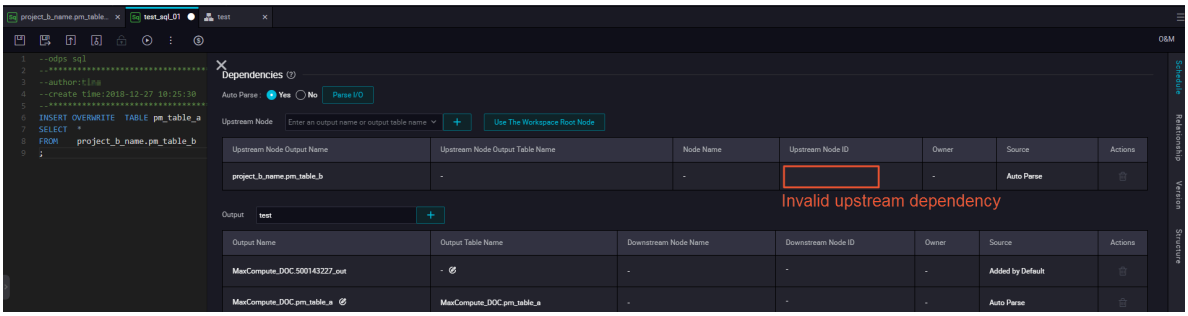


The configuration of task dependencies is essentially to set the dependencies between two nodes. Only the nodes that exist will be able to set up valid dependencies, task dependencies can be set successfully.

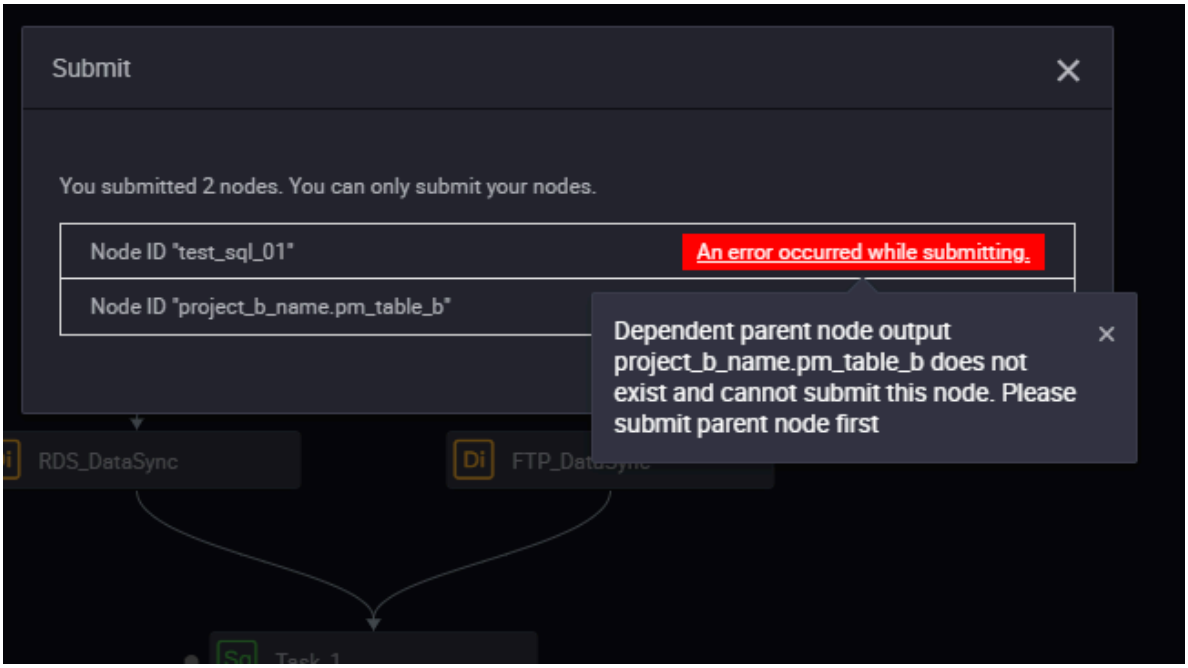
### Invalid upstream dependency

Invalid upstream dependencies are usually in two cases.

1. The parent node does not exist.



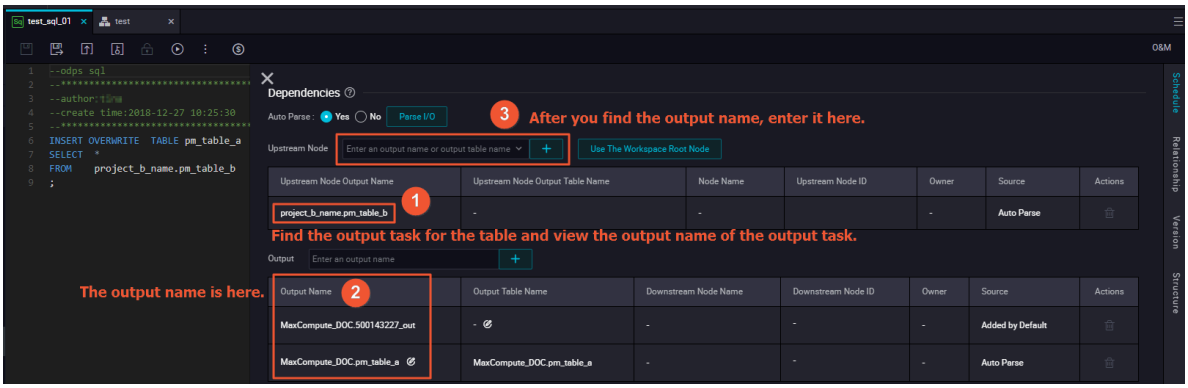

2. The parent node output does not exist.




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.



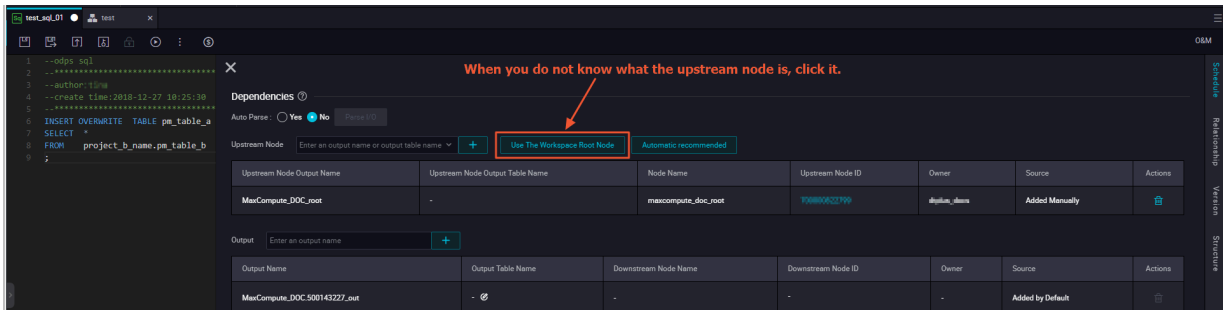

**Note:**

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.

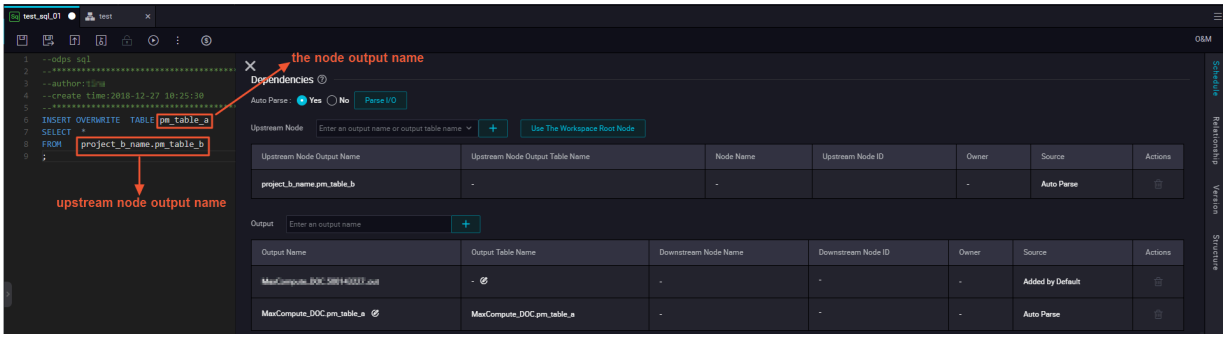
### Automatic parsing

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_na me . 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.

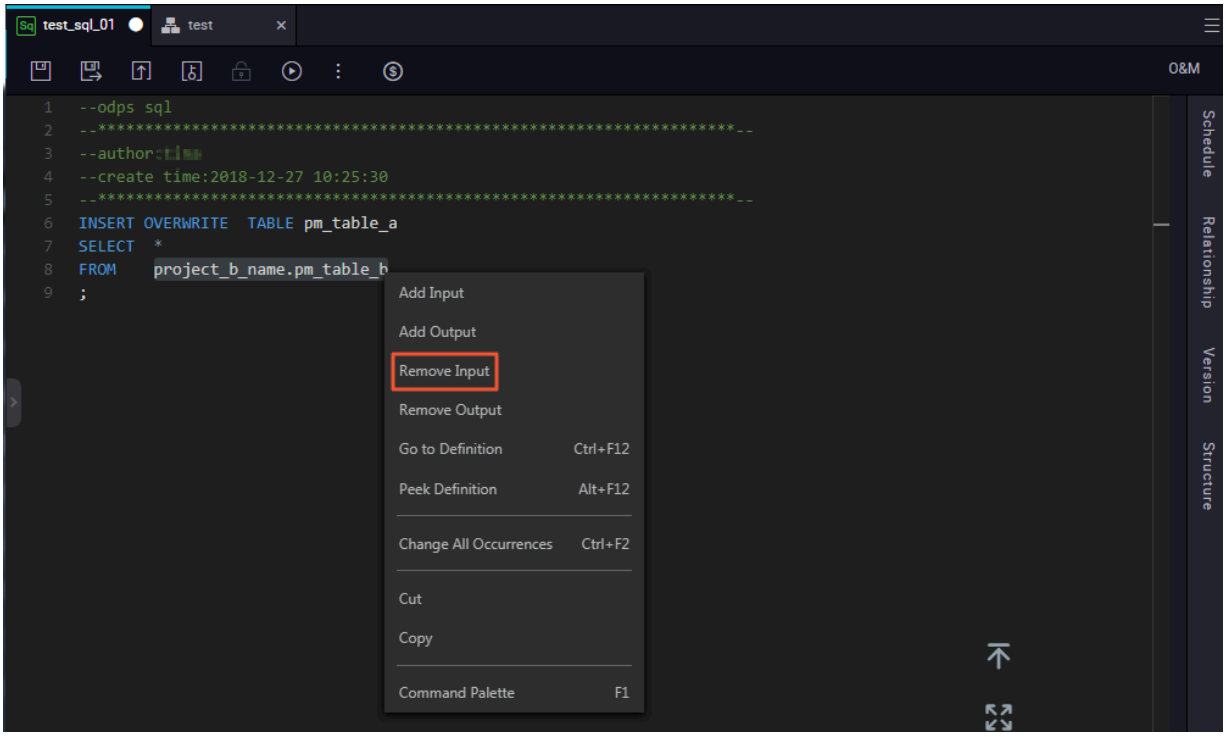
- If a table in the code is referenced or output for multiple times, only one scheduling dependency is parsed.

> **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.

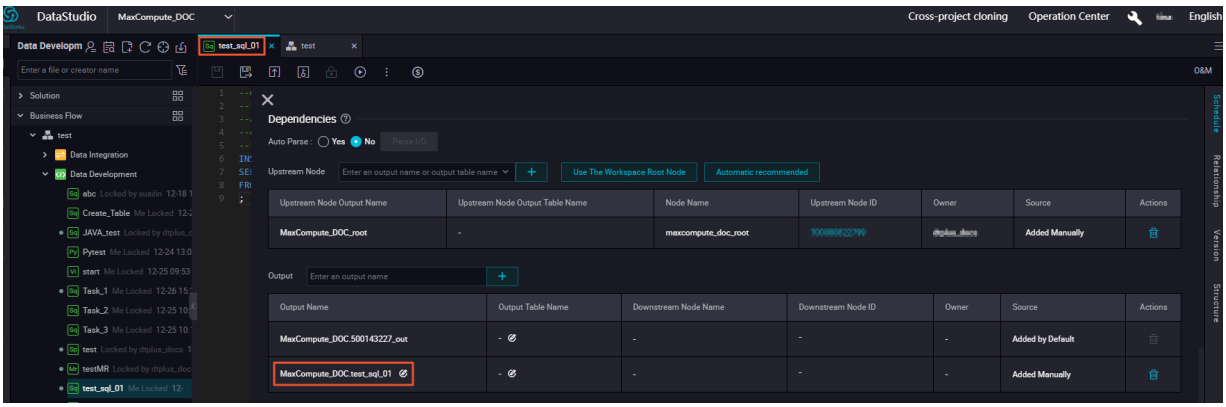

### How to delete the input and output of a table

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.



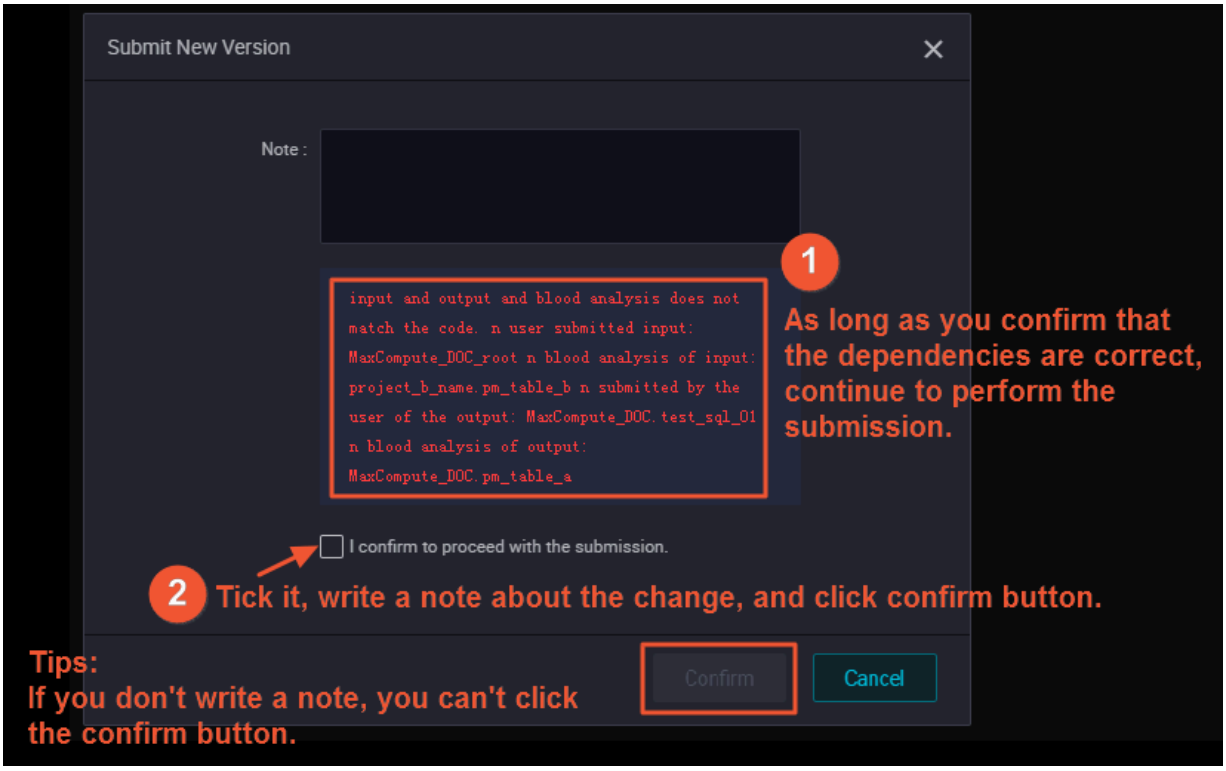

If you are upgrading from DataWorks to DataWorks V2.0, we set the node output for the migrated DataWorks task to `ProjectNam e . NodeName` for you by default.




## 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.




## 3.2 Use Eclipse to develop a Java-based UDF

This topic describes how to develop a Java-based user-defined function (UDF) by using the Eclipse-integrated ODPS plug-in.

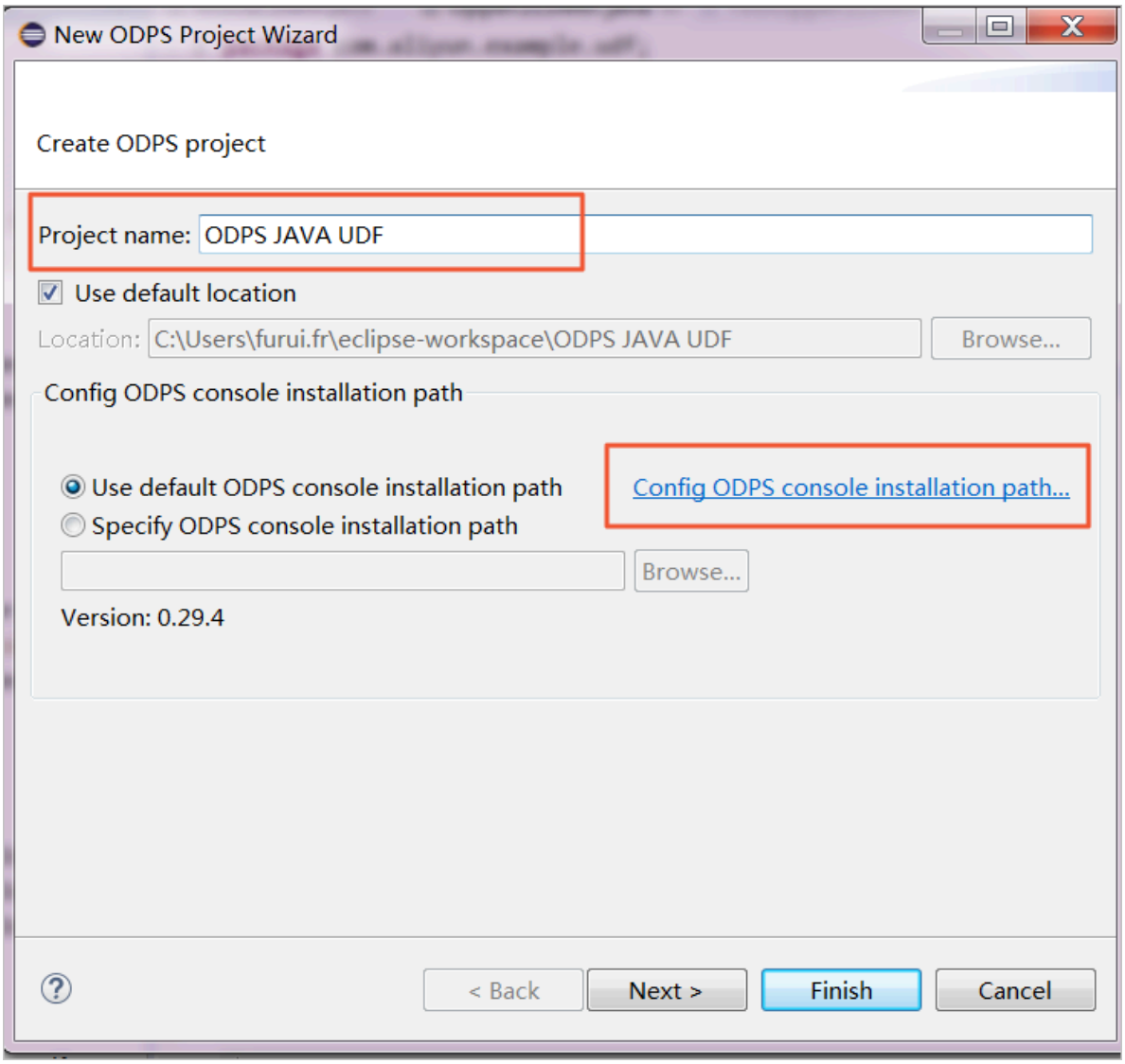
### Preparations

Before developing a Java-based UDF using Eclipse, you need to make the following preparations:

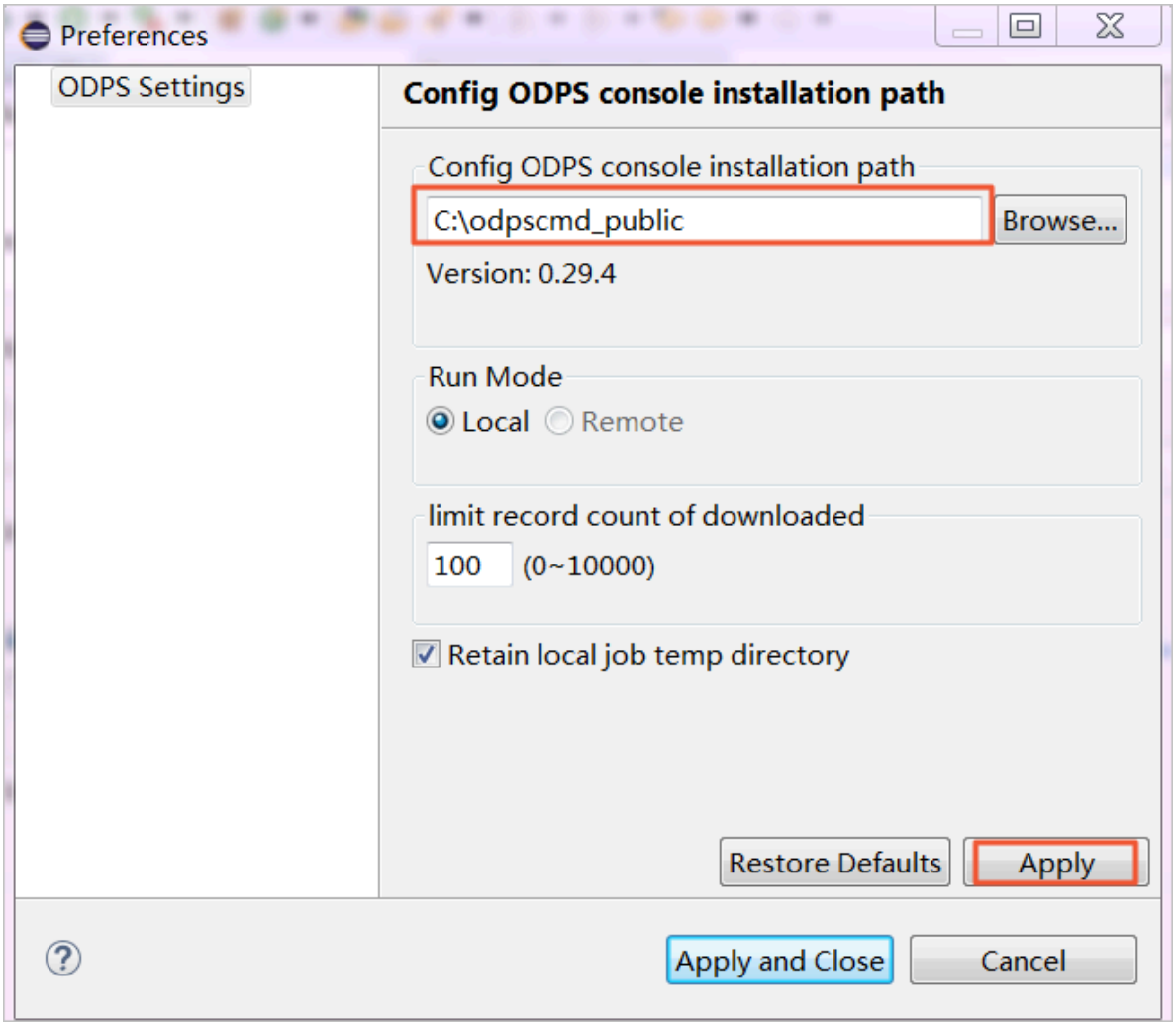
1. Use Eclipse to install the *ODPS plug-in*.

2. Create an ODPS project.

   In Eclipse, choose File > New > ODPS Project, enter the project name, and click Config ODPS console installation path to configure the installation path of the *odpscmd client*.




   Enter the installation package path and click Apply. The ODPS plug-in automatically parses the version of the odpscmd client.

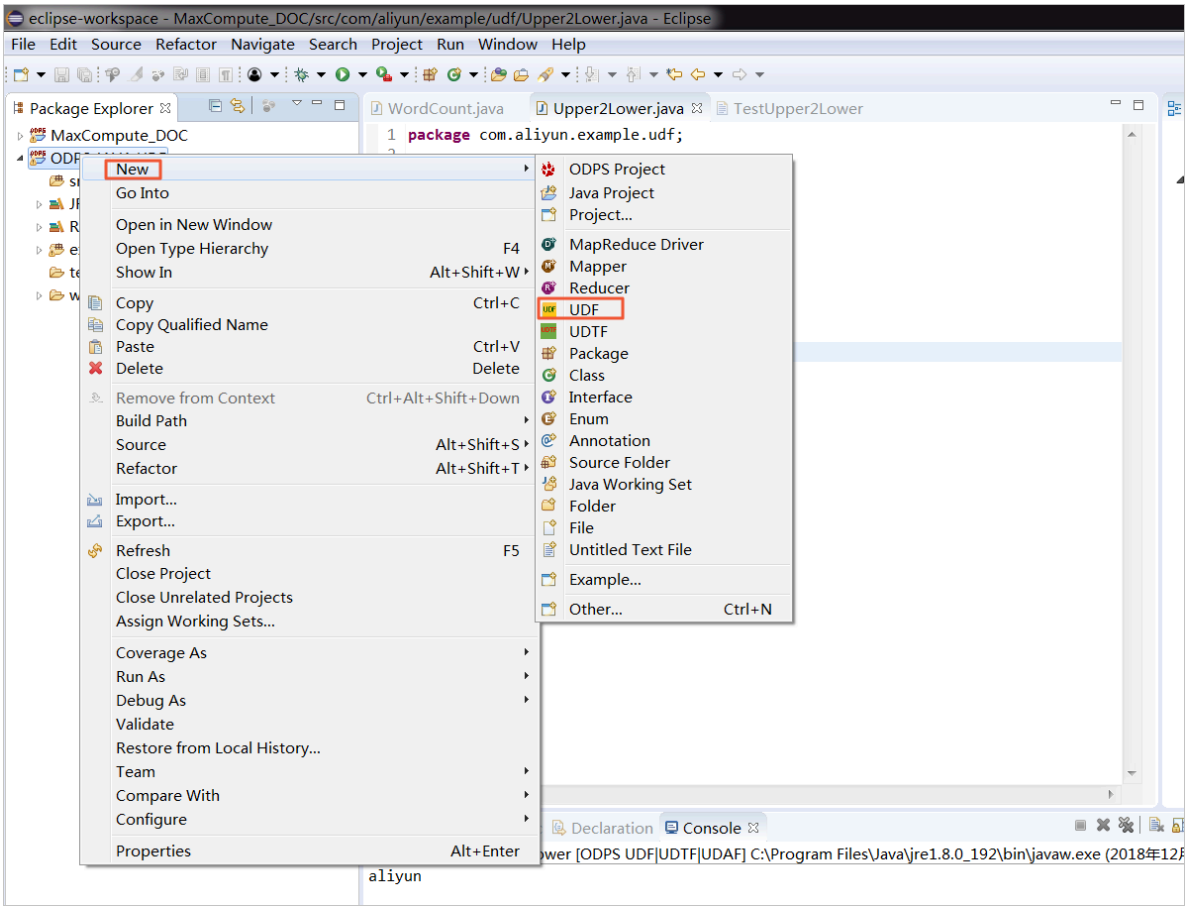



**Click Finish.**

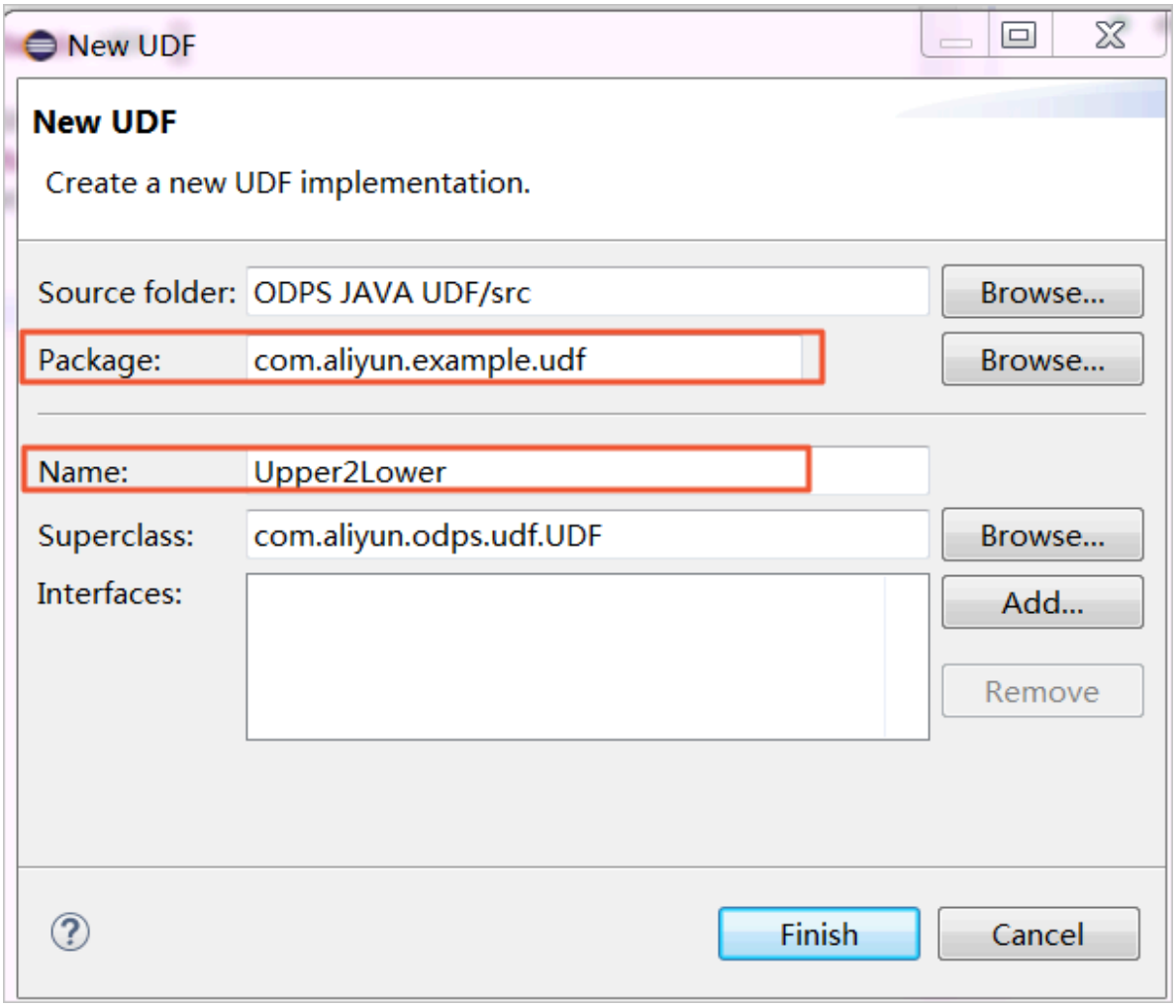
## Procedure

- 1. Create a Java-based UDF in the ODPS project.

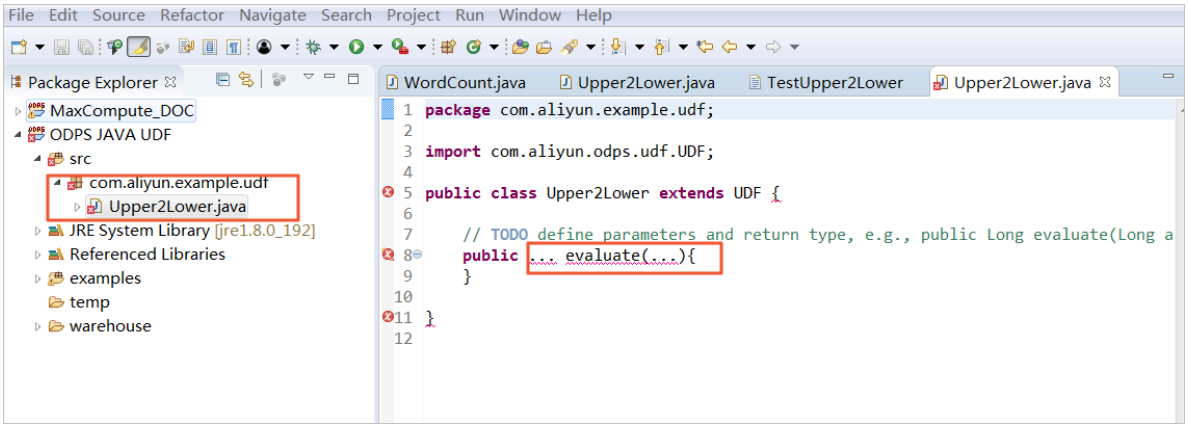
  One the Package Explorer pane, right-click the ODPS Java-based UDF project you have created, and choose New > UDF.




  Set the UDF package to com.aliyun.example.udf and name to Upper2Lower, and click Finish.

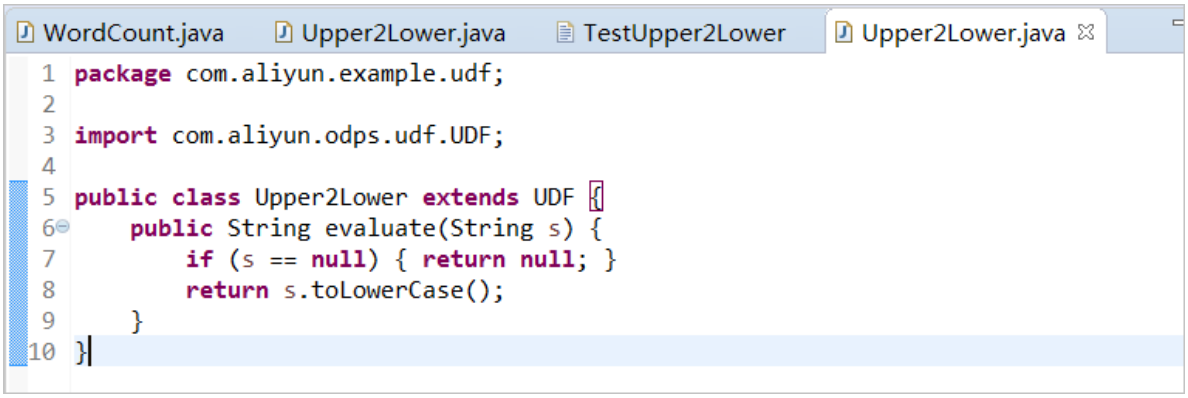



An automatic Java code is generated after you create a UDF. Do not change the name of the evaluate() function.



- 2. Implement the evaluate() function contained in the UDF file.

  Write the function code to be implemented into the evaluate() function. Do not change the name of the evaluate() function. The following is an example of how to convert uppercase letters to lowercase letters.




```
package com . aliyun . example . udf ;

import com . aliyun . odps . udf . UDF ;

public class Upper2Lowe r extends UDF {
    public String evaluate ( String s ) {
        if ( s == null ) { return null ; }
        return s . toLowerCas e ();
    }
}
```

  Save the code.

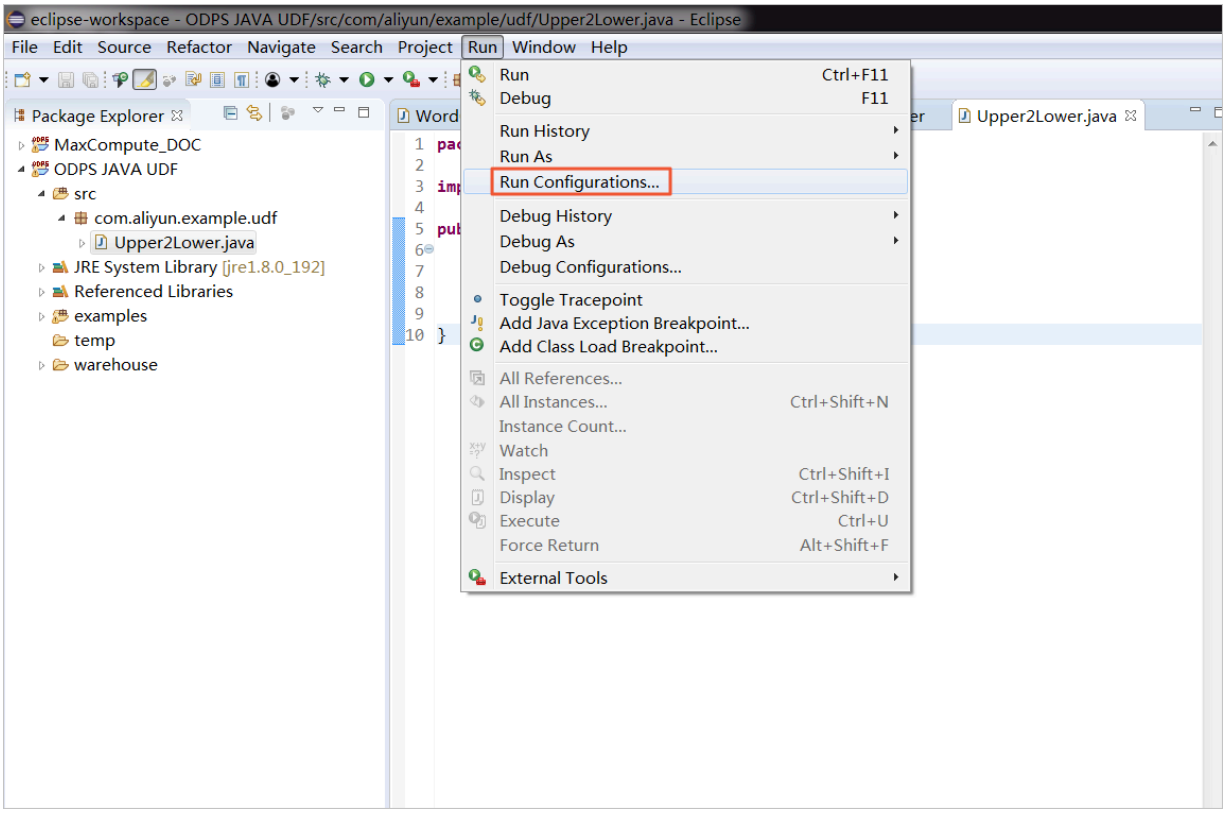
### Test the Java-based UDF code

Before testing the Java-based UDF code, store some uppercase letters on MaxCompute. Create a test table named upperABC using the `create table upperABC ( upper string );` SQL statement on the odpscmd client.

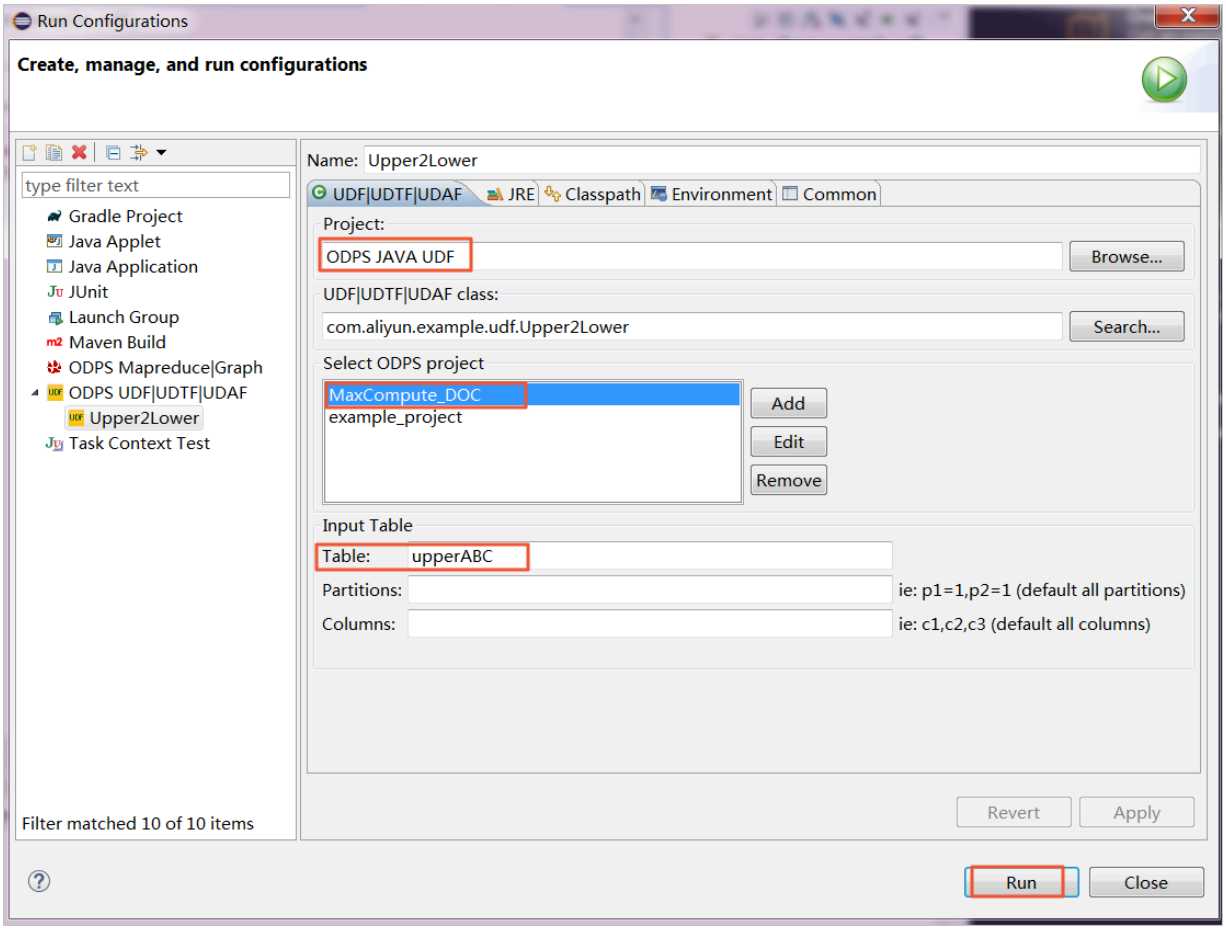
```
odps@ MaxCompute_DOC>create table upperABC(upper string) ;

ID = 20181214094323883gb23j292
OK
```

Use the `insert into upperABC values (' ALIYUN ');` SQL statement to insert the string of uppercase letters 'ALIYUN'.

Choose Run > Run Configurations to set the test parameters.



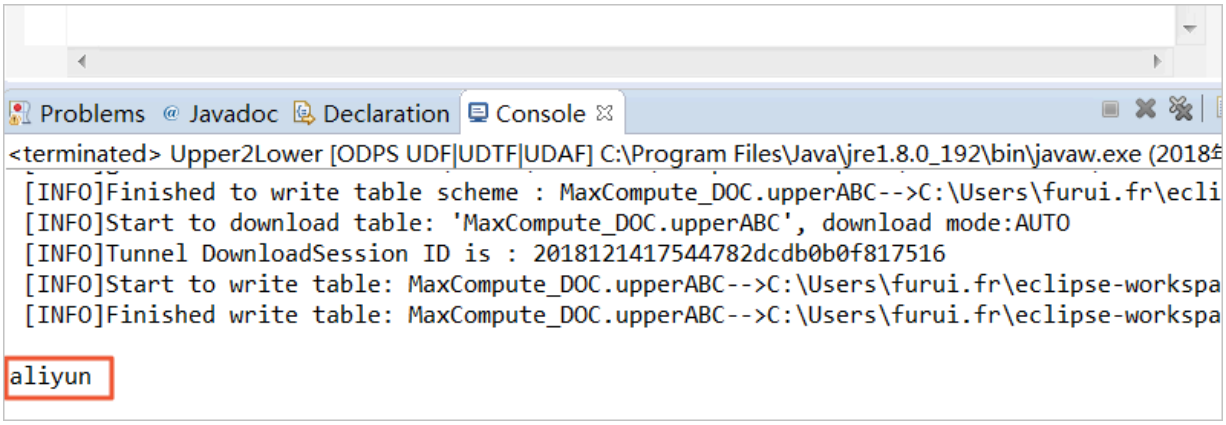

Set the test parameters. Set Project to the name of the Java ODPS project you have created, and set Select ODPS project to the MaxCompute project name. Note that the project name needs to match the name of that connected to the odpscmd client. Set Table to upperABC. After completing all the settings, click Run.




You can view the test result in the Console pane, as shown in the following figure.

**Note:**

Eclipse obtains the string of uppercase letters from the table and converts them to a string of lowercase letters, which is 'aliyun'. However, the uppercase letters stored on MaxCompute are not converted.

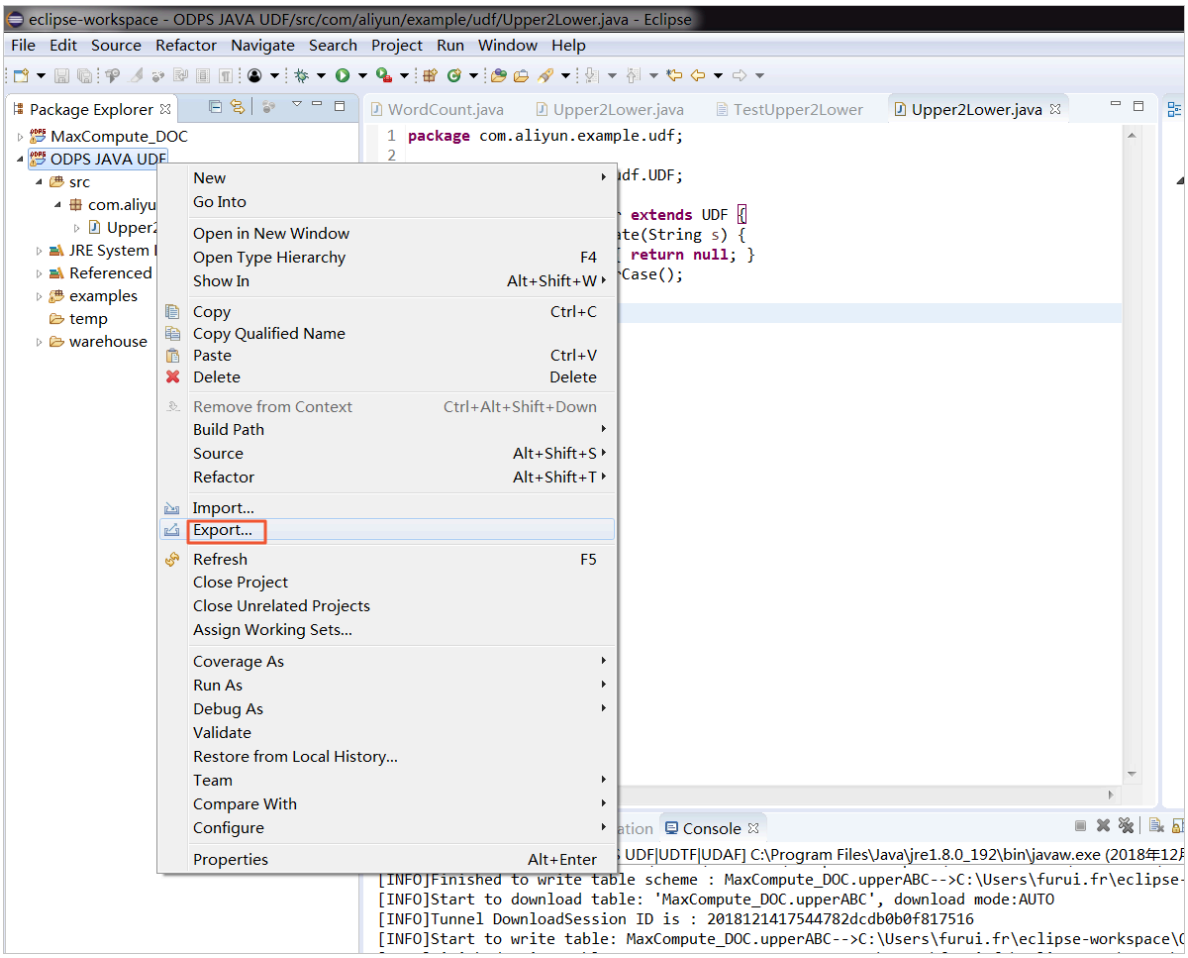


### Use the Java-based UDF

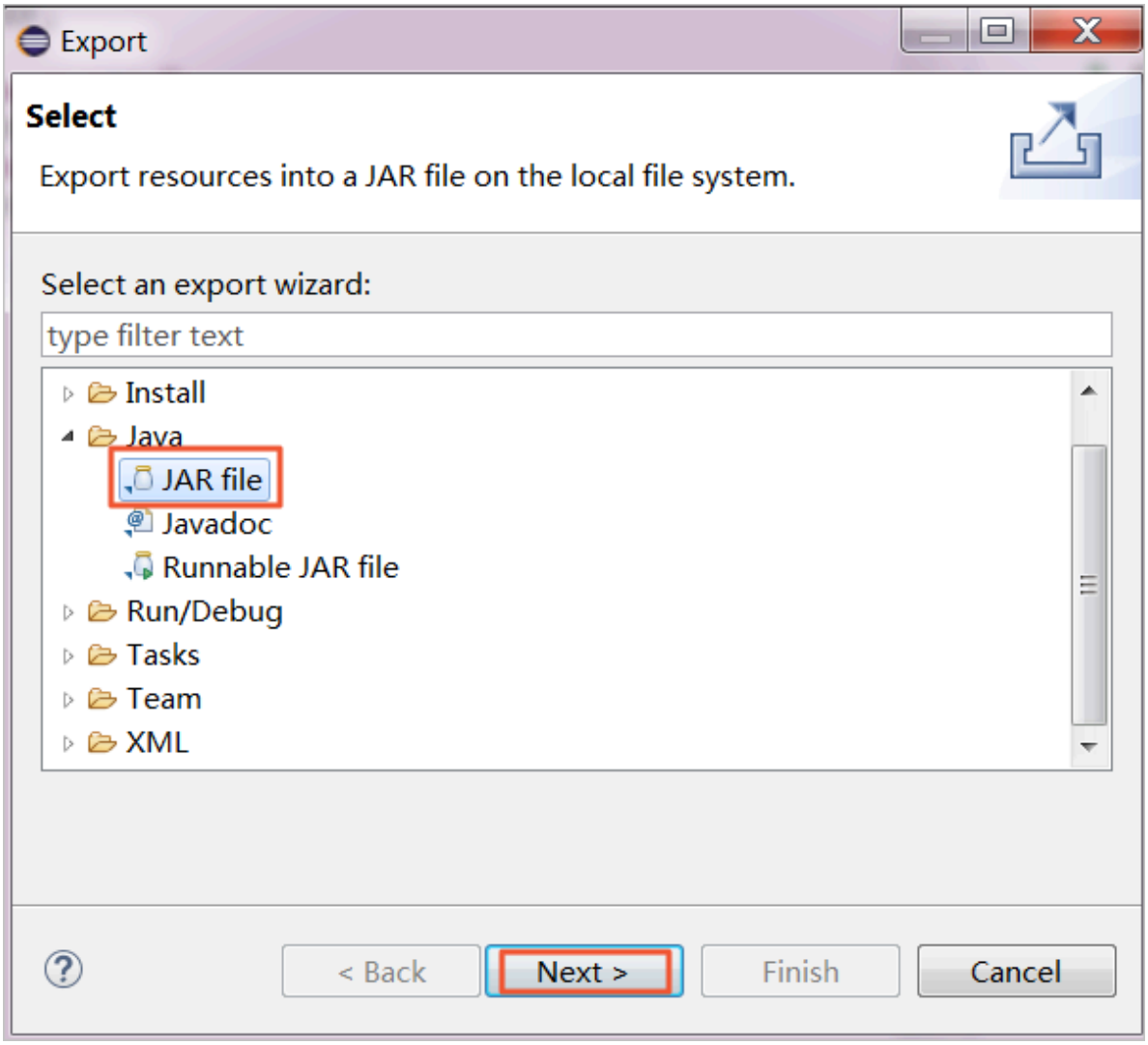
You can use the Java-based UDF after the test is successful. The procedure is as follows:

1. Export the JAR package.

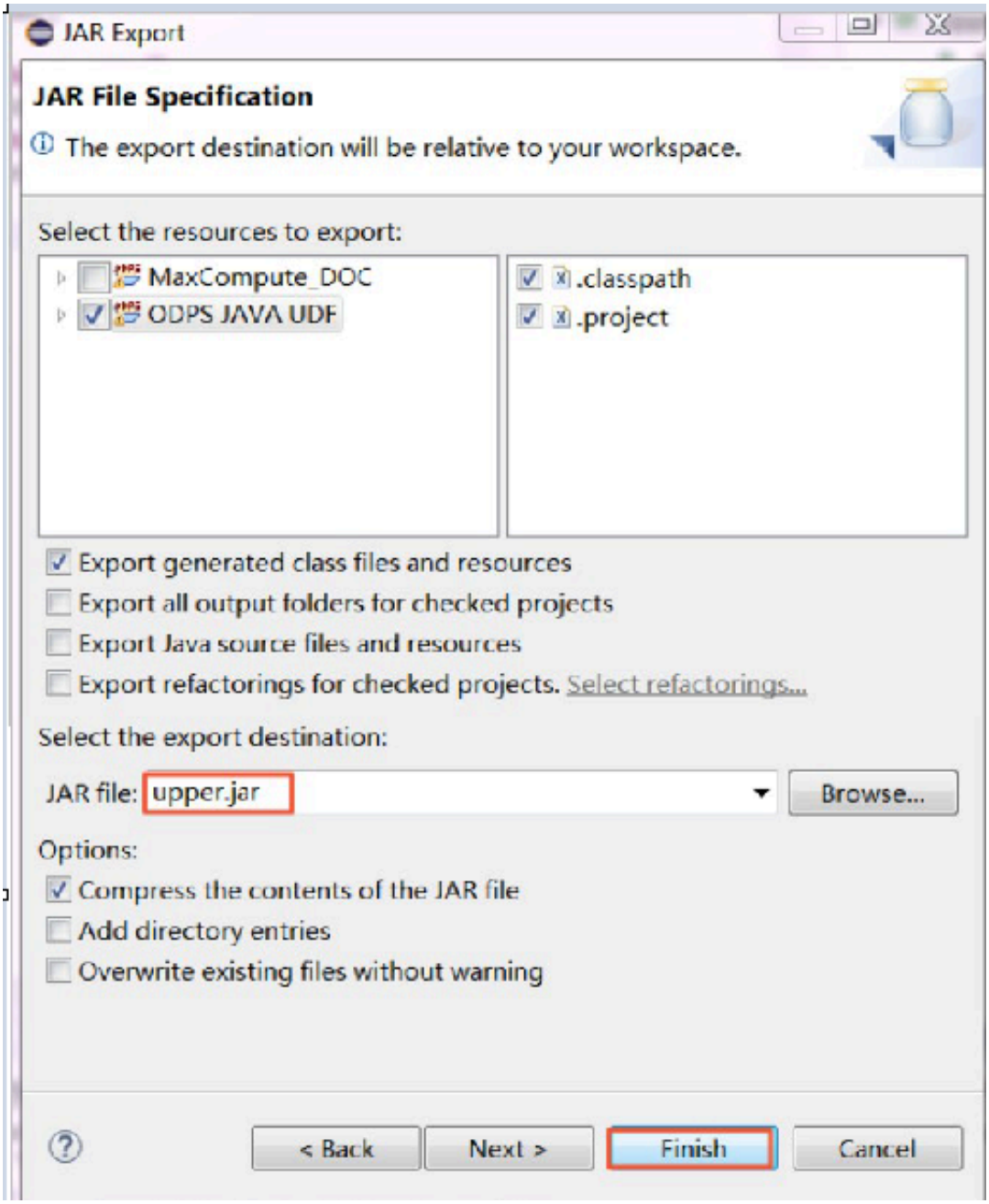
   Right-click the ODPS project you have created and select Export.




   On the displayed page, select JAR file and click Next.

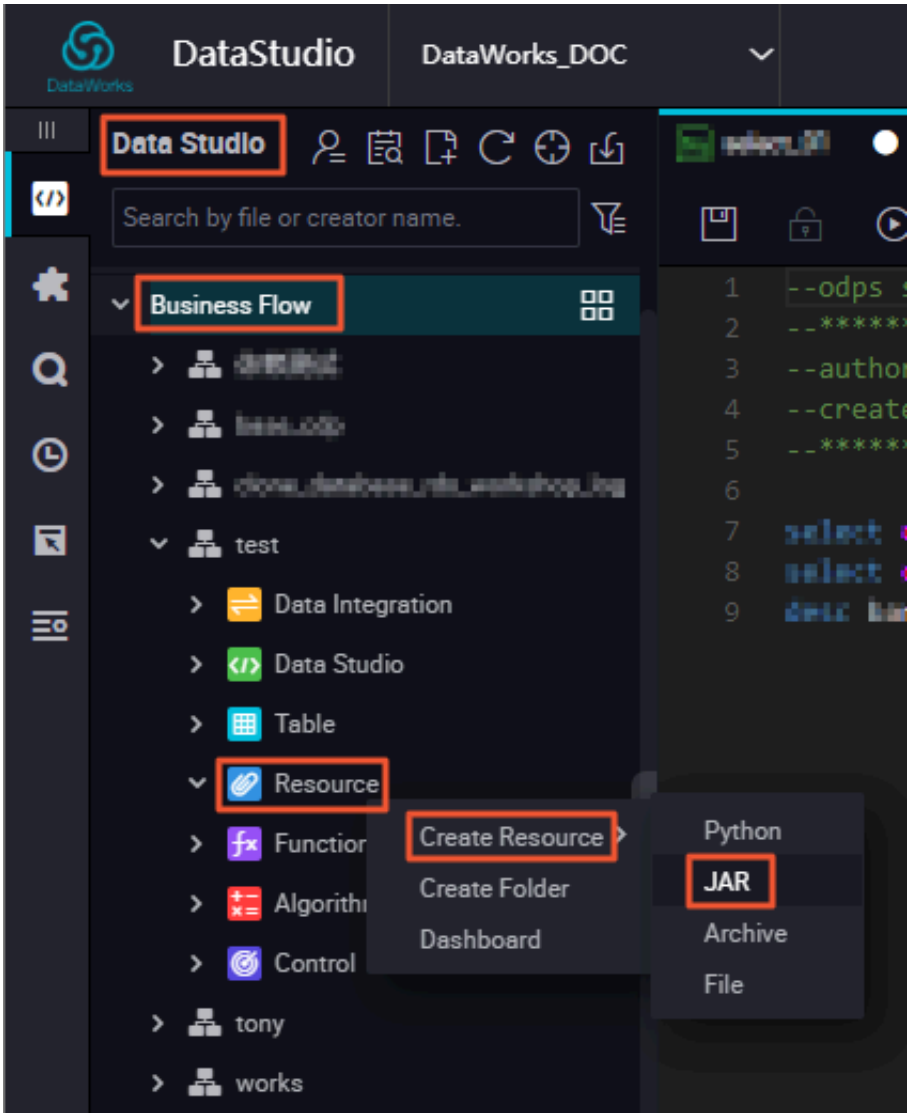



**Enter the JAR package name and click Finish. Then, the JAR package is exported to your workspace directory.**

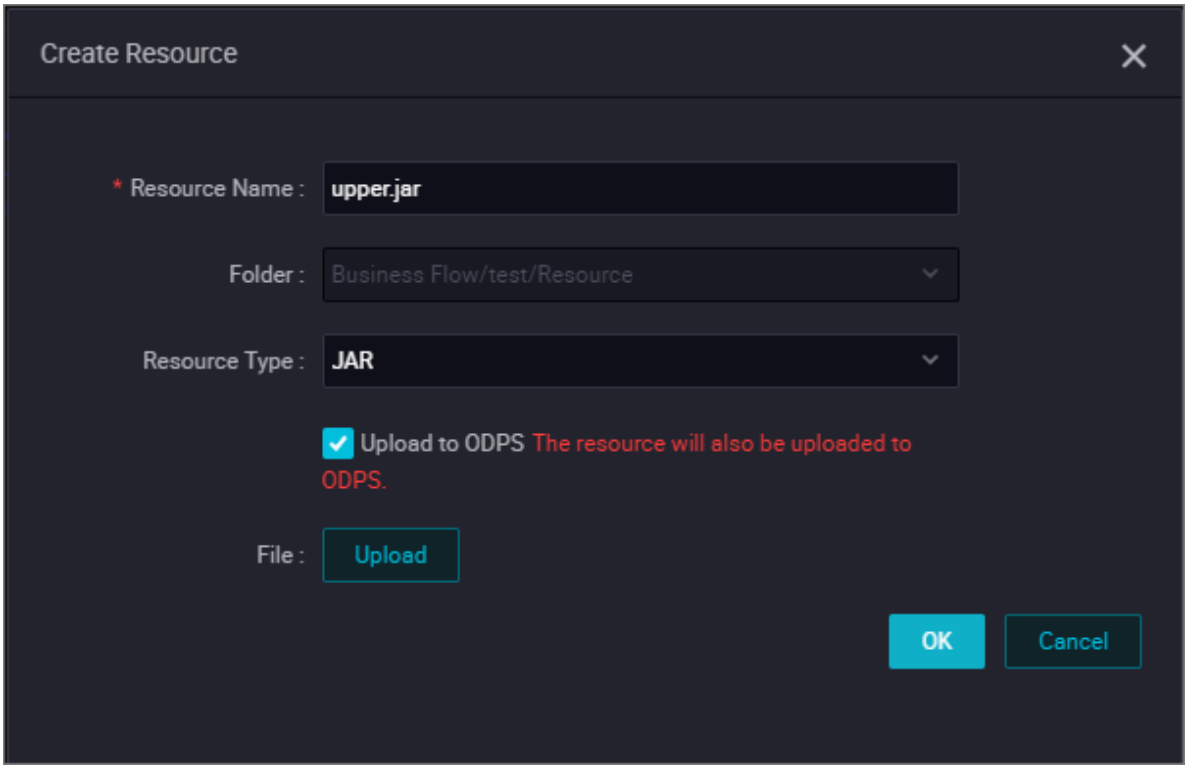

JAR Export
JAR File Specification
The export destination will be relative to your workspace.
Select the resources to export:
MaxCompute_DOC
ODPS JAVA UDF
.classpath
.project
Export generated class files and resources
Export all output folders for checked projects
Export Java source files and resources
Export refactorings for checked projects. Select refactorings...
Select the export destination:
JAR file: upper.jar
Browse...
Options:
Compress the contents of the JAR file
Add directory entries
Overwrite existing files without warning
< Back
Next >
Finish
Cancel

2. Upload the JAR package to DataWorks.

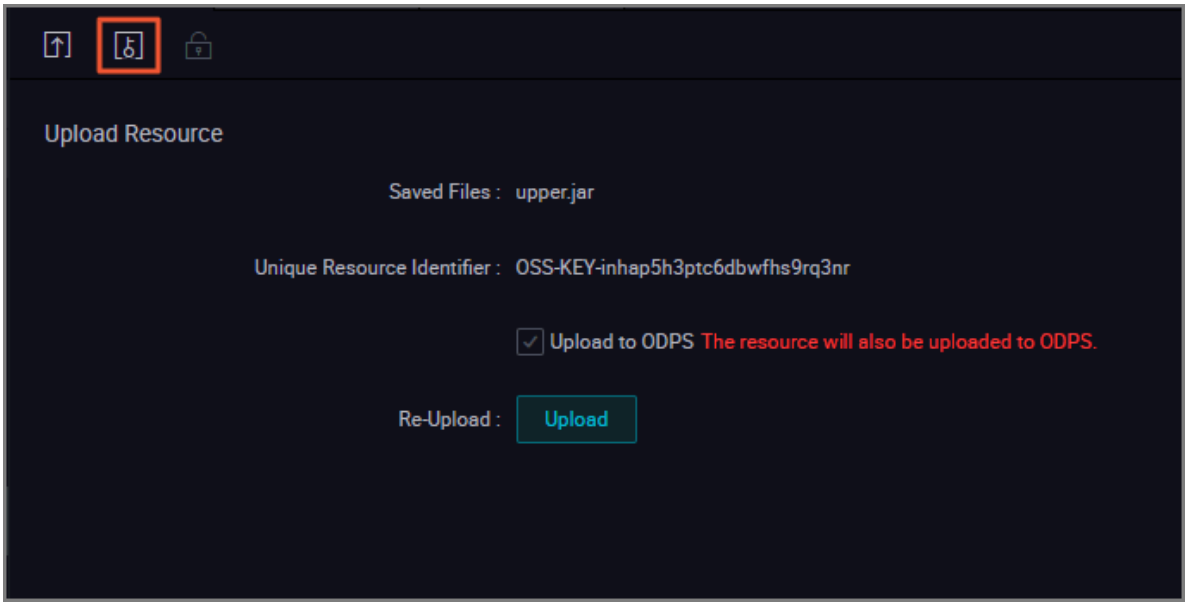
   Log on to the DataWorks console, find the MaxCompute_DOC project, and go to the *Data Studio* page. Choose Business Flow > Resource > Create Resource > JAR and create a *JAR resource*.




   On the displayed page, upload the JAR package you have exported.



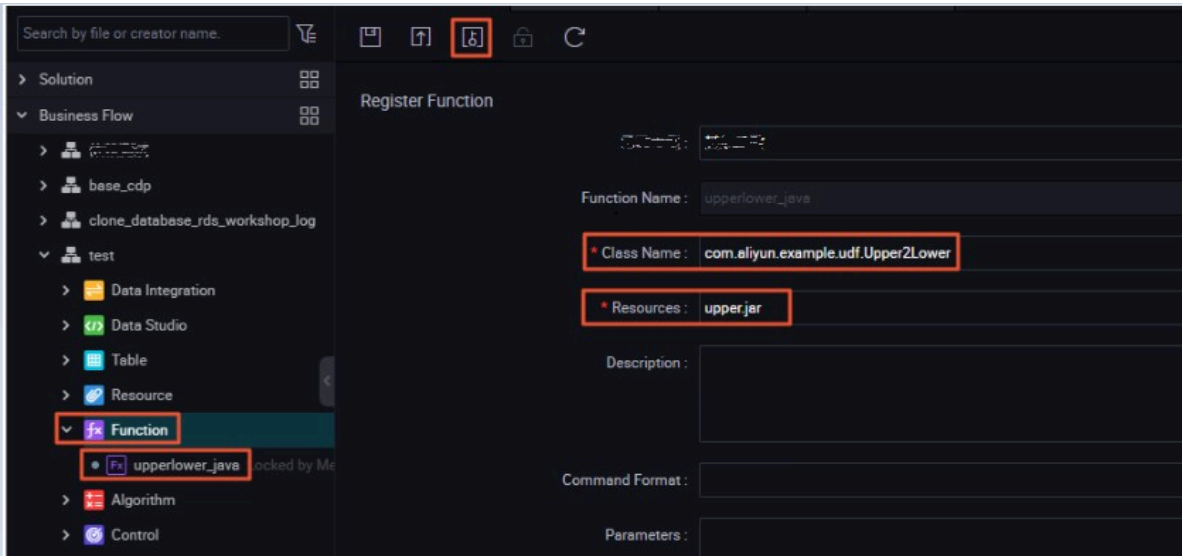

The JAR package is uploaded to DataWorks. To upload it to MaxCompute, click the JAR package and click Submit and Unlock.




You can run the `list resources` command on the odpscmd client to view the uploaded JAR package.

3. Create a resource function.

   After uploading the JAR package to your MaxCompute project, choose Business Flow > Function > Create Function and create a *function* named upperlower_Java. After completing these settings, click Save and Submit and Unlock.




   You can run the `list functions` command on the odpscmd client to view the registered function. Then, the upperlower_Java Java-based UDF registered using Eclipse can be used.

### Check the Java-based UDF

In the odpscmd CLI, run the `select upperlower _Java (' ABCD ') from dual ;` command. In the following figure, the output is abcd, indicating that the function has converted a string of uppercase letters to lowercase letters.


### Additional information

For more information about how to develop Java-based UDFs, see *Java UDF*.

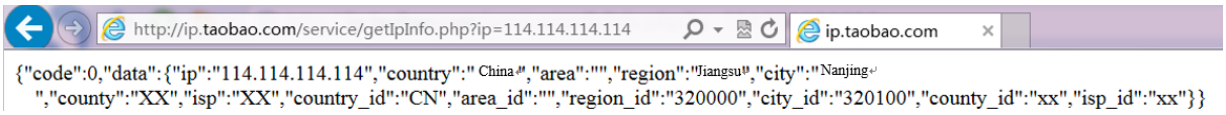
To use IntelliJ IDEA to develop a Java-based UDF, see *Use IntelliJ IDEA to develop a Java-based UDF*.

# 3.3 Use MaxCompute to analyze IP sources

This topic describes how to use MaxCompute to analyze IP sources. The procedure includes downloading and uploading data from an IP address library, writing a user-defined function (UDF), and writing a SQL statement.

## Background

The query APIs of *Taobao IP address library* are *IP address strings*. The following is an example.




HTTP requests are not directly allowed in MaxCompute. However, you can query IP addresses in MaxCompute using one of the following methods:

1. Run a SQL statement and then initiate an HTTP request. This method is inefficient. The request will be rejected if the query frequency is lower than 10 QPS.
2. Download the IP address library to the local server. This method is inefficient and will affect the data analysis in data warehouses.
3. Maintain the IP address library regularly and upload it to MaxCompute. This method is relatively effective. However, you need to maintain the IP address library regularly.

The following further describes the third method.

## Download an IP address library

1. You need to obtain data from an IP address library. This section provides a *demo of an incomplete UTF-8 IP address library*.
2. Download the UTF-8 IP address library and check the data format, as shown in the following figure.

```
0,16777215,"0.0.0.0","0.255.255","","Intranet IP","Intranet IP","Intranet IP"
16777216,16777471,"1.0.0.0","1.0.0.255","Australia","","","",""
16777472,16778239,"1.0.1.0","1.0.3.255","China","Fujian","Fuzhou","Telecom"
```

The first four strings of data are the starting and ending IP addresses, among which the first two are decimal integers and the second two are expressed in dot-decimal notation. The decimal integer format is used to check whether an IP address belongs to the target network segment.

### Upload data from the IP address library

1. Create a table data definition language (DDL) on the *MaxCompute client*, or *create a table on the GUI* in DataWorks.

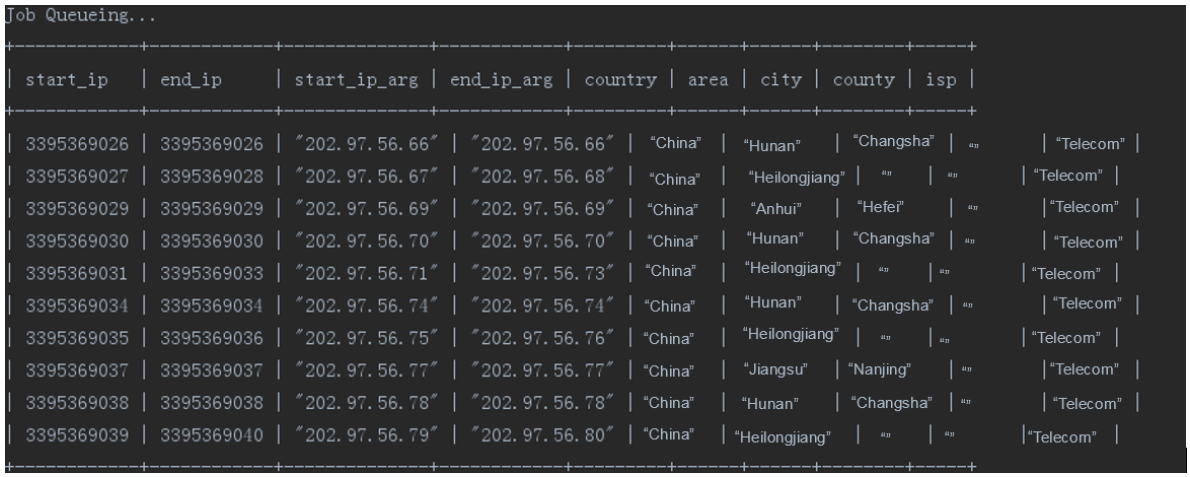
```
DROP TABLE IF EXISTS ipresource ;

CREATE TABLE IF NOT EXISTS ipresource
(
    start_ip BIGINT
    , end_ip BIGINT
    , start_ip_a rg string
    , end_ip_arg string
    , country STRING
    , area STRING
    , city STRING
    , county STRING
    , isp STRING
);
```

2. Run the *Tunnel commands* to upload the ipdata.txt.utf8 file, which is stored on the D drive.

```
odps @ workshop_d emo > tunnel upload D :/ ipdata . txt .
utf8 ipresource ;
```

You can use the `select count (*) from ipresource ;` SQL statement to view the uploaded data. Generally, the quantity of data increases in the library due to regular updates and maintenance.

3. Use the `select count (*) from ipresource limit 0 , 10 ;` SQL statement to view the first 10 pieces of data in the ipresource table, as shown in the following figure.

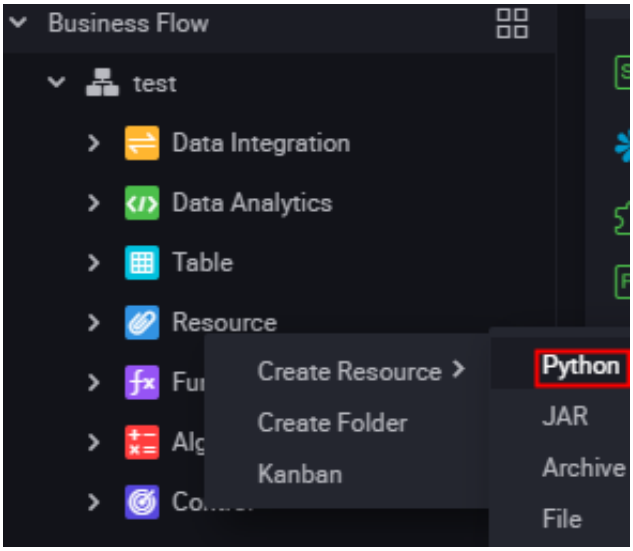


```
Job Queueing...
+------------+------------+--------------+--------------+---------+------+--------+--------+-----+
| start_ip   | end_ip     | start_ip_arg | end_ip_arg   | country | area | city   | county | isp |
+------------+------------+--------------+--------------+---------+------+--------+--------+-----+
| 3395369026 | 3395369026 | "202.97.56.66" | "202.97.56.66" | "China" | "Hunan" | "Changsha" | "" | "Telecom" |
| 3395369027 | 3395369028 | "202.97.56.67" | "202.97.56.68" | "China" | "Heilongjiang" | "" | "" | "Telecom" |
| 3395369029 | 3395369029 | "202.97.56.69" | "202.97.56.69" | "China" | "Anhui" | "Hefei" | "" | "Telecom" |
| 3395369030 | 3395369030 | "202.97.56.70" | "202.97.56.70" | "China" | "Hunan" | "Changsha" | "" | "Telecom" |
| 3395369031 | 3395369033 | "202.97.56.71" | "202.97.56.73" | "China" | "Heilongjiang" | "" | "" | "Telecom" |
| 3395369034 | 3395369034 | "202.97.56.74" | "202.97.56.74" | "China" | "Hunan" | "Changsha" | "" | "Telecom" |
| 3395369035 | 3395369036 | "202.97.56.75" | "202.97.56.76" | "China" | "Heilongjiang" | "" | "" | "Telecom" |
| 3395369037 | 3395369037 | "202.97.56.77" | "202.97.56.77" | "China" | "Jiangsu" | "Nanjing" | "" | "Telecom" |
| 3395369038 | 3395369038 | "202.97.56.78" | "202.97.56.78" | "China" | "Hunan" | "Changsha" | "" | "Telecom" |
| 3395369039 | 3395369040 | "202.97.56.79" | "202.97.56.80" | "China" | "Heilongjiang" | "" | "" | "Telecom" |
+------------+------------+--------------+--------------+---------+------+--------+--------+-----+
```

### Write a UDF

1. Choose Data Studio > Business Flow > Resource. Right-click Resource and choose Create Resource > Python. In the displayed dialog box, enter the name of the

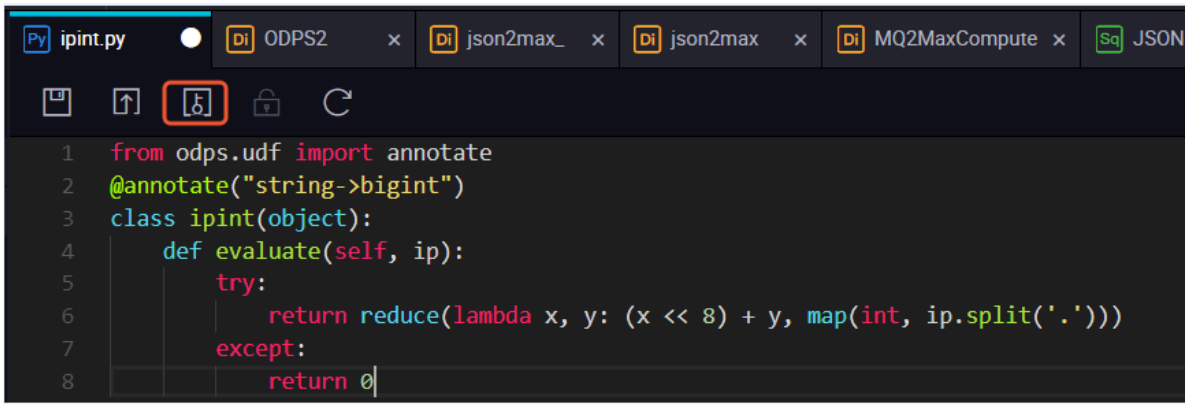
Python resource, select Upload to ODPS and click OK, as shown in the following figure.




2. Write code for the Python resource. The following is an example:

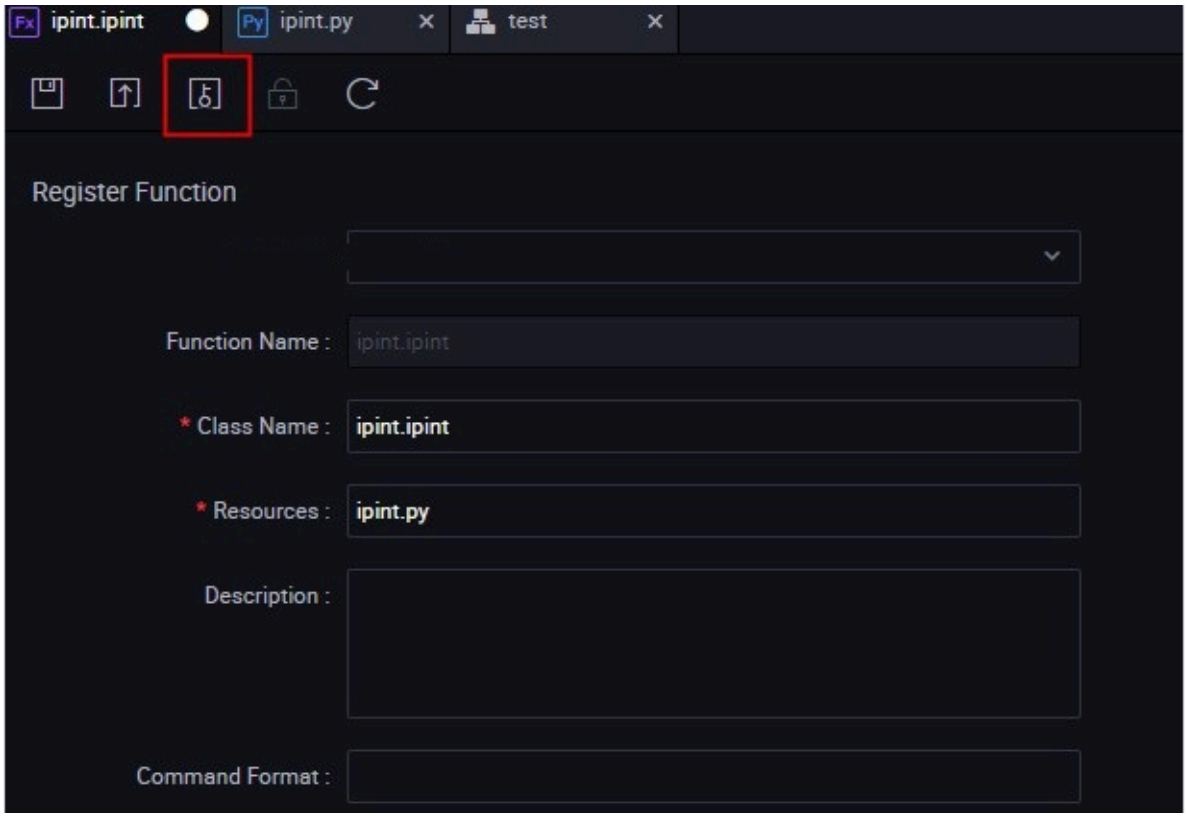
```
 from   odps . udf   import   annotate
@ annotate (" string -> bigint ")
 class   ipint ( object ):
  def   evaluate ( self ,  ip ):
   try :
    return   reduce ( lambda   x ,  y : ( x  <<  8 ) +  y ,  map (
int ,  ip . split ('.')))
   except :
    return   0
```

Click Submit and Unlock.

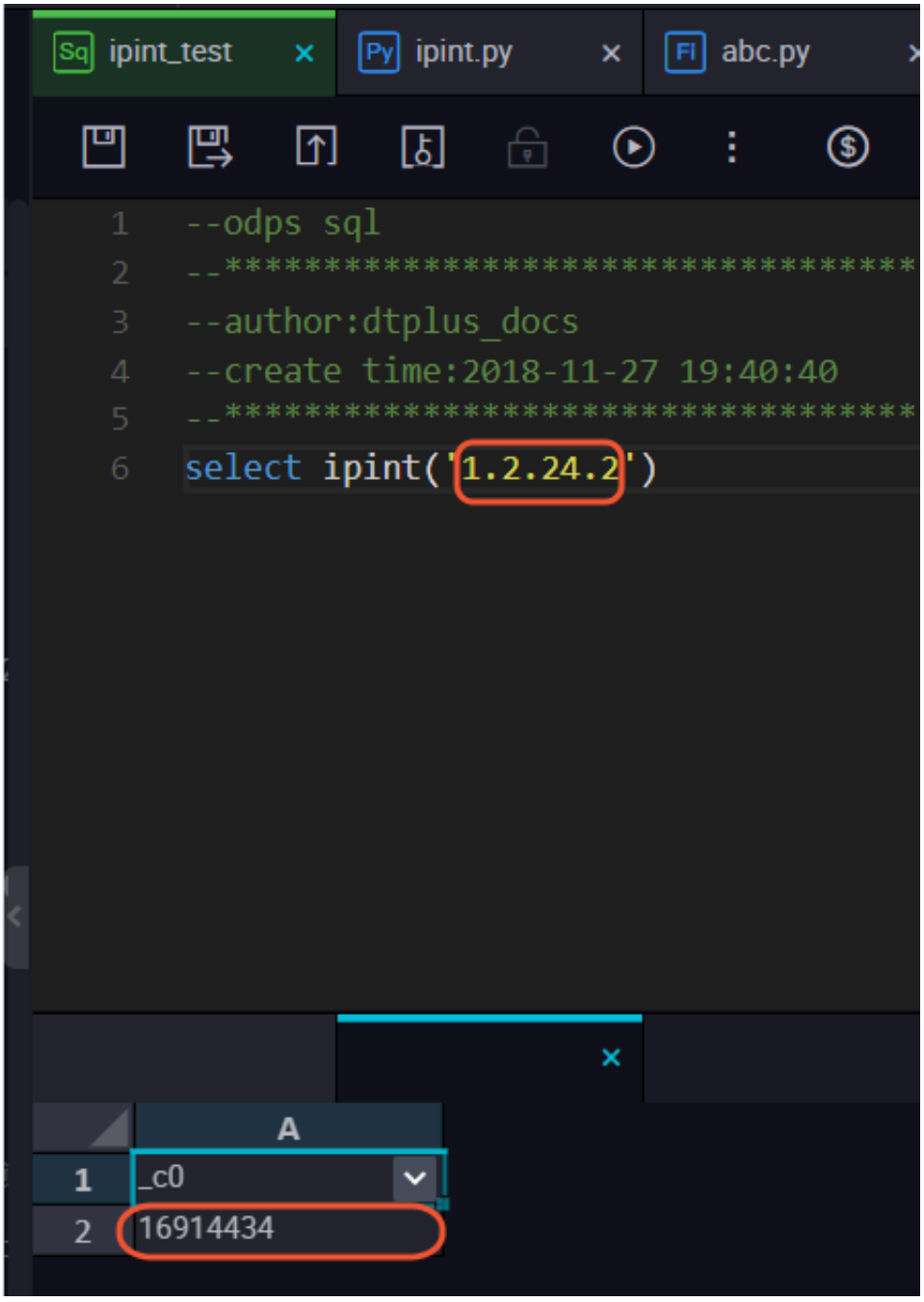


3. Choose Data Studio > Business Flow > Function. Right-click Function and select Create Function.

   Set the function class name to `ipint . ipint`, and the folder to the resource name, and click Submit and Unlock.



4. Create an ODPS SQL node and run the SQL statement to check whether the ipint function works as expected. The following is an example.




You can also create a local *ipint.py* file and use the *MaxCompute client* to upload the resource.

```
odps @ MaxCompute _DOC > add py D :/ ipint . py ;
OK : Resource ' ipint . py ' have been created .
```

After uploading the resource, use the client to *register the function*.

```
odps @ MaxCompute _DOC > create function ipint as ipint .
ipint using ipint . py ;
Success : Function ' ipint ' have been created .
```

The function can be used after registration. You can use `select ipint (' 1 . 2 . 24 . 2 ');` on the client to test the function.

**Note:**

You can perform *cross-project authorization* to share the UDF with other projects under the same Alibaba Cloud account.

1. Create a package named ipint.

```
odps @ MaxCompute _DOC > create package ipint ;
OK
```

2. Add the UDF to the package.

```
odps @ MaxCompute _DOC > add function ipint to package
ipint ;
OK
```

3. Allow a bigdata_DOC project to install the package.

```
odps @ MaxCompute _DOC > allow project bigdata_DO C to
install package ipint ;
OK
```

4. Switch to a bigdata_DOC project that needs to use the UDF and install the package.

```
odps @ MaxCompute _DOC > use bigdata_DO C ;
odps @ bigdata_DO C > install package MaxCompute _DOC .
ipint ;
OK
```

5. Then, the UDF can be used. If a user (such as Bob) of the bigdata_DOC project wants to access the resource, the administrator can grant the access permission to the user by using the ACL.

```
odps @ bigdata_DO C > grant Read on package MaxCompute
_DOC . ipint to user aliyun $ bob @ aliyun . com ; -- Use
the ACL to grant the package access permission to
Bob .
```

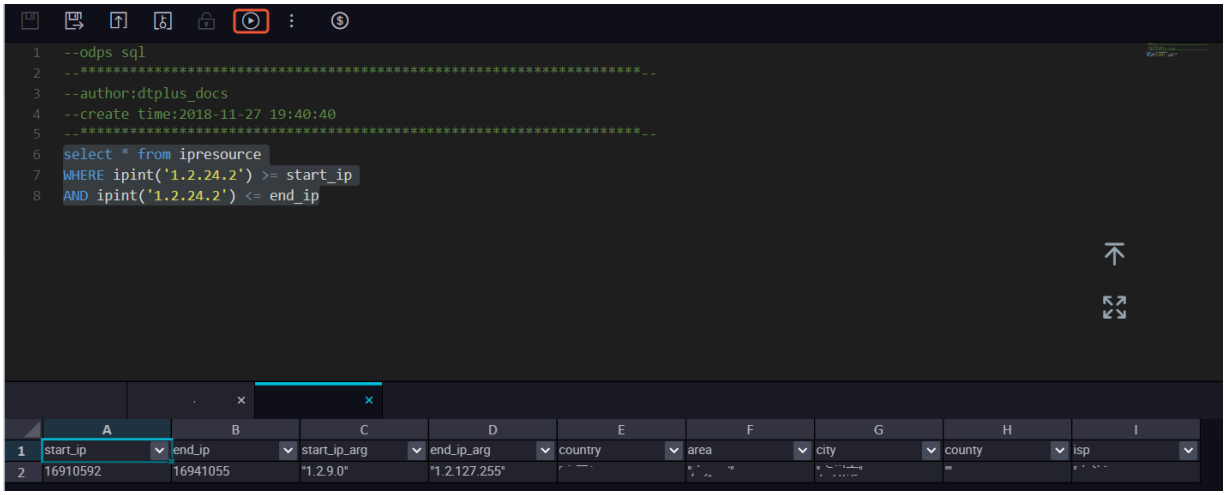
Use the IP address library in SQL

**Note:**

This section uses the IP address 1.2.254.2 as an example. You can use a specific field to query an IP address as needed.

You can use the following SQL code to view the test result:

```
select  *  from   ipresource
WHERE   ipint (' 1 . 2 . 24 . 2 ') >=  start_ip
AND   ipint (' 1 . 2 . 24 . 2 ') <=  end_ip
```




To ensure the data accuracy, you can regularly obtain data from the Taobao IP address library to maintain the ipresource table.

## 3.4 Performing a task at a specific time with branch node

### Background of branch nodes

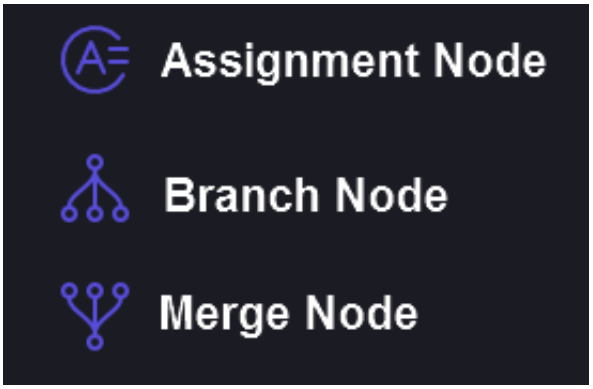
During the daily use of DataWorks, you may often encounter the following problem: I have a node that needs to be executed on the last day of each month. How should I set it up?

Answer: Before the *branch node* appears, the Cron expression can not express this scene, so it is temporarily unavailable to support.

Now, DataWorks *officially supports branch nodes*. With branch nodes, we can apply the switch-case programming model to perfectly meet the above requirements.

### Branch nodes and other control nodes

On the Data Development page, you can see the various control nodes currently supported by DataWorks, including assignment nodes, branch nodes, merge nodes and so on.




The functions of various types of control nodes:

- Pass its own results to the downstream assignment node:

  The *assignment node* reuses the characteristics that the *node context* depends. Based on the two existing constant/variable node context, the assignment node comes with a custom context output. DataWorks captures the select result or the print result of the assignment node. This result is used as the value of the context output parameter in the form of outputs for reference by downstream nodes.
- Determine which downstream branch nodes are normally executed:

  The *branch node* reuse the characteristics of the *input and output* set in the dependencies on DataWorks.

  For common nodes, the output of the node is only a globally unique string. When the downstream needs to set dependencies, searching for this globally unique string as input to the node can be hung into the list of downstream nodes.

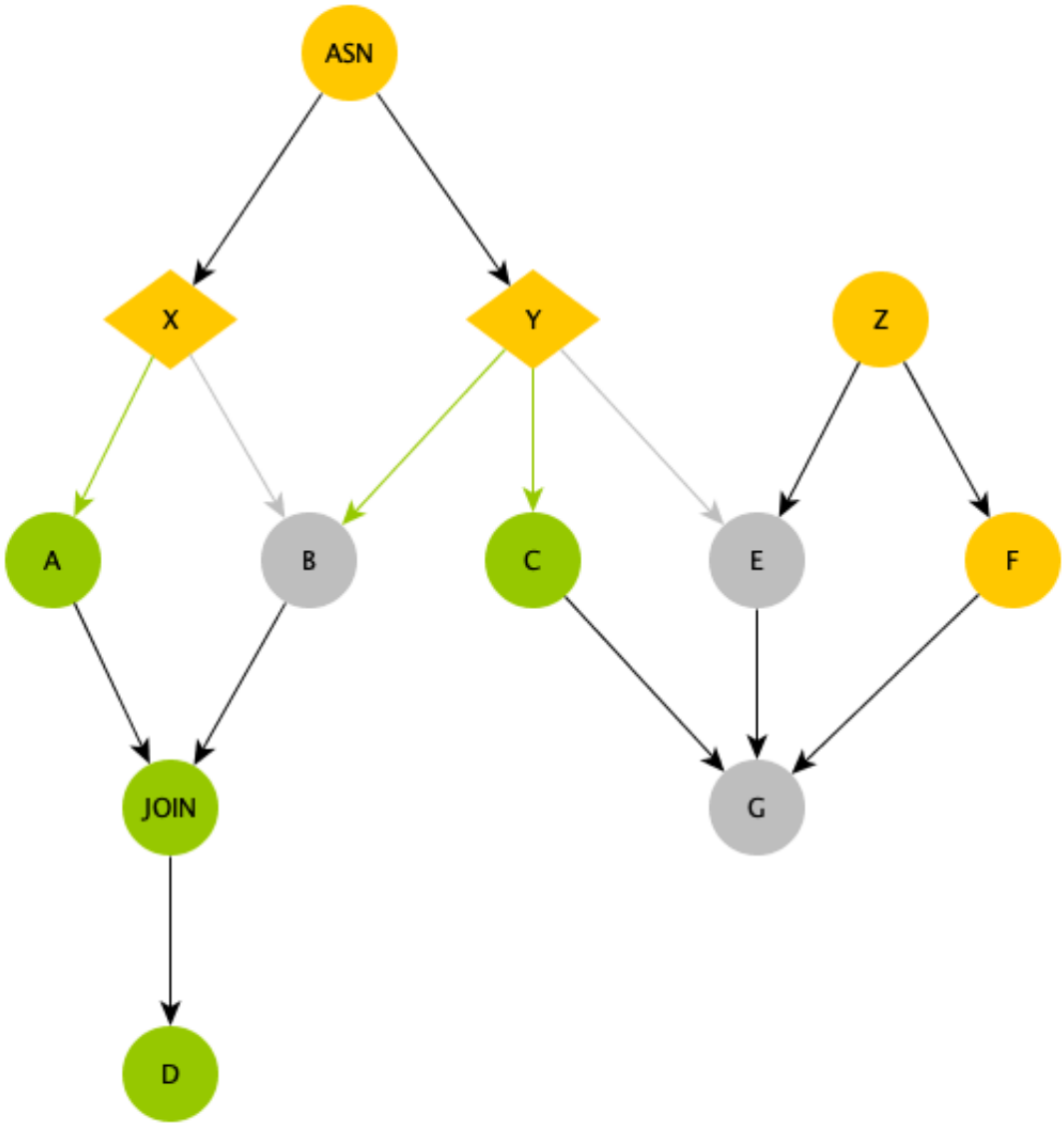
  However, for a branch node, we can associate a condition for each output: when the downstream set dependency, we can selectively use the output of a certain condition as the output of the branch node. In this way, when the node becomes the downstream of the branch node, it is also associated with the condition of the branch node: that is, the condition is satisfied, and the downstream corresponding to the output is executed normally; the other downstream nodes corresponding to the output that does not meet the criteria are set to run empty.
- The merge node that is normally scheduled regardless of whether the upstream performs normally:

  For branches that are not selected by the branch node, DataWorks puts all node instances on this branch link as empty run instances. That is to say, once an

upstream of a certain instance is running empty , this instance itself becomes empty.

Dataworks can currently prevent this empty run attribute from being passed without restriction by *merge node*: for a merge node instance, no matter how many empty run instances its upstream has, it will succeed directly and will no longer leave the downstream empty running.

From the figure below, you can see the logical relationship of the dependency tree in the presence of a branch node.




- ASN: an *assignment node*, which is used to calculate more complex situations to prepare for branch node conditional selection.

- X/Y: branch nodes, they are downstream of the assignment node ASN, and make branch selection according to the output of the assignment node. As shown in the green line in the figure, the node X selects the left branch, the node Y selects the two branches on the left:
  - The node A/C are executed normally because they are downstream of the output selected by the node X/Y.
  - Although the node B is downstream of the branch selected by the node Y, since the node X does not select this output, the node B is set to run empty.
  - The node E is not selected by node Y, so even if there is an ordinary upstream named node Z, it is also set to run empty.
  - The upstream node E of the nodeG runs empty, so even if the node C/F are both executed normally, the node E also runs empty.
  - How can the empty running properties no longer be passed down?

    JOIN node is a merge node. Its special function is to stop the transfer of empty run properties. You can see that because the node D is downstream of the JOIN node, the empty run attribute of the node B is blocked, and the node D can start running normally.

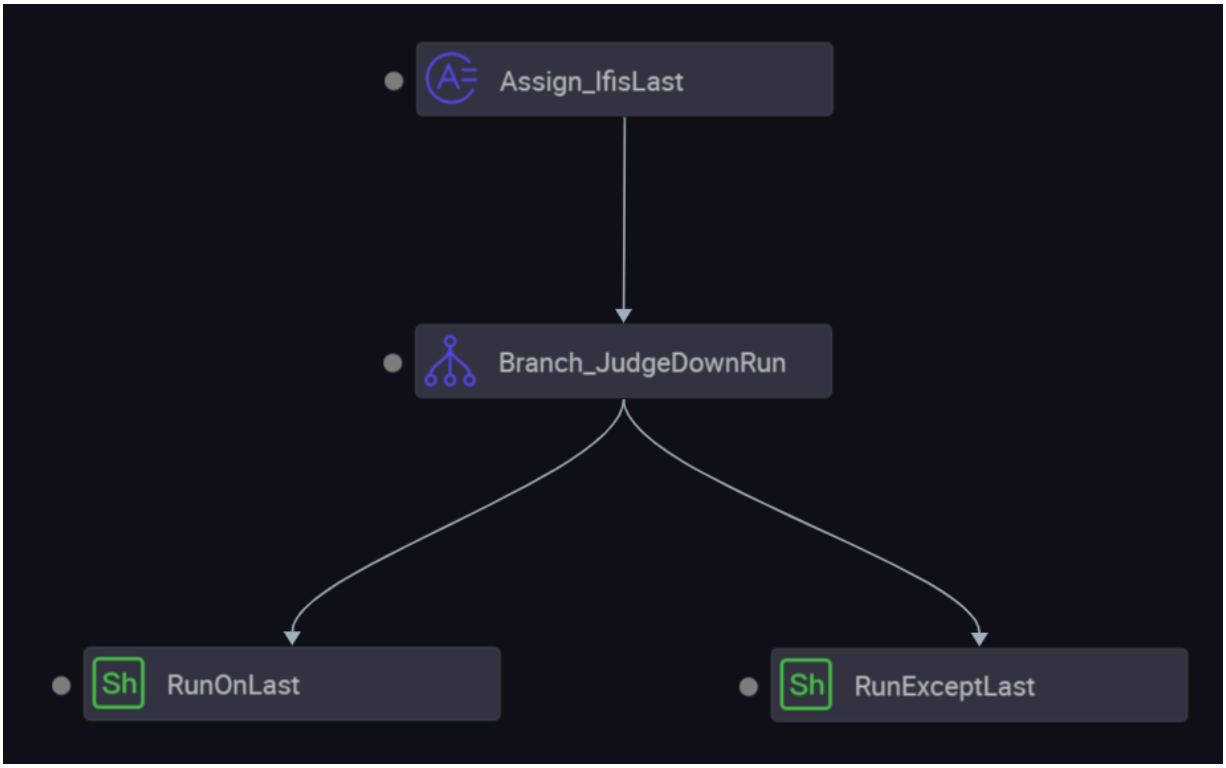
By using branch nodes to cooperate with other control nodes, we can satisfy the requirement scenario where a node only runs on the last day of each month.

### Use a branch node

Define task dependencies

First you need to define a set of task dependencies:

1. The root assignment node calculates whether the current day is the last day of the month by timing `SKYNET_CYC TIME` . If it is, the output is 1, and if it is not, the output is 0. This output is captured by DataWorks and passed to the downstream.
2. The branch node defines the branch according to the output of assignment node.
3. The two shell nodes are hung under the branch node and perform different branch logic.



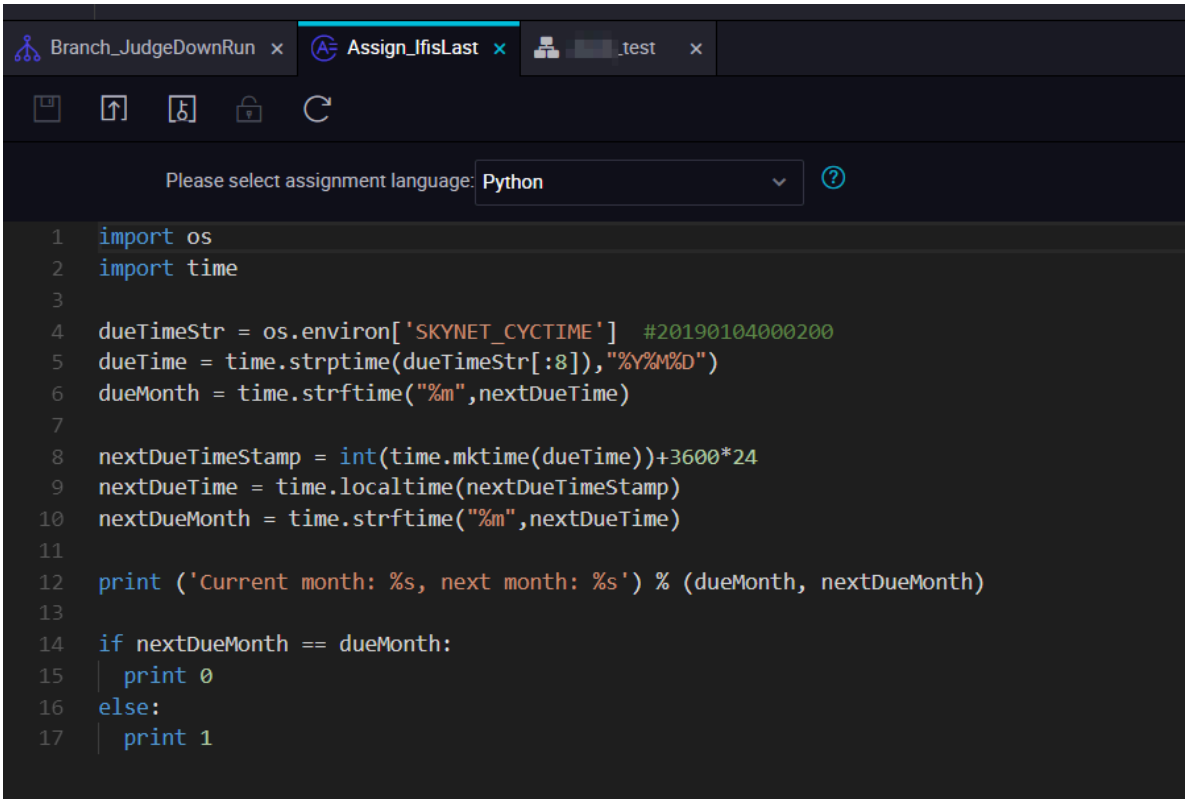

**Define assignment nodes**

**The assignment node comes with an outputs when it is new, the code for the assignment node supports three kinds of SQL/SHELL/Python.**

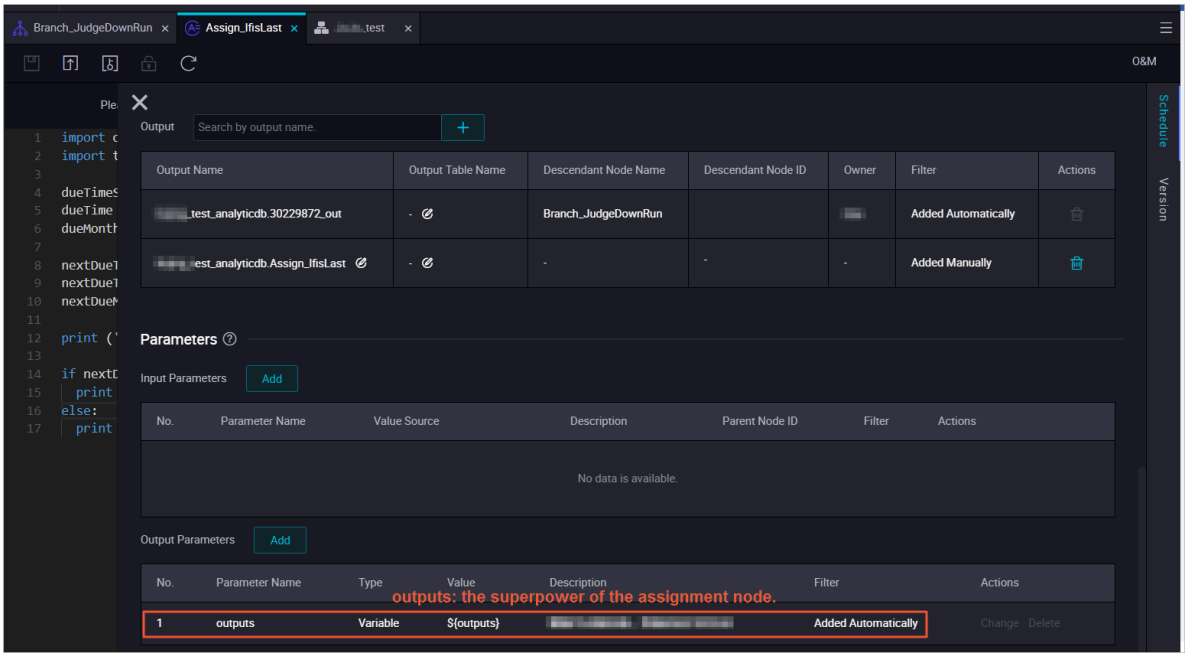
- **For SQL types, DataWorks captures the SQL of the last SELECT statement as the value of outputs.**
- **For SHELL/Python types, DataWorks captures the last line of standard output as the value of outputs.**

**In this article, the Python type is used as the code for the assignment node, and the scheduling properties and code settings are as follows.**

- The code is as follows:



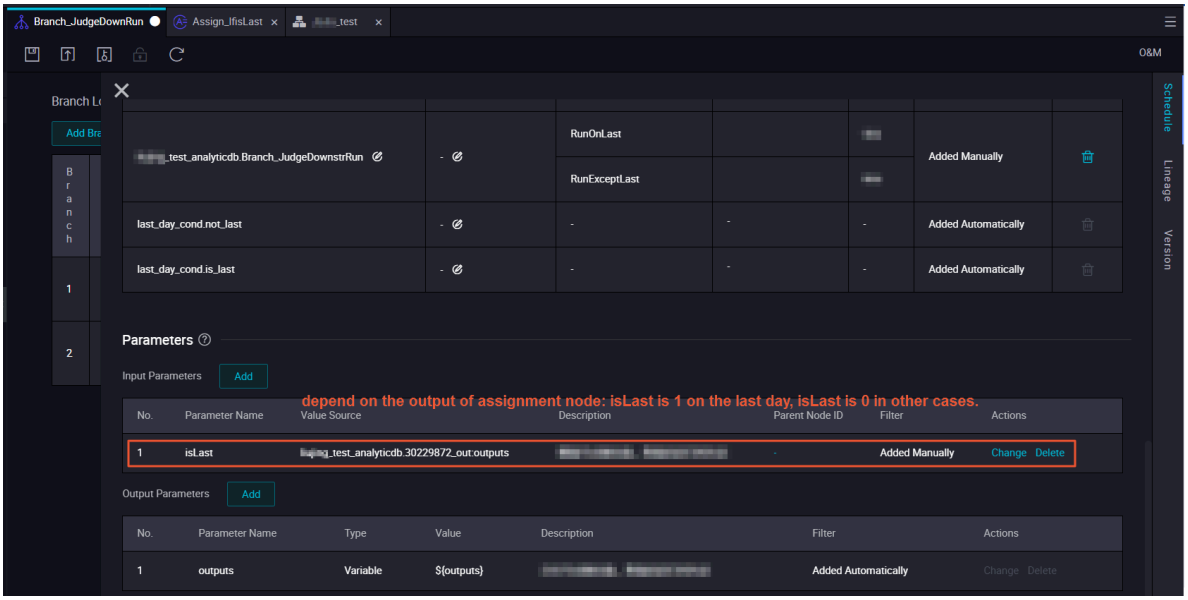

- Schedule configuration



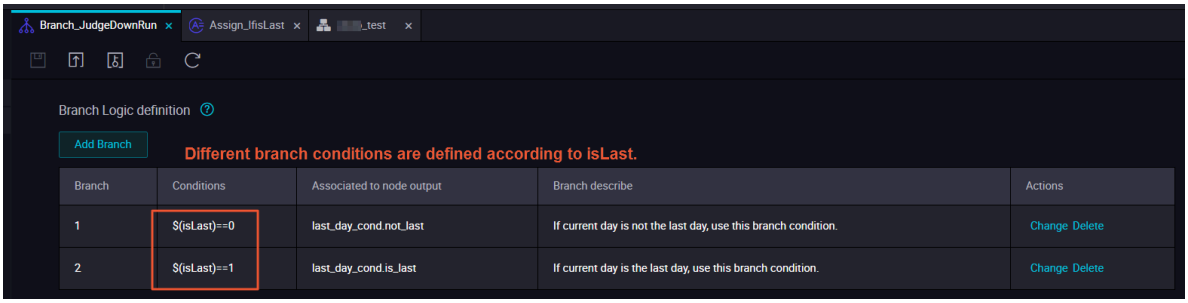

## Define branches

Branch nodes can define conditions with simple Python syntax expressions, each of which is bound to an output. This means that the downstream node under this output is executed when the condition is met, and the other nodes are set to run empty.

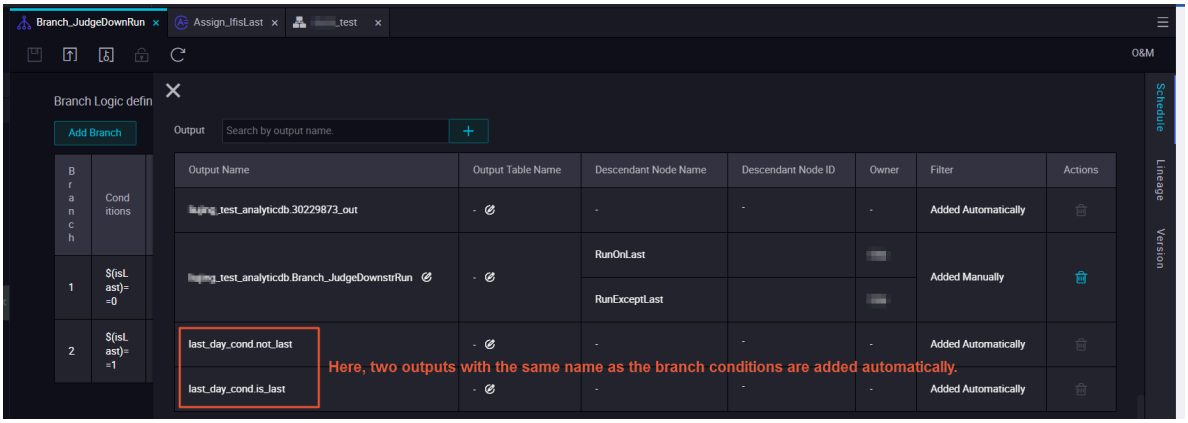
- Schedule configuration




- Branch configuration



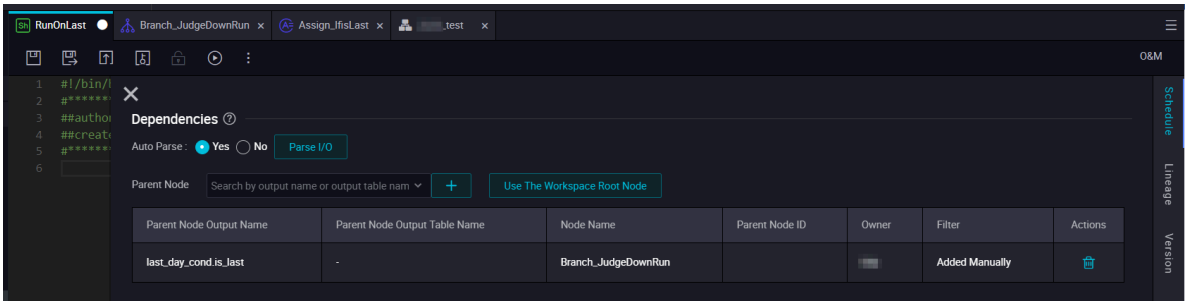

- Schedule configuration generates output of conditional bindings



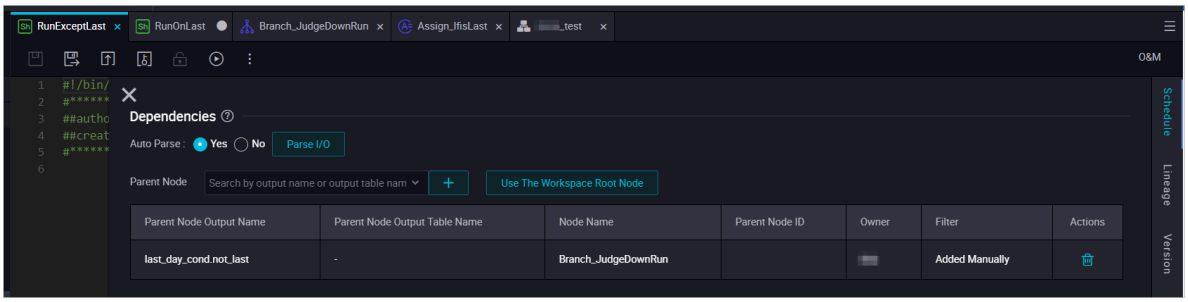

Hang the execution task nodes under different branches

Finally, it is important to note when setting dependencies on the nodes that actually perform tasks: you can see that the branch node already has three outputs, according to the logic of setting dependencies in the past, any one of these three outputs can be regarded as input; however, since the output of the branch node is now associated with the condition, it should be carefully selected.

- Node dependencies performed on the last day of each month




- Node dependencies performed at other times of each month

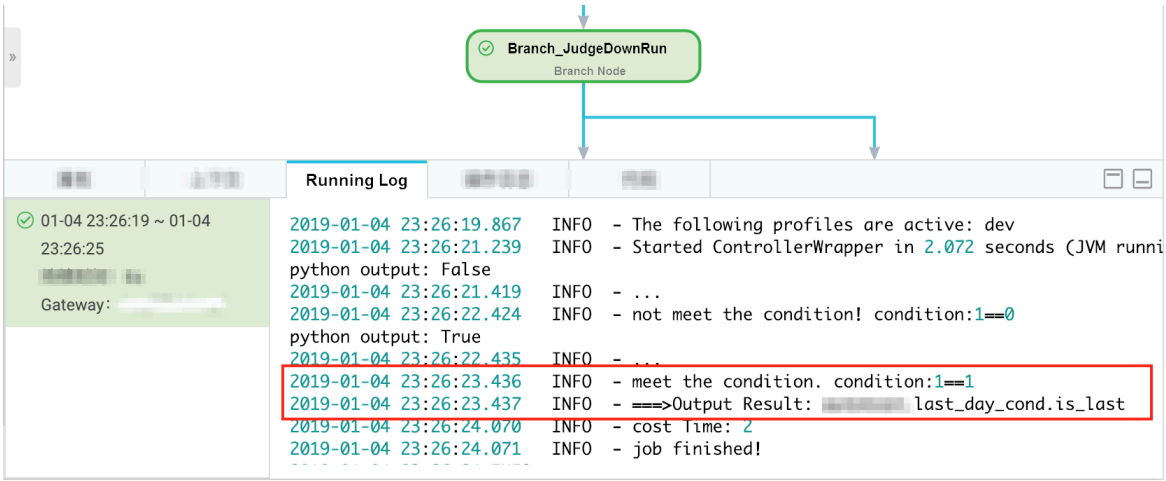



### Result verification

Once completing all of the above configuration, you can submit and publish the task. After publishing, you can perform *patch data* to test the effect: the business date 2018-12-30 and 2018-12-31 are selected , that is, the timing is 2018-12-31 and 2019-01-01 respectively, so that the first batch of patch data should trigger the logic of "last day", the second batch triggers the logic of "non-last day". We look at the difference between the two.

Business date 2018-12-30 (timing 2018-12-31)

- Branch selection results of branch node

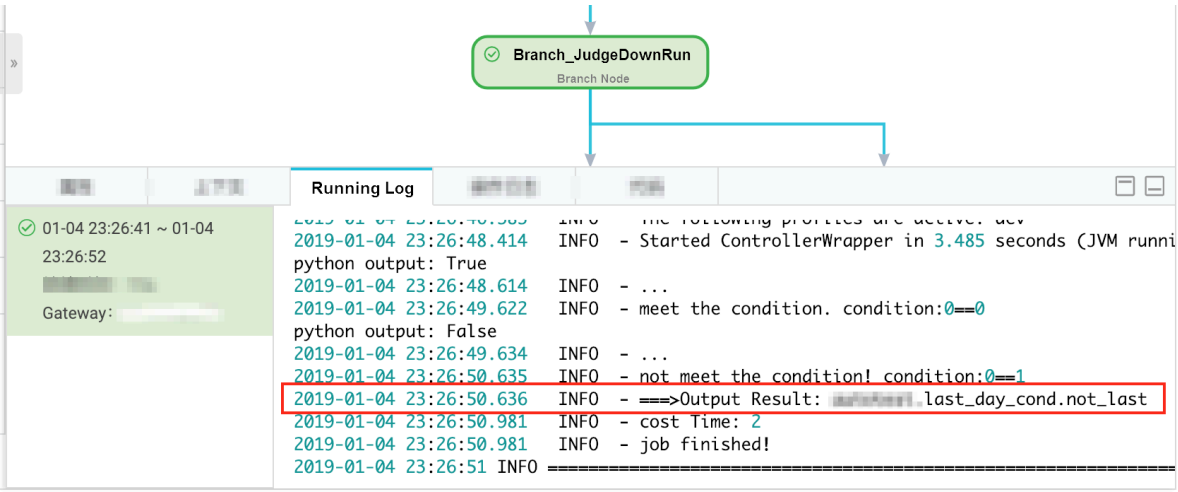



- The node "RunOnLast" is executed normally.

- The node "RunExceptLast" is set to run empty.

Business date 2018-12-31 (timing 2019-01-01)

- Branch selection results of branch node




- The node "RunOnLast" is set to run empty.
- The node "RunExceptLast" is executed normally.

## Summary

Based on the branch node, you have achieved the goal which execute on last day of each month. Of course, this is the easiest way to use a branch node. By using an assignment node with a branch node, you can combine a variety of conditions to meet your business needs.

Finally, the main points of using branch nodes are reviewed.

- DataWorks captures the last SELECT statement or the last line of the standard output stream of the assignment node as the output of an assignment node for downstream references.
- Each output of the branch node is associated with the condition, and the downstream branch node is used as the upstream. It is necessary to understand the meaning of the conditions associated with each output before selecting.
- Unselected branches are set to run empty, and the empty run properties are passed down until the merge node is encountered.
- In addition to blocking the empty run properties, the merge node has more powerful features to wait for your mining.

# 4 Data security

## 4.1 Log on to DataWorks through a specific IP address with a RAM user

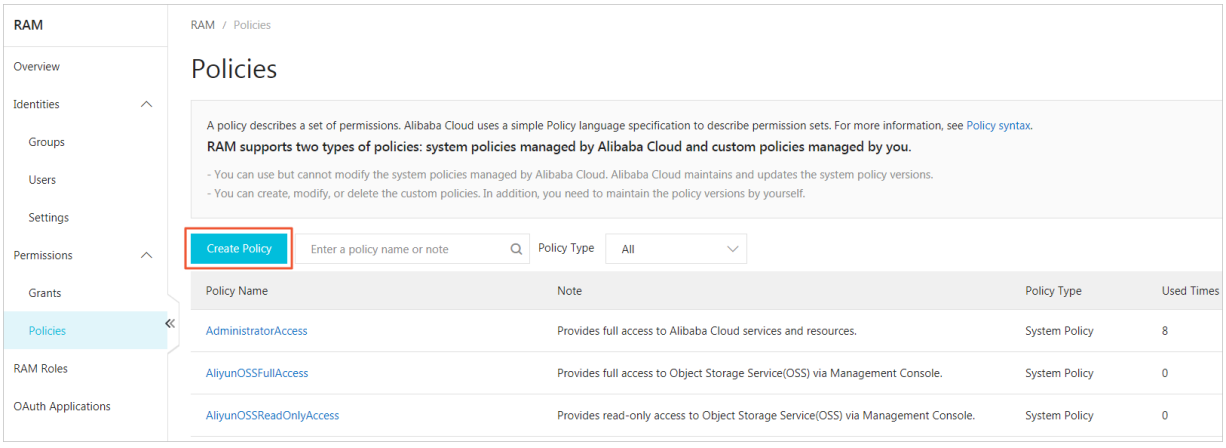
In the process of data development , some users with strict permission control require that the RAM user can only log in through a specific IP address in the company. This article describes how to use a RAM user to log on to DataWorks through a specific local IP address.

### Prerequisites

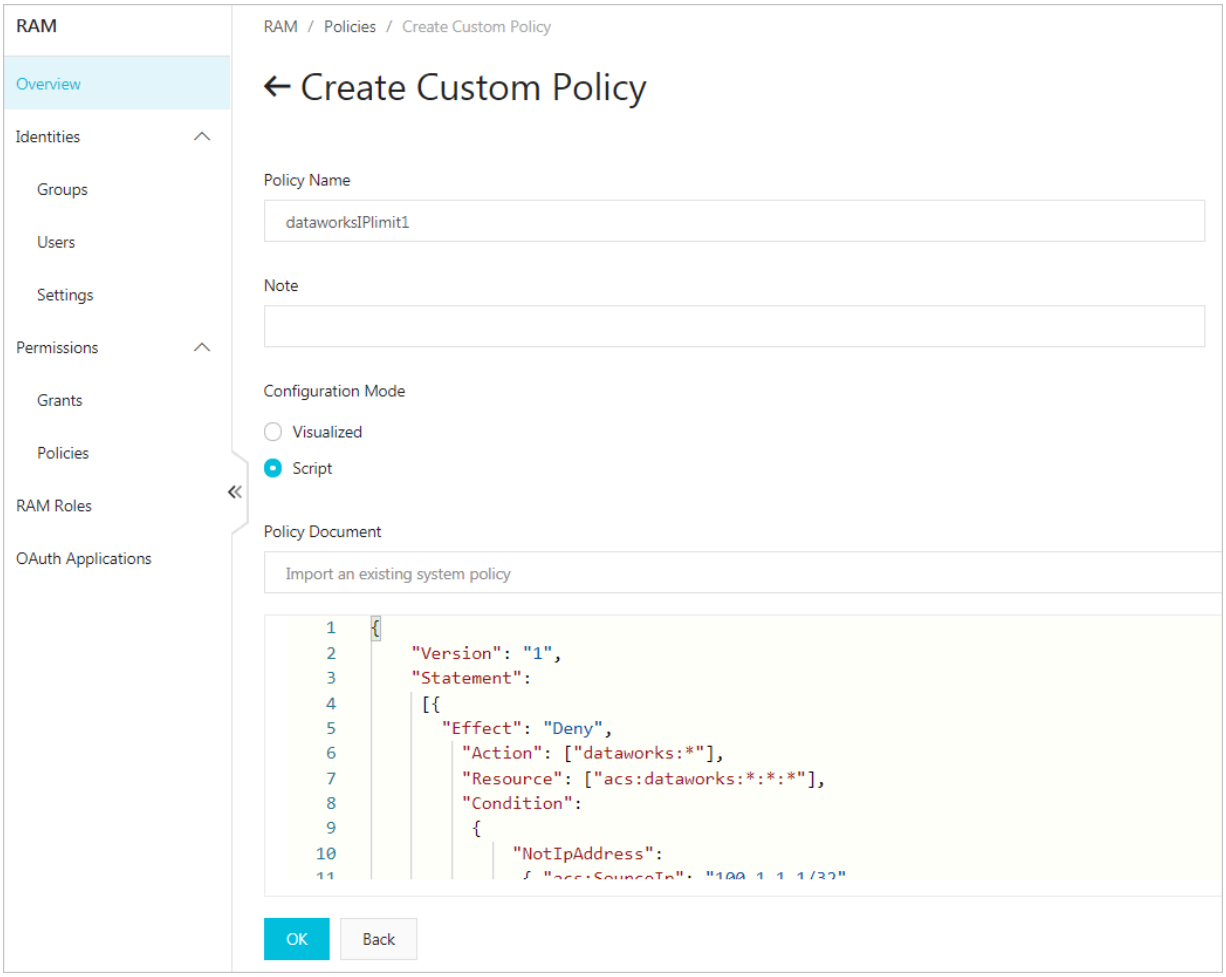
First, you need to refer to *Create a RAM user* to complete the creation of RAM user and authorize it. The permission AliyunDataworksFullAccess is the default system permission and cannot be modified. You need to create an additional custom authorization policy.

### Configure a custom permission policy

Log on to the *RAM console*, click Create Policy in the Policies column to enter the edit page. In this case, the policy name is dataworksIPlimit1.




Select Script for the option of configuration mode, and enter your custom policy.

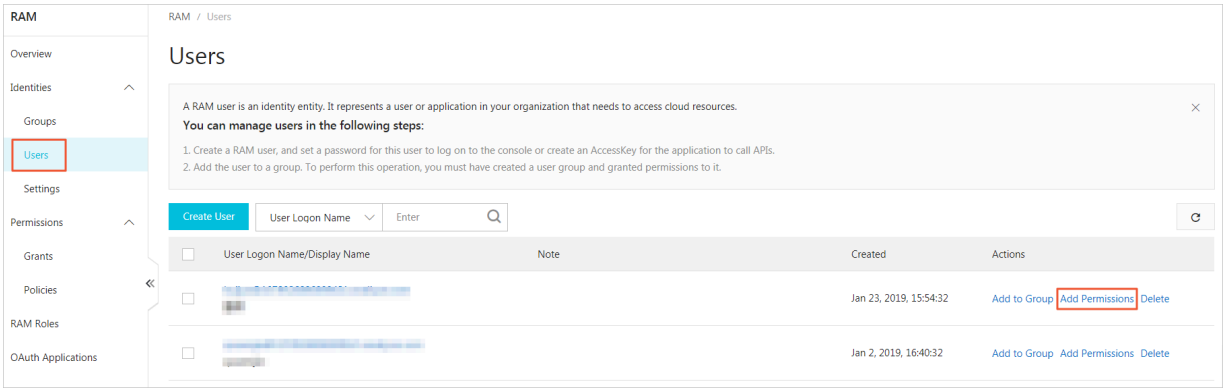



The complete content of the custom policy is shown in the following figure. "acs: SourceIp" is the IP address that you allow access to DataWorks. In this example, it is 100.1.1.1/32. After entering the information, click OK to create the authorization.

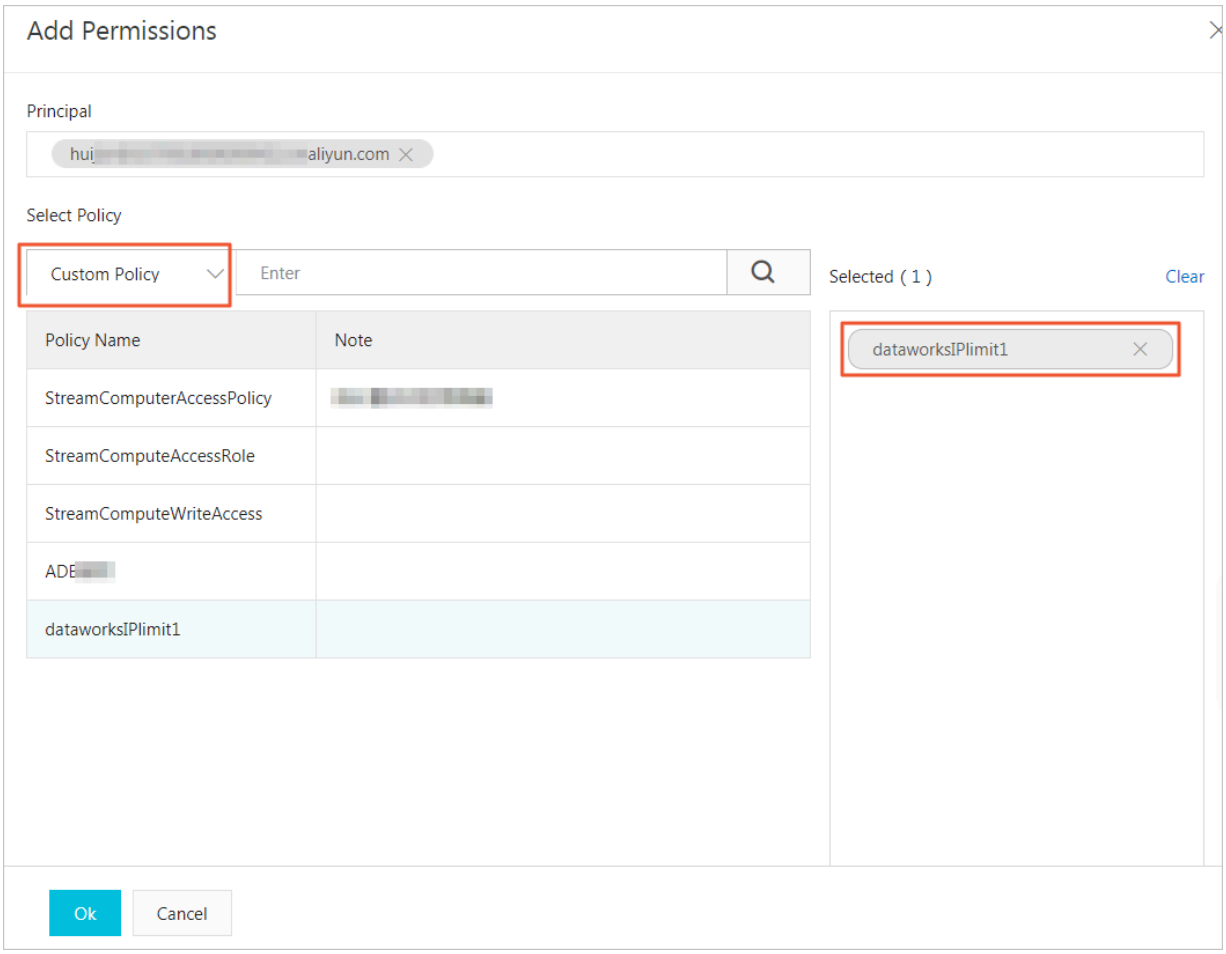
```
{
      " Version ": " 1 ",
      " Statement ":
        [{
          " Effect ": " Deny ",
            " Action ": [" dataworks :*"],
            " Resource ": [" acs : dataworks :*:*:*"],
            " Condition ":
             {
                " NotIpAddre  ss ":
                 {
                    " acs : SourceIp ": " 100 . 1 . 1 . 1 / 32 "
                  }
              }
         }]
}
```

## Add custom permissions

On the RAM console, click Identities > Users, choose the RAM user you want to control, and clickAdd Permissions.

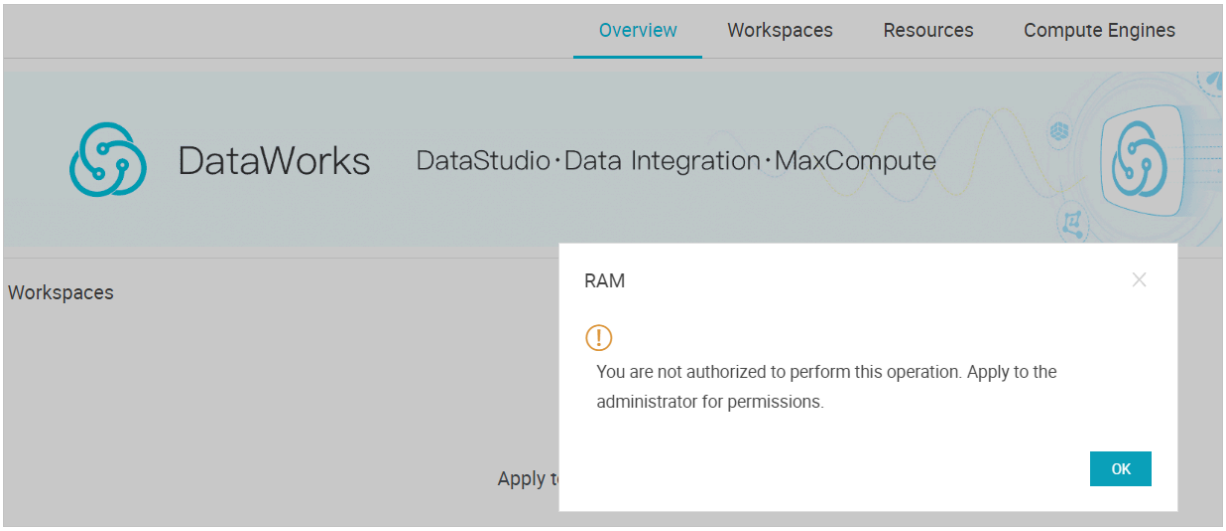



**Select Custom Policy, add the custom policy you just created to the Selected, and clickOK.**




## Verification

**Log on to the DataWorks console using an IP address different from `100 . 1 . 1 . 1 / 32` and find that the login failed.**


Overview
Workspaces
Resources
Compute Engines
DataWorks
DataStudio·Data Integration·MaxCompute
Workspaces
RAM
You are not authorized to perform this operation. Apply to the administrator for permissions.
OK