DataWorks

Quick Start

Quick Start

This guide describes how to quickly perform data development and O&M operations.

> **NOTE:**
>
> If this is the first time you are using DataWorks, make sure that you have prepared an account and configured the project roles and project according to steps in Preparation. Then, go to the DataWorks console page and click **Enter Workspace** after a project to go to the **Data Development** page of DataWorks to start data development.

Generally, DataWorks project space data development and O&M involve the following operations:

Step 1: Create a table and upload data

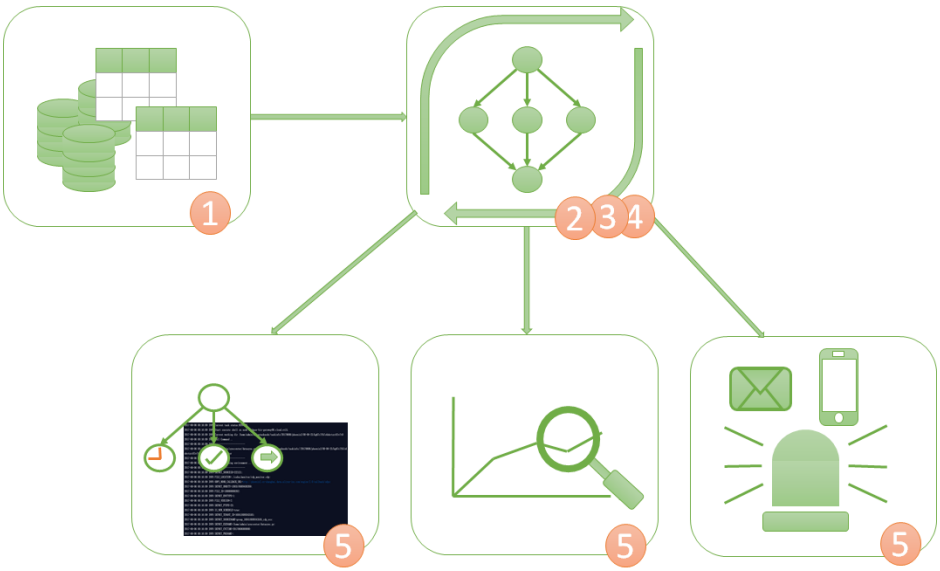
Step 2: Create a flow

Step 3: Create a synchronization task

Step 4: Set the period and dependency

Step 5: Perform O&M and log troubleshooting

A general process is shown in the following figure:



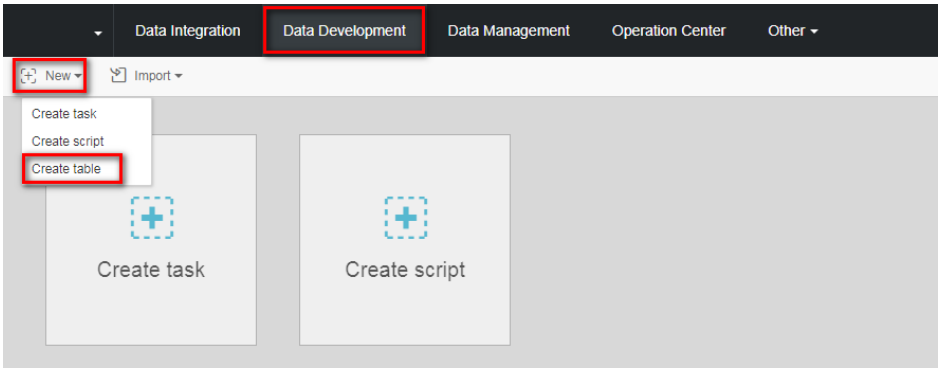
This section uses the creation of the tables bank_data and result_table as an example to describe how to create a table and upload data. The table of bank_data is used to store business data, while the

result_table is used to store results after data analysis.

Instructions

Create bank_data

Log on to the project, and click **Data Development** > **New** > **Create Table**.

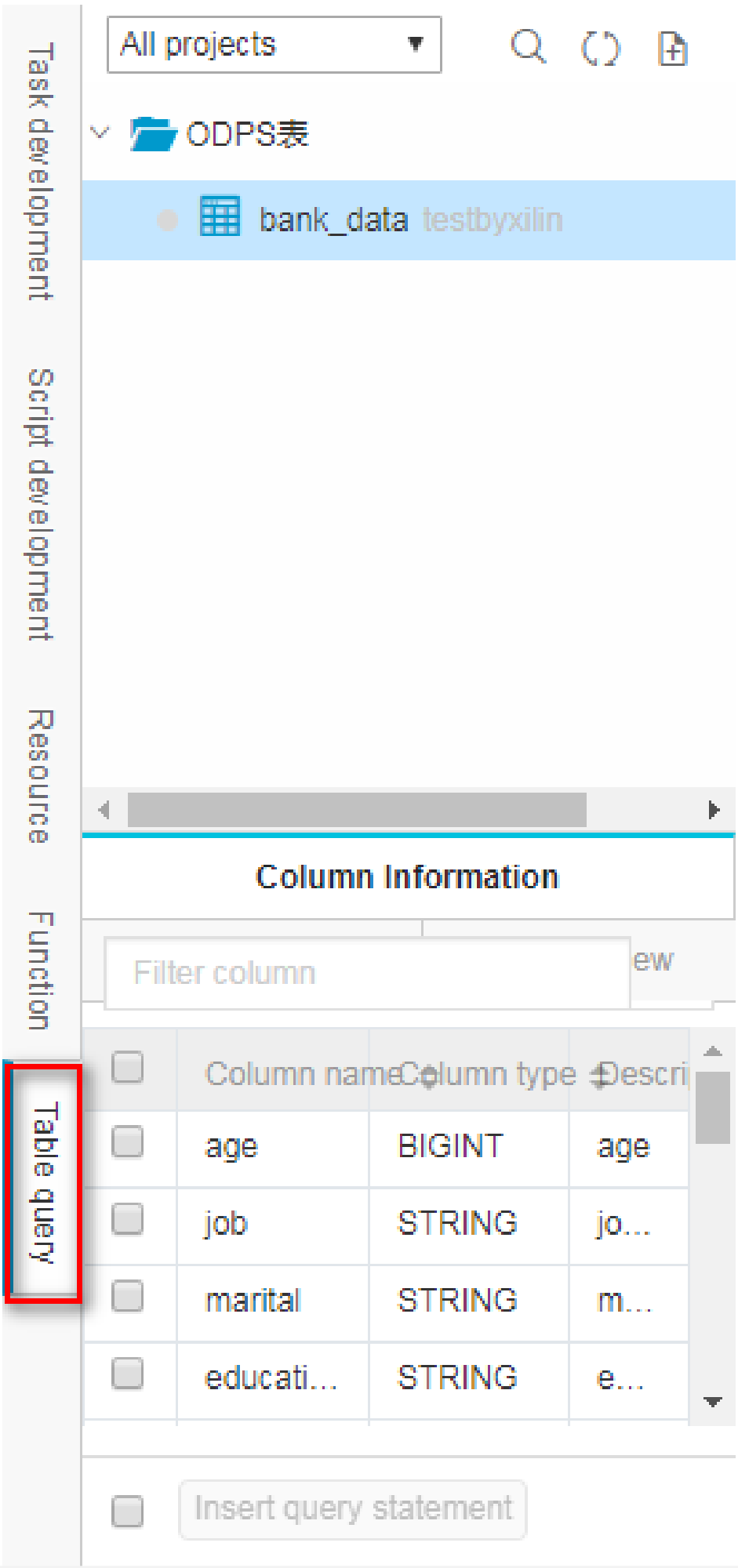


Enter the table creation statements, and click **OK**.For more information on table creation SQL syntax, see MaxCompute-based table creation, view, and deletion.

The statements used for table creation in this example are as follows:

```
CREATE TABLE IF NOT EXISTS bank_data
(
age BIGINT COMMENT 'age',
job STRING COMMENT 'job type',
marital STRING COMMENT 'marital status',
education STRING COMMENT 'educational level',
default STRING COMMENT 'credit card ownership',
housing STRING COMMENT 'mortgage',
loan STRING COMMENT 'loan',
contact STRING COMMENT 'contact information',
month STRING COMMENT 'month',
day_of_week STRING COMMENT 'day of the week',
duration STRING COMMENT 'Duration',
campaign BIGINT COMMENT 'contact times during the campaign',
pdays DOUBLE COMMENT 'time interval from the last contact',
previous DOUBLE COMMENT 'previous contact times with the customer',
poutcome STRING COMMENT 'marketing result',
emp_var_rate DOUBLE COMMENT 'employment change rate',
cons_price_idx DOUBLE COMMENT 'consumer price index',
cons_conf_idx DOUBLE COMMENT 'consumer confidence index',
euribor3m DOUBLE COMMENT 'euro deposit rate',
nr_employed DOUBLE COMMENT 'number of employees',
y BIGINT COMMENT 'has time deposit or not'
);
```

After the table is created, click **Table Query** in the left-side navigation bar and enter the table name for search.


Task development
Script development
Resource
Function
Table query
All projects
ODPS表
bank_data testbyxilin
Column Information
Filter column
ew
Column name
Column type
Descri
age
BIGINT
age
job
STRING
jo...
marital
STRING
m...
educati...
STRING
e...
Insert query statement

Create result_table

Click **Data Development** > **New** > **Create Table**.

On the Create Table page, enter the table creation statements, and click **OK**. The statements used for table creation are as follows:

```
CREATE TABLE IF NOT EXISTS result_table
(
education STRING COMMENT 'educational level',
num BIGINT COMMENT 'number of people'
);
```

After the table is created, click **Table Query** in the left-side navigation bar and enter the table name for search.

Upload local data to bank_data

DataWorks supports the following operations:

- Upload data in local text files to a table in the workspace.
- Use the data integration module to import business data from multiple different data sources to the workspace.

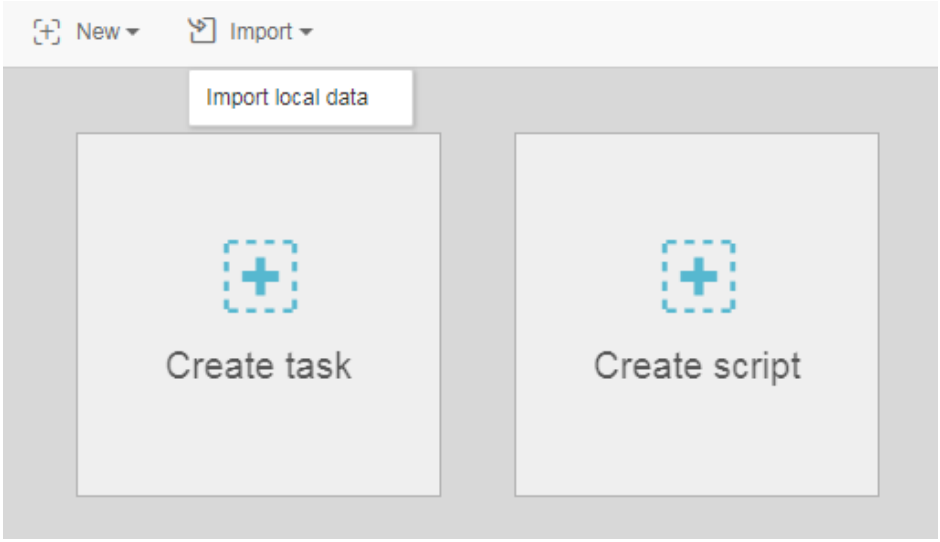
NOTE:

In this section, local files are used as the data source. Local text file uploads have the following restrictions:

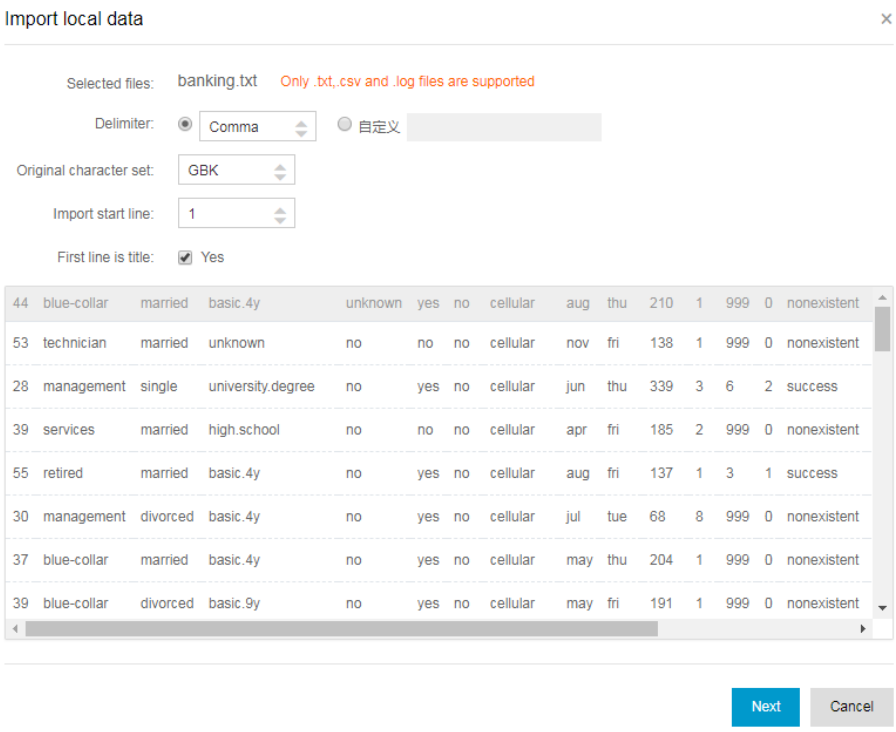
- File type: Only .txt and .csv files are supported.
- File size: The file size cannot exceed 10 MB.
- Operation objects: Partition and non-partition tables can be imported, but Chinese partition values are not supported.

Using the import of the local file banking.txt to DataWorks as an example, the instruction is as follows:

Click **Import** > **Import Local Data**.

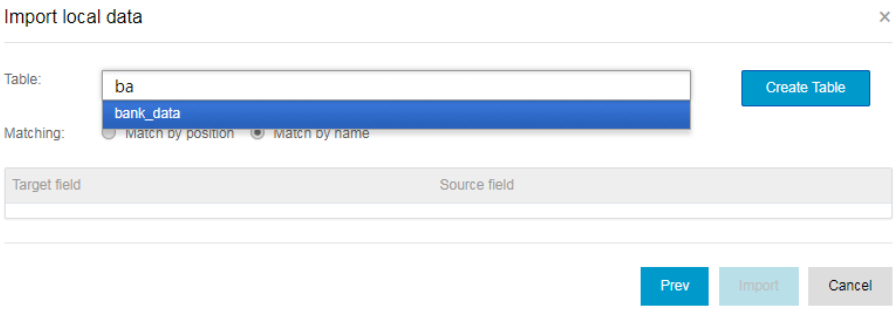


Select a local data file, configure the import information, and click **Next**.

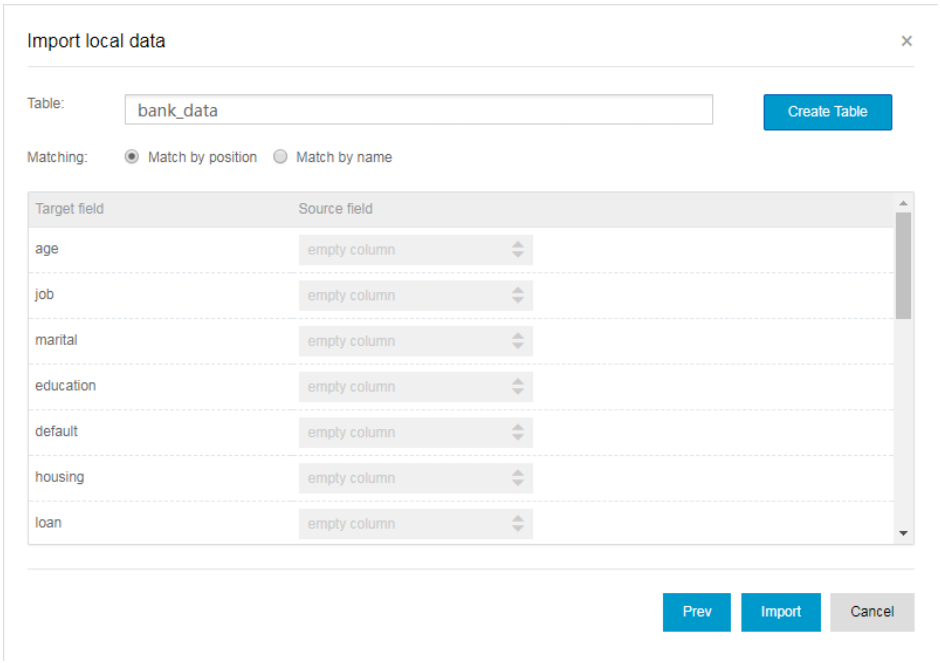


Enter at least two letters to search for the table by name. Select the table to which the data is to be imported, for example, bank_data.

To create a new table, click **Create Table**.



Select the field matching method (“Match by Position” is used in this example), and click **Import**.



After the file is imported, the system displays a data import success or failure prompt.

Other data import methods

Create a data synchronization task

Applicability:

Data saved in multiple source types, including RDS, MySQL, SQL Server, PostgreSQL, MaxCompute, OCS, DRDS, OSS, Oracle, FTP, dm, HDFS, and MongoDB.

For information on DataWorks data synchronization task creation, see Create a data synchronization task.

Upload a local file

Applicability:

The file size cannot exceed 10 MB, and only .txt and .csv files are supported. Only non-partitioned tables are supported.

For information on DataWorks local file uploads, see the **Upload local data to bank_data** section.

Use Tunnel commands to upload files

Applicability:

Local files and other resource files are larger than 10 MB.

Using the Tunnel commands provided by the MaxCompute Client to upload or download data, you can upload a local data file to a partitioned table.

For details, see Tunnel command operations.

Use DataX open-source tools

Applicability:

DataX is applicable to the import of local data in batches. The imported data must have a two-dimensional table structure. This method can be used in other scenarios that described above. For details, see DataX.

For more information about DataX open-source tools, go to the DataX open-source website.

Subsequent steps

You have learned how to create a table and upload data. You can go to the next tutorial for further study. This tutorial shows you how to create a flow for further data analysis and computing in the project space. For details, see Create a flow for data analysis.

The data development function of DataWorks supports the graphic design of data analysis flows and processes data and forms mutual dependencies through flow tasks and inner nodes. Currently, it supports multiple task types, including ODPS_SQL, data synchronization, OPEN_MR, SHELL, machine learning, and virtual nodes. For details about the use of each task type, see Task type description.

This section uses the creation of a flow task named “work” as an example to show how to create nodes in a flow, configure dependencies, and conveniently design and display steps and sequences for data analysis. We briefly describe how to use the data development function for further data analysis and computing in the workspace.

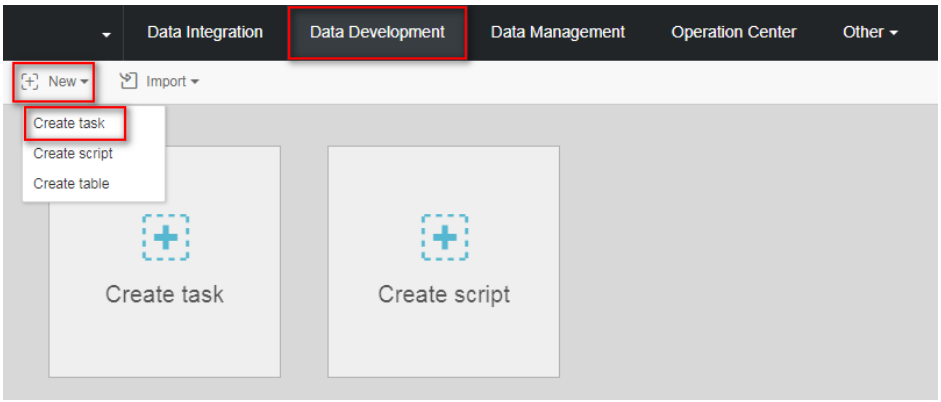
Prerequisites

You have prepared the business data table bank_data, the data it contains, and the result_table in the workspace according to Create a table and upload data.

Instruction

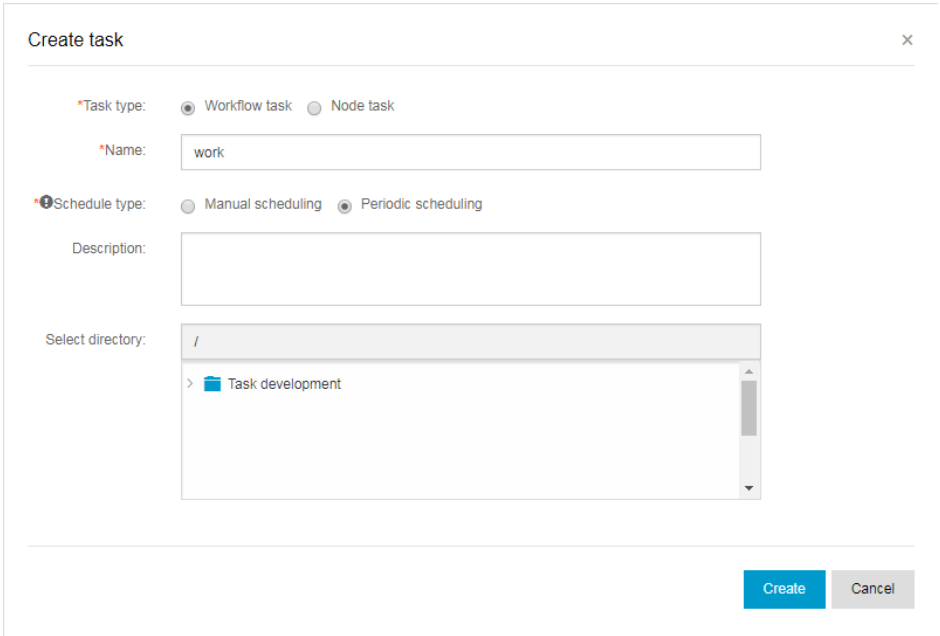
Create a flow

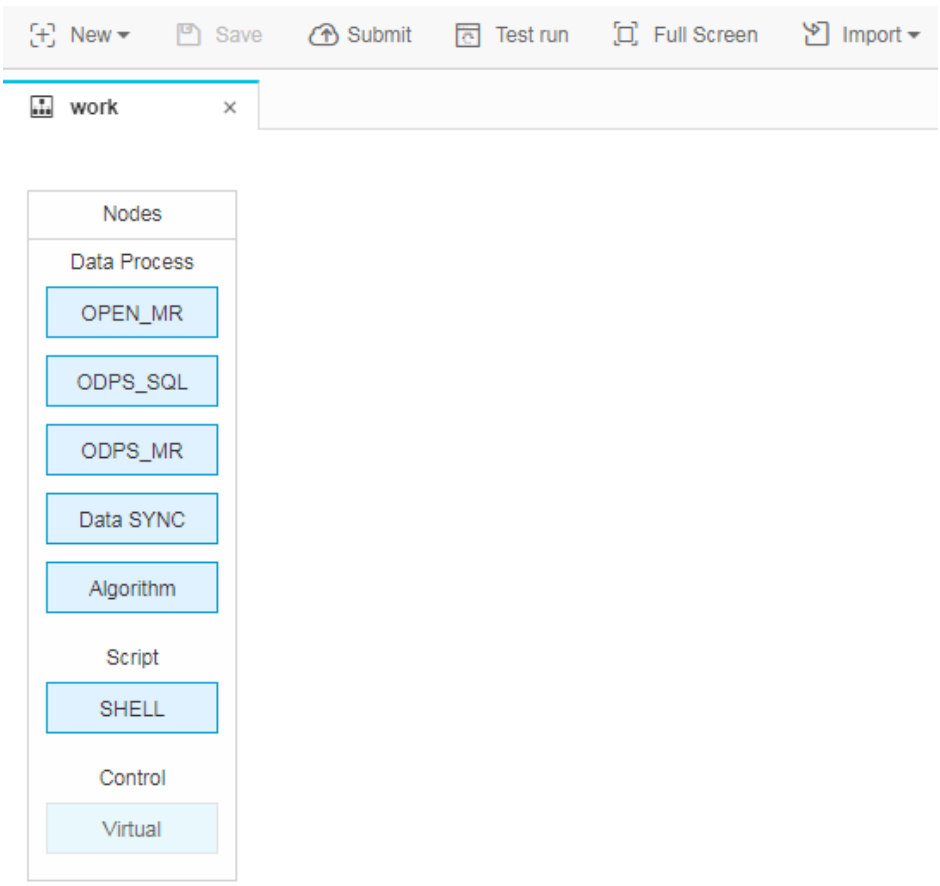
log on to the project, and click **Data Development** > **New** > **Create Task**.



Select the relevant content in the dialog box and specify the task type as **Flow task**.

Note：Once selected, the scheduling attribute cannot be changed.





Create a node and dependency on the flow canvas

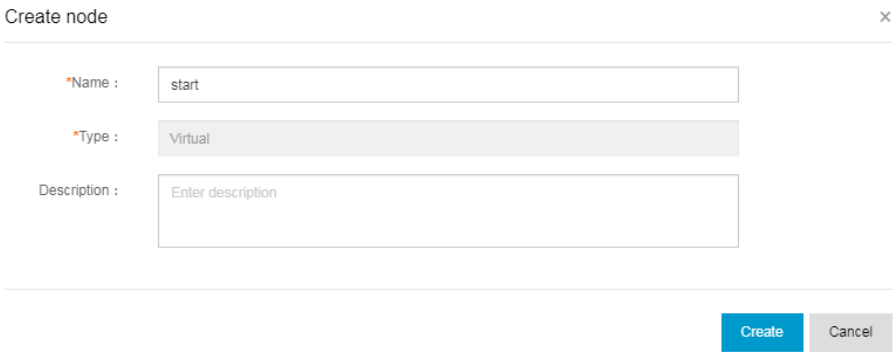
This section shows how to create a virtual node “start” and an odps_sql node “insert_data”, and to configure “insert_data” to depend on “start”.

Note：

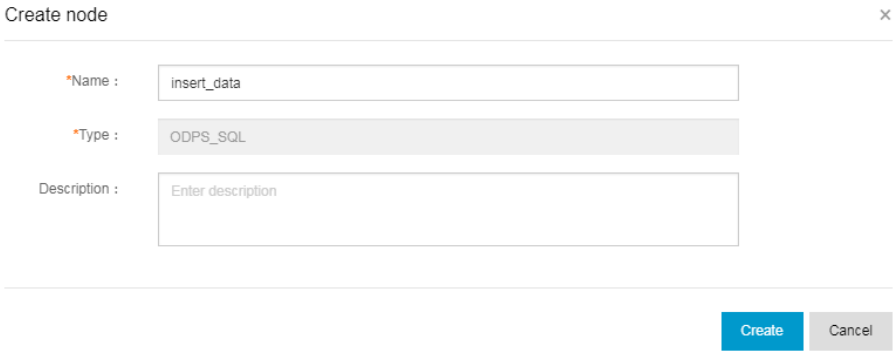
- As a control-type node, the virtual node does not affect the data during flow operation and is only used for O&M control of downstream nodes.
- When a virtual node is depended on by other nodes and its status is manually set to failure by the O&M personnel, its downstream nodes that have not yet run cannot be triggered. This prevents further propagation of erroneous upstream data during the O&M process. For details, see the section on virtual nodes in Task type description.

In summary, we recommend that you create a virtual node as the root node to control the whole flow when designing a flow.

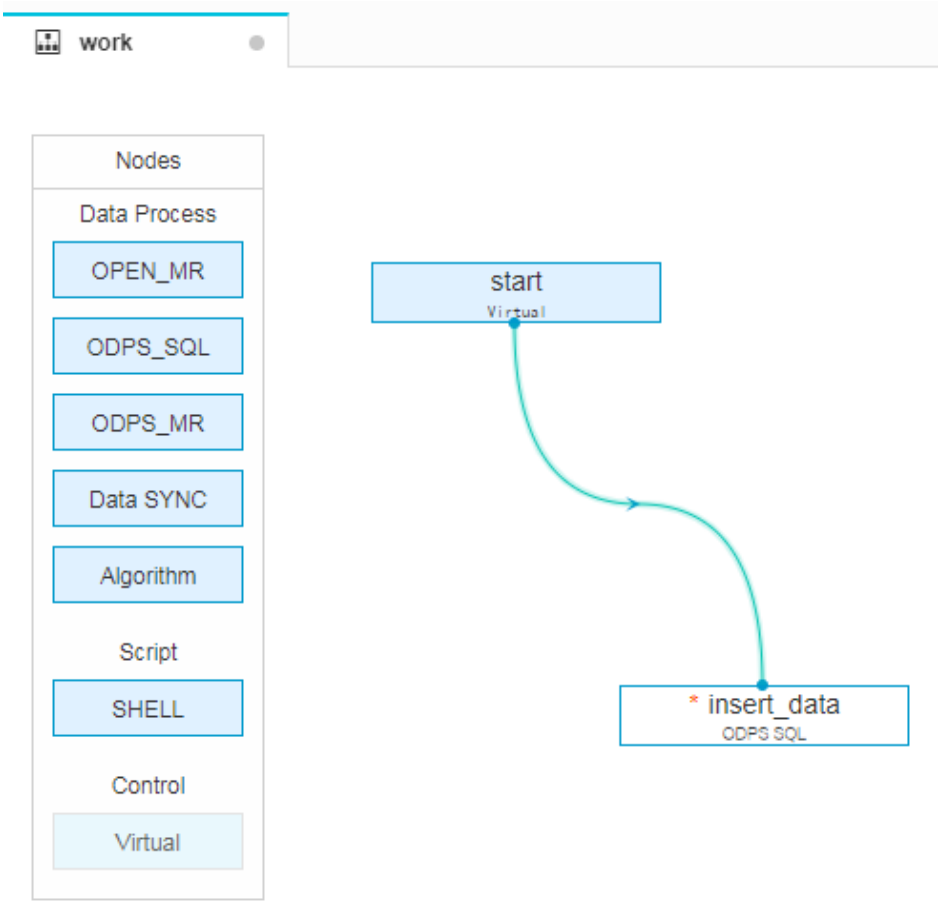
Double-click the virtual node, and enter the node name” start”.



Double-click **ODPS_SQL** and enter the node name “insert_data”.



Click the start note, and draw a line between start and insert_data to make insert_data dependent on start.



Edit the code in ODPS_SQL

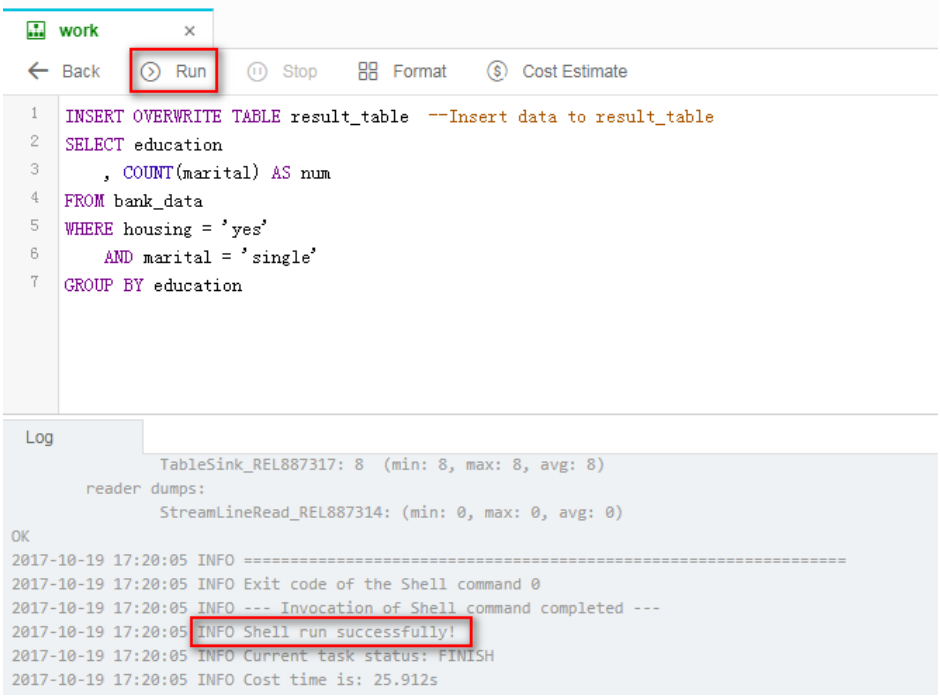
This section describes how to use the SQL code in the ODPS_SQL node **insert_data** to query the quantity of mortgages for individual persons with different education backgrounds, and save the results for analysis or display by subsequent nodes. The SQL statements are as follows. For details about the syntax, see the MaxCompute documentation.

```
INSERT OVERWRITE TABLE result_table  --Insert data to result_table
SELECT education
, COUNT(marital) AS num
FROM bank_data
WHERE housing = 'yes'
AND marital = 'single'
GROUP BY education
```

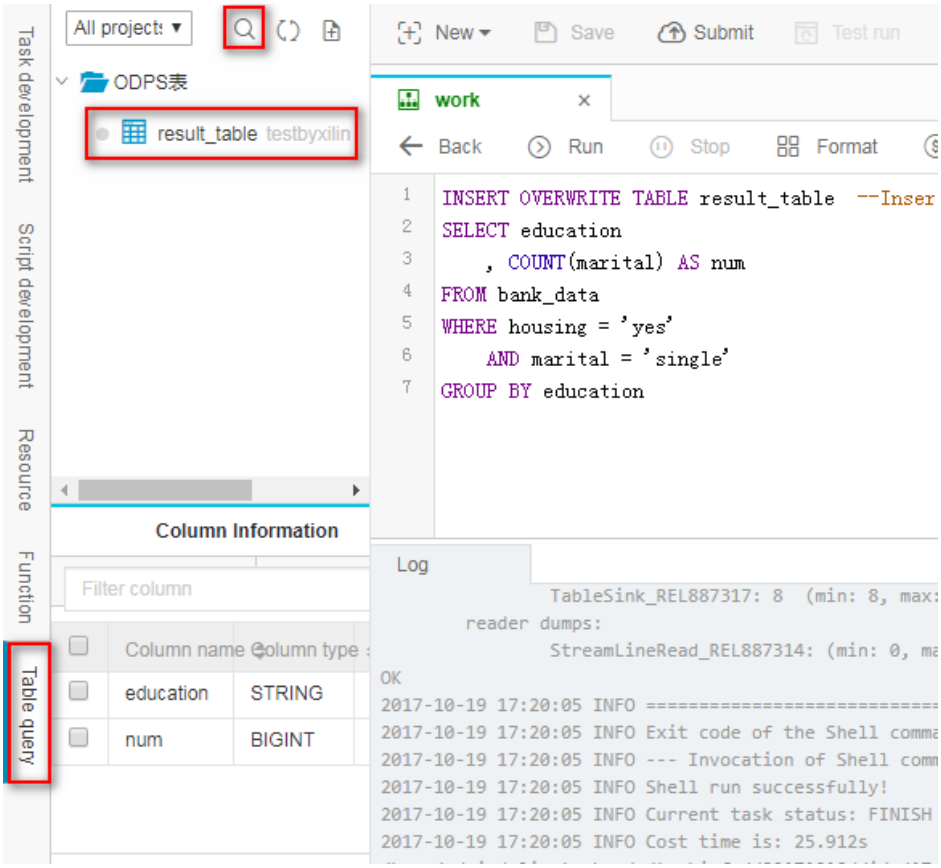
Run and debug ODPS_SQL

After editing the SQL statements in the insert_data node, click **Save** to prevent code loss.

Click **Run** to view the operations logs and results.

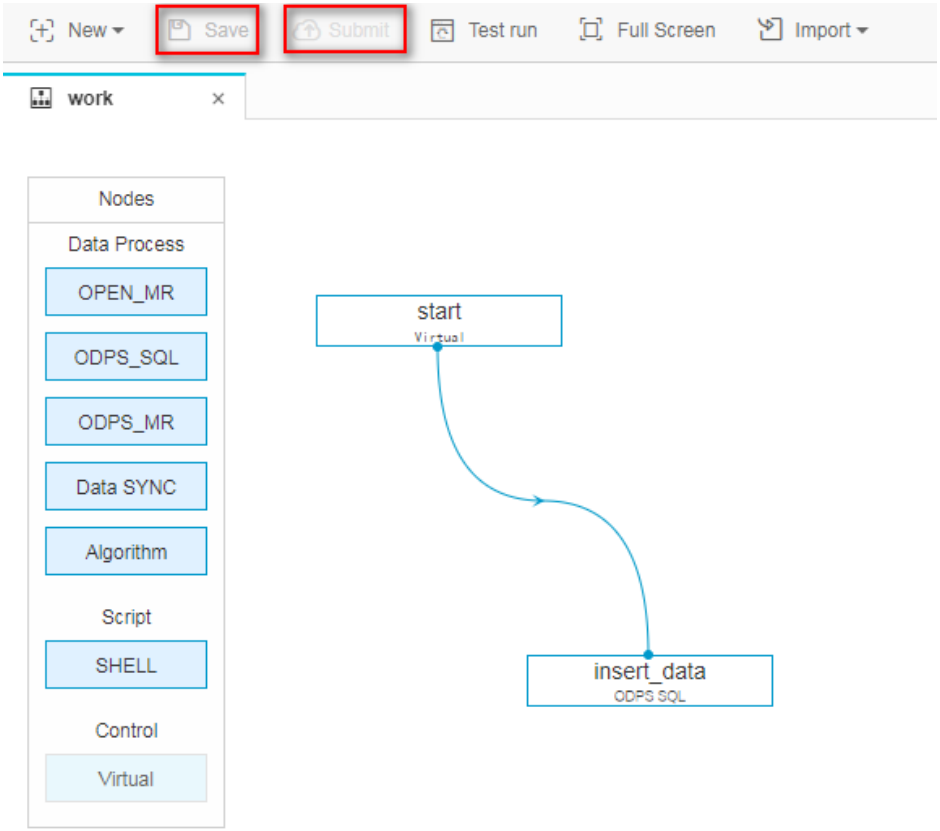


Then, click **Table Query** on the left to query data in the table.



Save and submit a flow

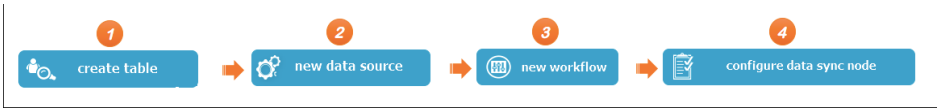
After running and debugging the ODPS_SQL node “insert_data”, return to the flow page, and save and submit the whole flow.



Subsequent steps

Now, you know how to create, save, and submit a flow. Continue to the next tutorial for further study. This tutorial shows you how to create a synchronization task to export data to data sources of different types. For details, see Create a synchronization task to export results.

Currently data source types supported by the data synchronization jobs include: **MaxCompute, RDS (MySQL, SQL Server, PostgreSQL), Oracle, FTP, ADS, OSS, OCS, and DRDS**.



Taking the data synchronization from RDS to MaxCompute for example, the detailed descriptions can be found below:

Step 1: Create a data table

For details on how to create a MaxCompute table, refer to **Create a Table**.

Step 2: Create a data source

> **Note:**
>
> Only users with the project administrator role are allowed to create a new data source.

Preparation

Currently only the China East 1 (Hangzhou) region is supported as a RDS data source, and the Beijing region is not yet supported. In addition, when the RDS data sources in the Hangzhou region cannot be connected to during testing, you should add a whitelist of data synchronization server IP addresses onto your RDS:

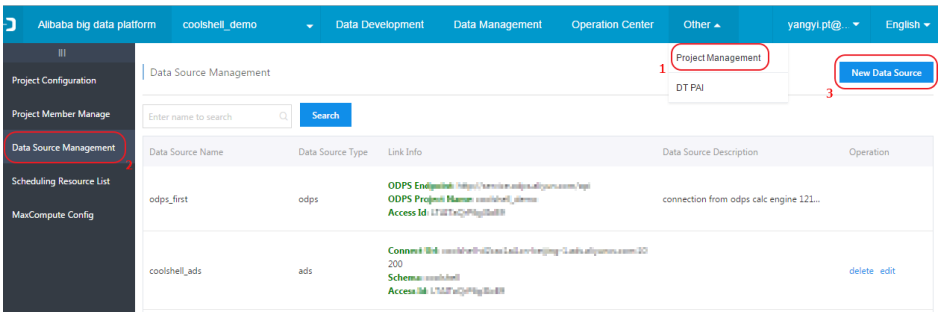
10.152.69.0/24,10.153.136.0/24,10.143.32.0/24,120.27.160.26,10.46.67.156,120.27.160.81,10.46.64.81,121.43.110.160,10.117.39.238,121.43.112.137,10.117.28.203,118.178.84.74,10.27.63.41,118.178.56.228,10.27.63.60,118.178.59.233,10.27.63.38,118.178.142.154,10.27.63.15

The specific steps are as follows:

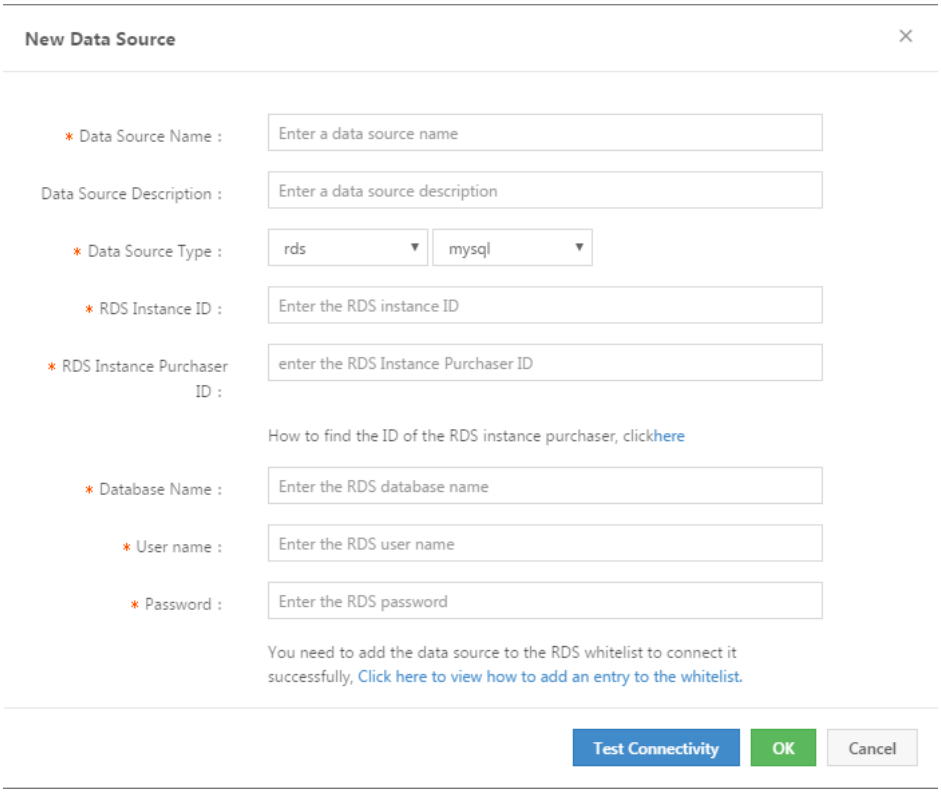
Go to **Alibaba Cloud Dataplus platform > DataWorks Kit > Console** as a developer, click the **Enter Work Zone** in the action bar of the corresponding project.

Click **Manage Projects** in the top menu bar, and then click **Manage Data Sources** in the left navigation bar.

Click **New Data Source**.



Fill in the configuration items in the “New Data Source” pop-up box.



Specific descriptions of the configuration items in the figure above are as follows:

- Data source name: A data source name may consist of letters, numbers, and underscores. It must begin with a letter or an underscore and cannot exceed 30 characters in length.
- Data source descriptions: A brief description of the data source. The description should not exceed 1,024 characters in length.
- Data source type: The data source type selected currently (RDS>MySQL>RDS).
- RDS instance ID: The ID of the MySQL data source RDS instance.
- RDS instance purchaser ID: The purchaser ID of the MySQL data source RDS instance.

Note: If you have selected the JDBC form to configure the data source, the format of the JDBC connection information is: jdbc:mysql://IP:Port/database.

- Database name: The database name of the data source.
- User name/password: The user name and password of the database.

Click **Test Connectivity**.

If the test result is connected successfully, click the **Save** button to save the configuration information.

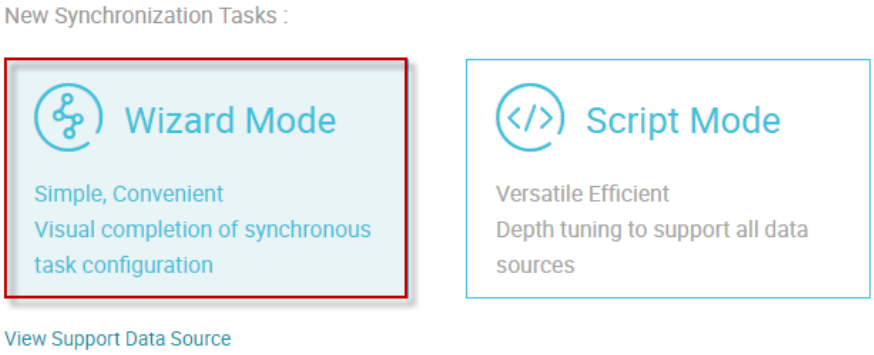
For detailed configurations of other types of data sources (MaxCompute, RDS, Oracle, FTP, ADS, OSS, OCS, and DRDS), see Data Source Configuration.

Step 3: Create a new job

Take the "wizard mode" new task as an example.

1. In the **data integration** interface, click on the left navigation bar to synchronize tasks;

   Click the **wizard mode** in the interface to get to the task configuration page.

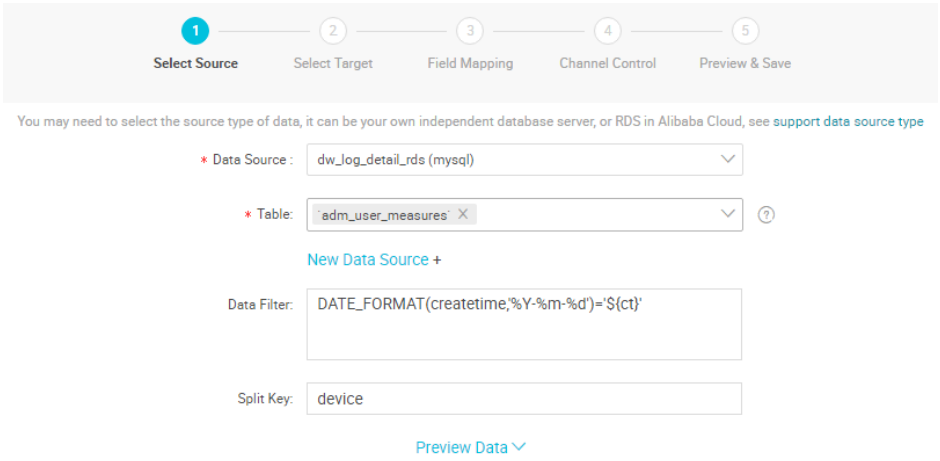


Step 4: Configure the data synchronization job

The synchronization job node includes five configuration items: **"Select Data Source and Target"**, **"Field Mapping"**, **"Channel Control"** and **Preview & Save**.

Select the data source

Select **Data Source** (The data source has been created in Step 2), and then select the data table.



Extraction Filtering: You can specify the WHERE filter based on the corresponding SQL syntax (You do not need to specify the WHERE keyword). The WHERE filter will be used as a condition of incremental synchronization.

The WHERE filter is used for source data filtering. The specified column, table, and WHERE filter are concatenated to create an SQL command for data extraction. The WHERE filter can be used for full synchronization and incremental synchronization. Specific descriptions are as follows:

- Full synchronization:
  Full synchronization is usually executed when data is imported for the first time. You do not need to configure the WHERE filter. You can set the WHERE filter limit to 10 to avoid a large data size during tests.
- Incremental synchronization:
  In the actual service scenario, incremental synchronization usually synchronizes the data generated on the current day. Before compiling the WHERE filter, you usually need to first determine the field that describes the increment (timestamp) in the table. For example, if in Table A, the field that describes the increment is “creat_time”, you need to compile “creat_time>$yesterday” in the WHERE filter and assign a value to the parameter in parameter configuration.

Splitting key: If the data synchronization job is RDS/Oracle/MaxCompute, the splitting key configuration will be displayed on the page.
only supports integer fields. During data reading, the data will be split based on the configured fields to achieve concurrent reading, improving data synchronization efficiency. The splitting key configuration item will only be displayed when the synchronization job is for importing RDS/Oracle data into MaxCompute.

If the source is a MySQL data source, the data synchronization job also supports database- and table-based data importing (on a premise that the table structure must be consistent, no matter whether the data is stored in the same database or different databases).

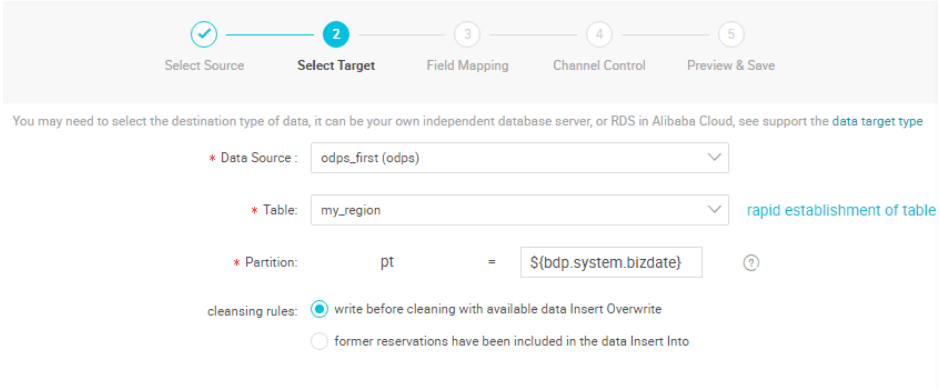
Database- and table-based data importing supports the following scenarios:

Multiple tables in the same database: Click **Search Table** to search for the tables and add the tables you want to synchronize.

Multiple tables in different databases: First click **Add** to select the source database, and then click **Search Table** to add the tables.

Select the data target

Click **rapid establishment of table** and you will be able to convert the tabulation statements of the source table to DDL statements conforming to the MaxCompute SQL syntax to create a target table. After making the necessary selections, click **Next**.



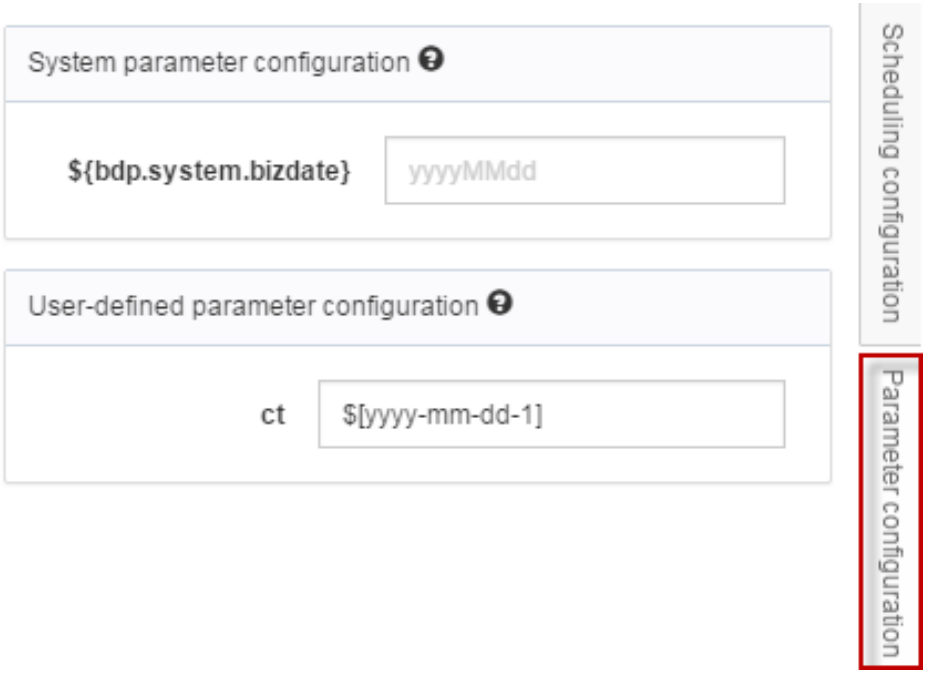
Partition information: Partitioning helps you to easily search for the special columns introduced by some data. By specifying the partition, you can quickly locate the desired data. Constant partitions and variable partitions are supported.

Clearing rules:

Clear existing data before writing: Before data importing, all the data in the table or partition should be cleared, which is equivalent to **Insert Overwrite**.

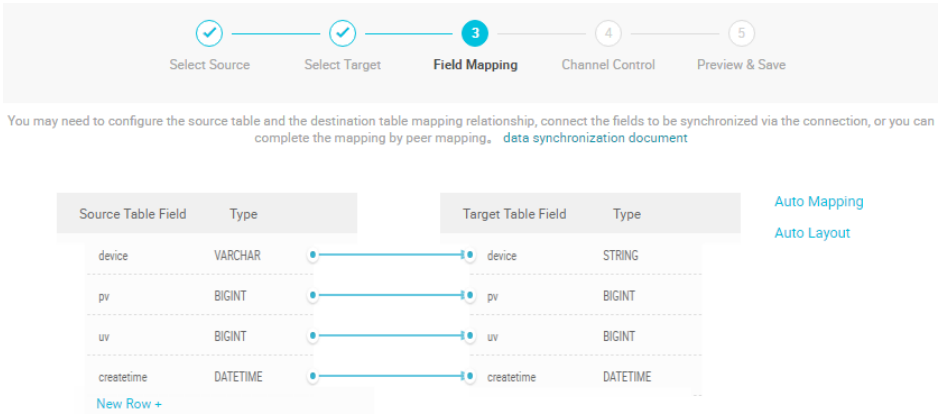
Keep existing data before writing: No data needs to be cleared before data importing. New data is always appended with each run, which is equivalent to **Insert into**.

Assign values to parameters in the parameter configuration, as shown in the figure below:



Field Mapping

You need to configure the field mapping relationships. The **Source Table Fields** on the left correspond one to one with the **Target Table Fields** on the right.

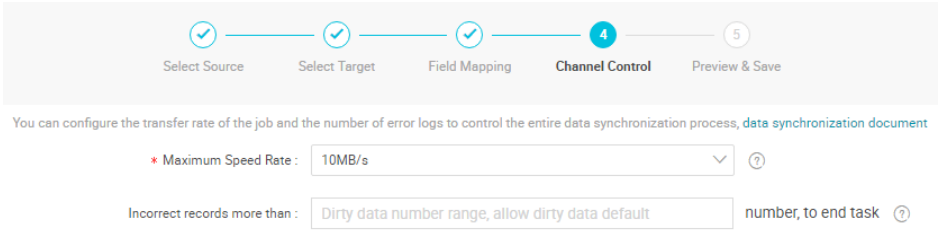


- Add/Delete: Click **Add a Line** to add a single field. Move and hover the cursor on a line above, and click the **Delete** icon, and you will delete the current field.

Writing method for user-defined variables and constants:To import a constant or variable to a field in the MaxCompute table, you only need to click the **Insert** button and enter the value of the constant or variable enclosed in single quotation marks. For example, for the '${yesterday}' variable, you can then assign a value to the variable using the parameter configuration component, such as yesterday=$[yyyymmdd].

Channel Control

The **Channel Control** is used to configure the maximum speed of the job and the dirty data check rules, as shown in the figure:



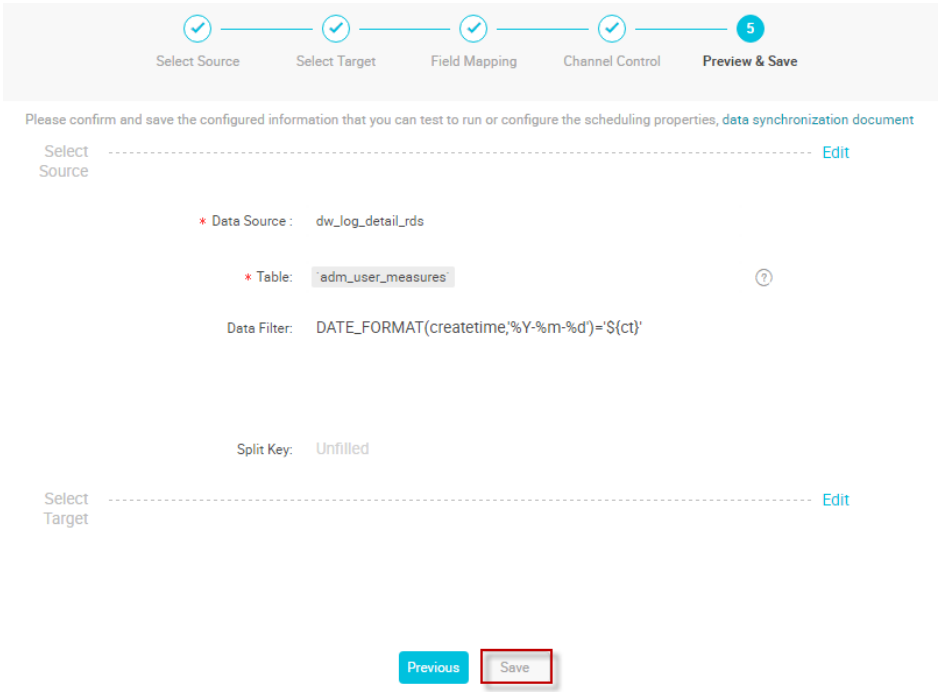
- The maximum speed of the job refers to the speed of the current data synchronization job, with a maximum value of 10 MB/s supported (The channel traffic measured value is the measured value of the data synchronization job, and does not represent the actual traffic of the network interface card).

Dirty data check rules (available for writing data to RDS and Oracle):

- When the number of error records (that is the volume of dirty data) exceeds the configured quantity, the data synchronization job ends.

Preview & Save

When you complete the above configuration, click **next** to preview, if correct, click **save**, as shown below:

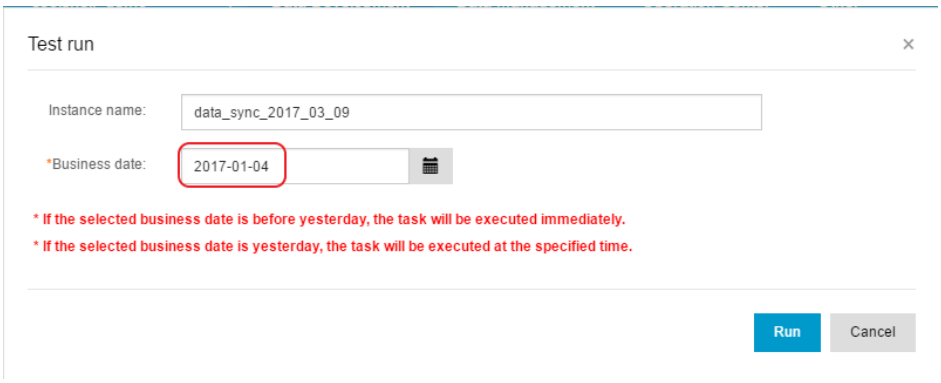


Step 5: Submit the data synchronization job and test the workflow

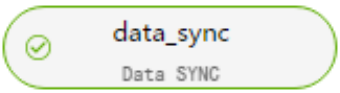
Click the top menu bar to submit the job.

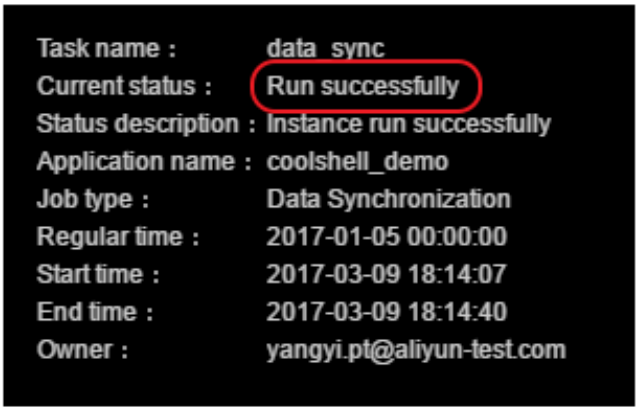
After the job is submitted successfully, click Test Run.

> Because some createtime values in the source table in this example are 2017-01-04, while the scheduling time parameters used in the configuration are $[yyyy-mm-dd-1] and ${bdp.system.bizdate}, we set the partition value of the target table to 20170104 to assign the value of 2017-01-04 to the createtime parameter in the test. The 2017-01-04 should be selected as the business time in the test, as shown in the figure below:



After the test task is triggered successfully, you can click **Go to O&M Center** to view the task progress.





View the synchronized data.

```
1  read my_region ;
```

Log | Results[1]

No.	device	pv	uv	createtime	pt
1	android	937	73	2017-01-04 20:51	20170104
2	iphone	428	31	2017-01-04 20:49	20170104
3	macintosh	830	107	2017-01-04 20:51	20170104
4	unknown	4124	444	2017-01-04 20:51	20170104
5	windows_pc	5650	649	2017-01-04 20:51	20170104

DataWorks provides powerful scheduling capabilities including time-based or dependency-based

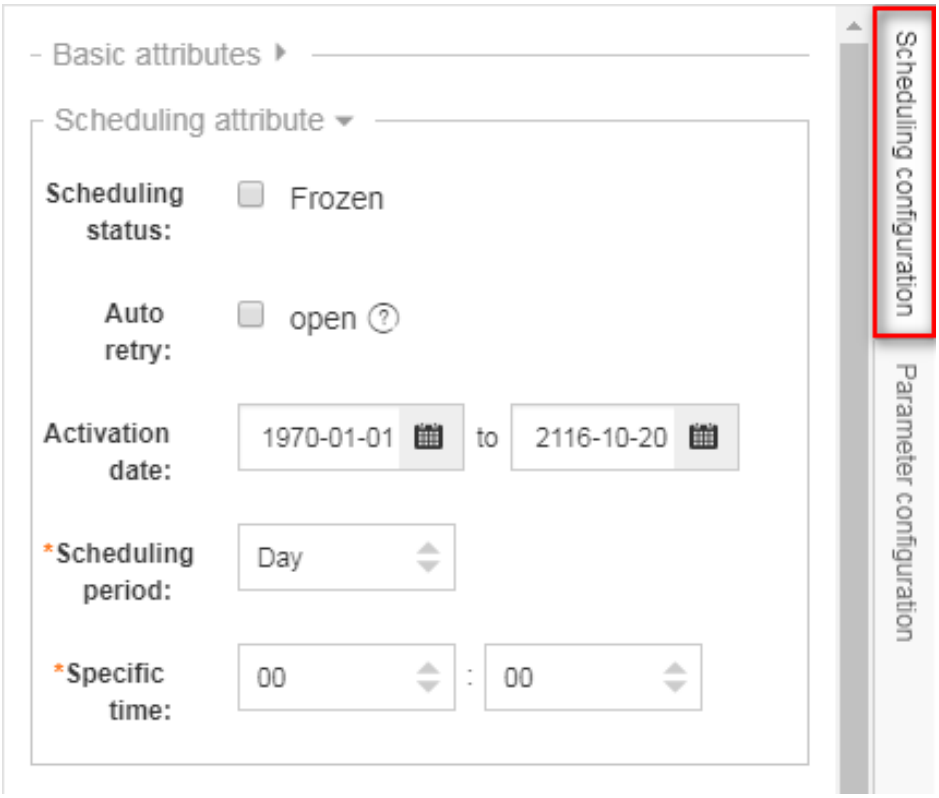
task trigger mechanisms to perform **tens of millions** of tasks accurately and punctually each day based on DAG relationships. It supports scheduling by minute, hour, day, week and month. For details, see Scheduling configuration description.

This section uses write_result created in Create a synchronization task as an example and configures the scheduling period to weekly, to explain the scheduling configurations and task O&M functions of DataWorks.

Instruction

Configure the scheduling attribute of a synchronization task

Go to the **Data Development > Task Development** page, and double-click the synchronization task you want to configure (write_result), then click **Scheduling Configuration** to configure the **scheduling attribute** of the task, as shown in the following figure:



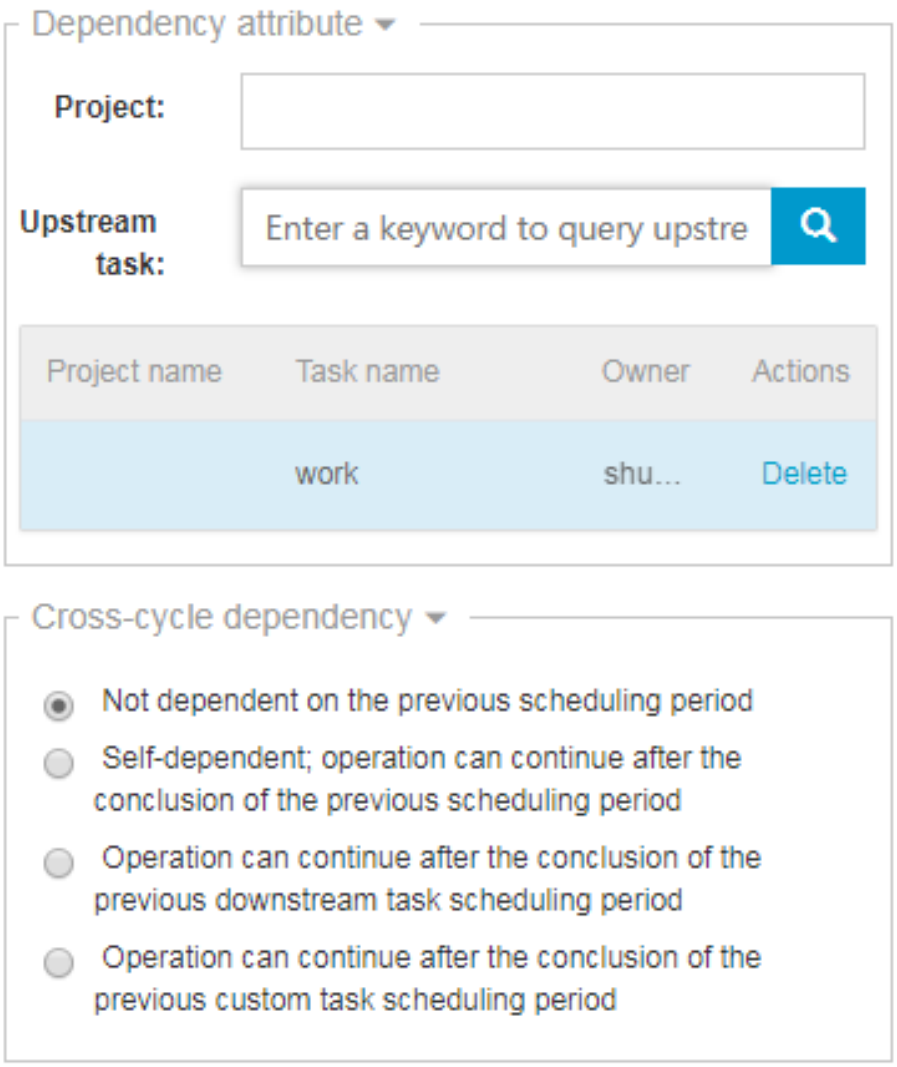
The configuration parameters are described below:

- Scheduling status: When this parameter is selected, the task is paused.
- Error retry: When this parameter is selected, error retry is enabled.
- Start date: The date on which the task takes effect, which can be set based on actual needs.
- Scheduling period: The operating period of the task, which can be set by month, week, day, hour, and minute. For example, a task can be scheduled weekly.
- Specific time: The specific operating time of the task. For example, you can set up the task to

run at 02:00 each Tuesday.

Configure the dependency attribute of a synchronization task

After configuring the scheduling attribute of a task, you can configure its dependency attribute, as shown in the following figure:



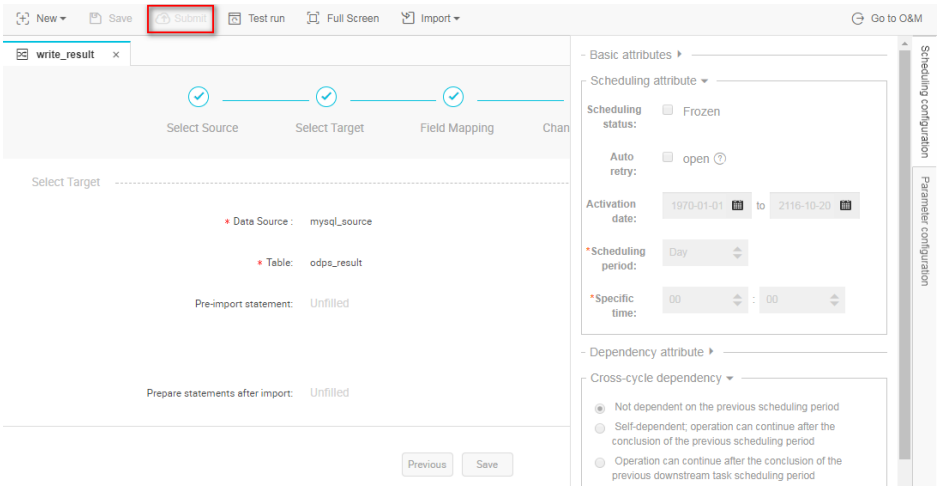
You can configure an upstream dependency for a task. In this way, even if the scheduled time of an instance of the current task is reached, the task can run only after the instance of its upstream task is completed.

The configuration in the above figure indicates that instances of the current task will be triggered only after the instance of the upstream task write_result is finished. You can enter “work” in the upstream task to configure an upstream task for write_result.

If no upstream task is configured, the current task is triggered by the project by default. Therefore, by default, the upstream task of the current task is project_start in the scheduling system. By default, a project_start task is created as a root task for each project.

Submit a synchronization task

Save the synchronization task “write_result”, and click **Submit** to submit it to the scheduling system, as shown in the following figure:



The system automatically generates an instance for the task at each time point according to the scheduling attribute configuration and periodically runs the task from the second day only after a task is submitted to a scheduling system.

> NOTE:
>
> If a task is submitted after 23:30, the scheduling system automatically generates instances for the task and periodically runs the task from the third day.

Subsequent steps

Now you know how to set the scheduling attribute and dependency of a synchronization task. Continue to the next tutorial for further study. This tutorial shows you how to perform periodic O&M for submitted tasks and view the log troubleshooting results. For details, see Perform periodic O&M and view log troubleshooting results.

In the previous operations, you have set a synchronization task to be run at 02:00 every Tuesday. After the task is submitted, you can view the automatic operation results in the scheduling system from the second day. Now, how can we check whether the instance schedule and dependency are as expected? DataWorks provides three triggering methods: test run, data population, and periodic running. Details about the three methods are as follows:

Test run: The task is triggered manually. If you need to check the timing and operation of a single task, test run is recommended.

Data population: The task is triggered manually. This method applies if you need to check the timing and dependencies of multiple tasks or re-execute data analysis and computing from a root task.

Periodic running: The task is triggered automatically. After a task is submitted successfully, the scheduling system automatically generates task instances at different time points starting from 00:00 of the second day. It checks whether upstream instances of each instance have run successfully at the scheduled time. If all the upstream instances have run successfully at the scheduled time, the current instance runs automatically without manual intervention.

NOTE:

The scheduling system periodically generates instances based on the same rules that apply in both manual and automatic triggering modes.

The period can be set to monthly, weekly, daily, hourly, or even by minute. The scheduling system always generates an instance for the task on the specified day or at the specified time.

The scheduling system only regularly runs the instance on the specified date and generates operation logs.

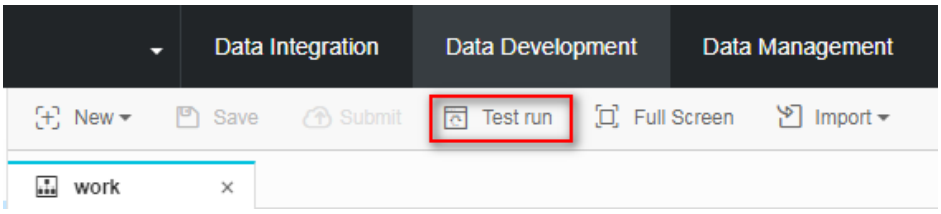
Instances rather than on the specified date are not run, and their statuses are directly changed to "Successful" when the running conditions are met. Therefore, no running logs are generated.

The following instructions show how to configure these three triggering methods.

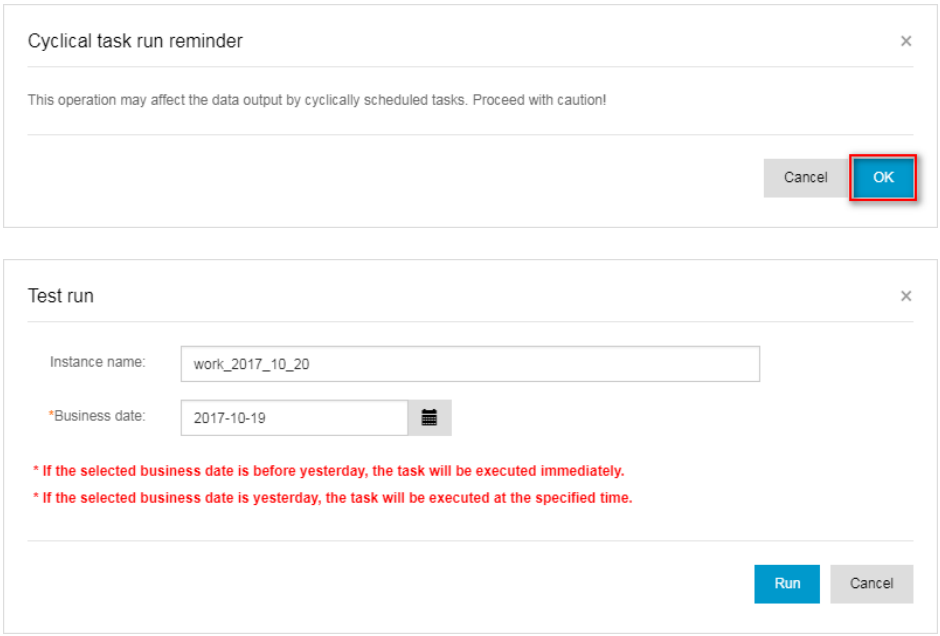
Test run

Manually trigger the test run

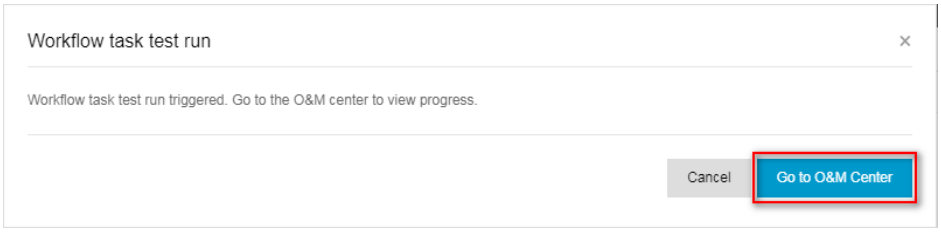
Click the **Test Run** button on the flow page.



As promoted on the page, click **Confirm** and **Run**.



Click **Go to O&M Center** to view the task operation status.



View the information and operation logs of the test instance

Click the task name to view the instance DAG. In the instance DAG view, right-click an instance to view its dependencies and detailed information. Also, you can terminate or re-run the instance. In the instance DAG view, double-click an instance and a dialog box appears, showing the task attributes, running logs, operation logs, and code.

Note:

In test run mode, the task is triggered manually. The task runs immediately as long as the set time is reached, regardless of the instance' s upstream dependencies.

According to the previously mentioned instance generation rules, set up the task write_result to run at 02:00 each Tuesday. If the business date of test run is Monday (business date = running date -1), the instance runs at 02:00. If not, the instance status is changed to "Successful" at 2:00 and no logs are generated.

Data population

Manually trigger data population

If you need to check the timing and dependency of **multiple tasks** or re-execute data analysis and computing from a root task, go to the **O&M Center > Task List > Task Scheduling** page and click **Data Population Task** to run multiple tasks of a specific period of time.

Instruction

Log on to the **O&M Center > Task Scheduling page** and enter the task name.

Select the task query results and click the **Data Population** button.

Set the business date of the data population as May 11, 2017 to May 12, 2017, select the insert_data and write_result node tasks, and click **OK**.

Click **View Data Population Results**.

View the information and operation logs of the data population instance

On the **Data Population Instance** page, find the task instance: Click the task name to view the instance DAG. In the instance DAG view, right-click an instance to view its dependencies and detailed information. Also, you can terminate or re-run the instance. In the instance DAG view, double-click an instance and a dialog box appears, showing the task attributes, running logs, operation logs, and code.

Note:

Data population task instances are dependent on instances from the previous day. For example, for a data population task within the period from September 15, 2017 to September 18, 2017, if the instance on the 15th is failed to run, the instance on the 16th is not run.

According to the previously mentioned instance generation rules, set up the task write_result to run at 02:00 each Tuesday. If the business date selected during data population is Monday (service date = running date -1), the instance runs at 02:00. If not, the instance status is changed to “Successful” at 02:00 and no logs are generated.

Periodic automatic run

In periodic automatic run mode, the scheduling system automatically triggers tasks according to all task scheduling configurations. Therefore, no operation portal is provided on the page. You can view the instance information and operation logs in either of the following methods:

Go to the **O&M Center > Task Scheduling** page, select parameters such as service date or running date, search instances corresponding to the task write_result, and then right-click on an instance to view its information and operation logs.

Click the task name to view the instance DAG. In the instance DAG view, right-click an instance to view its dependencies and detailed information. Also, you can terminate or re-run the instance. In the instance DAG view, double-click an instance and a dialog box appears, showing the task attributes, running logs, operation logs, and code.

Note:

If the initial status of a task instance is “Not Run”, when the scheduled time is reached, the scheduling system checks whether all the upstream instances are successful.

The instance is triggered only when all of its upstream instances are successful and its scheduled time is reached.

For an instance in Not Run status, check that all its upstream instances are successful and its scheduled time has been reached.