Alibaba Cloud DataWorks

Quick Start

Issue: 20180911

MORE THAN JUST CLOUD | Alibaba Cloud

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

Contents

1 Instructions

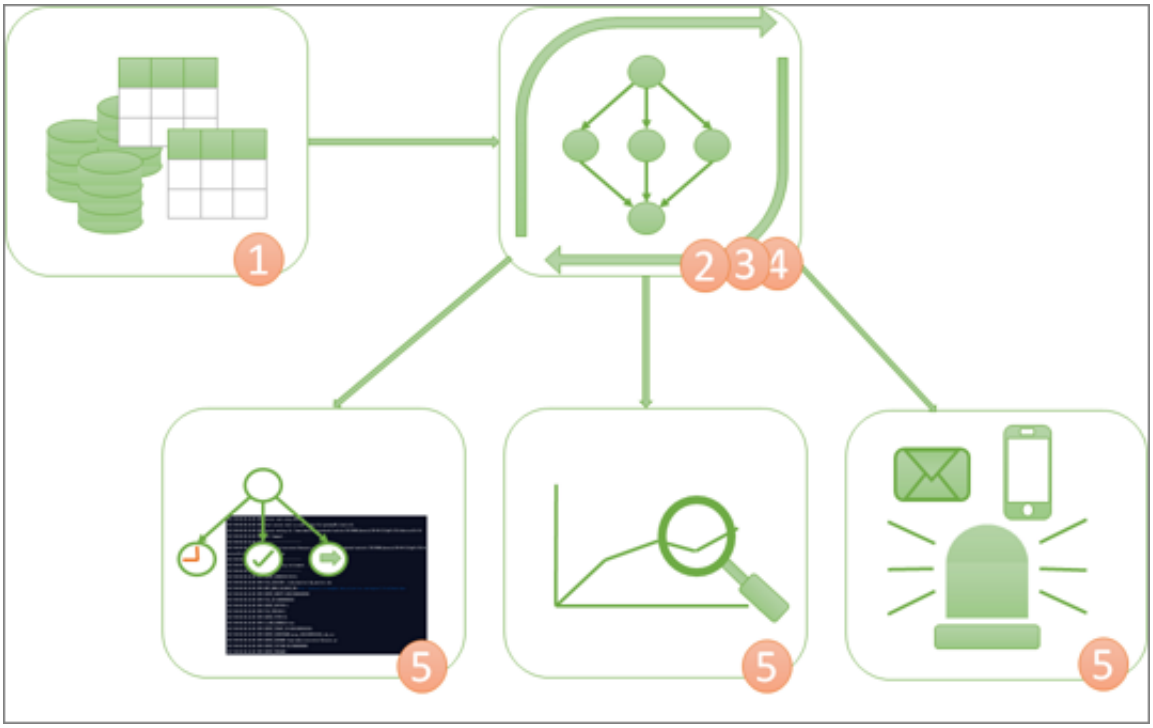
This module will guide you through a complete data development and operations operation.

Note:
If you are using DataWorks for the first time, make sure that you have already done the work based on the *preparation* module, prepare accounts, project roles, project space, and so on, then enter the DataWorks Management Console, start the data development operation by clicking **enter workspace** after the corresponding project.

Typically, data development and operations through the project space of DataWorks include the following actions:

- *Step 1: Create a table and upload data*
- *Step 2: Create a Business Flow*
- *Step 3: Create a synchronization task*
- *Step 4: Scheduling and dependence settings*
- *Step 5: O&M and view log troubleshooting results*

A general process is shown in the following figure:



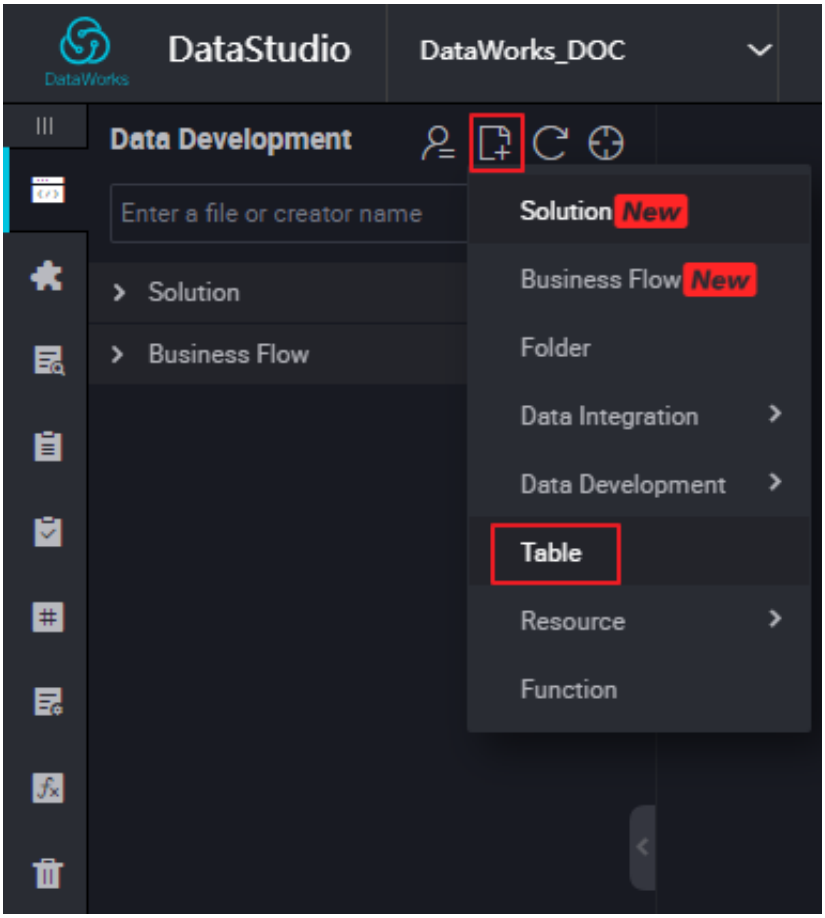
2 Step 1: Create a table and upload data

In this article, we use 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 stores the business data, while the result_table stores the results after data analysis.

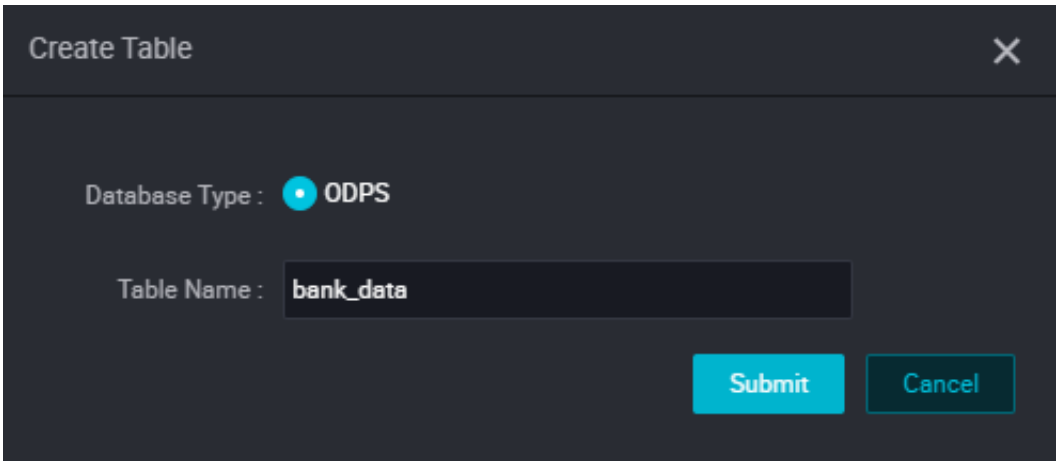
Procedure

Create a table called bank_data

1. After *Create a project* , click **Enter workspace**in the corresponding project.
2. Go to the **Data Studio (original data development)** page and select **new** > **table**.

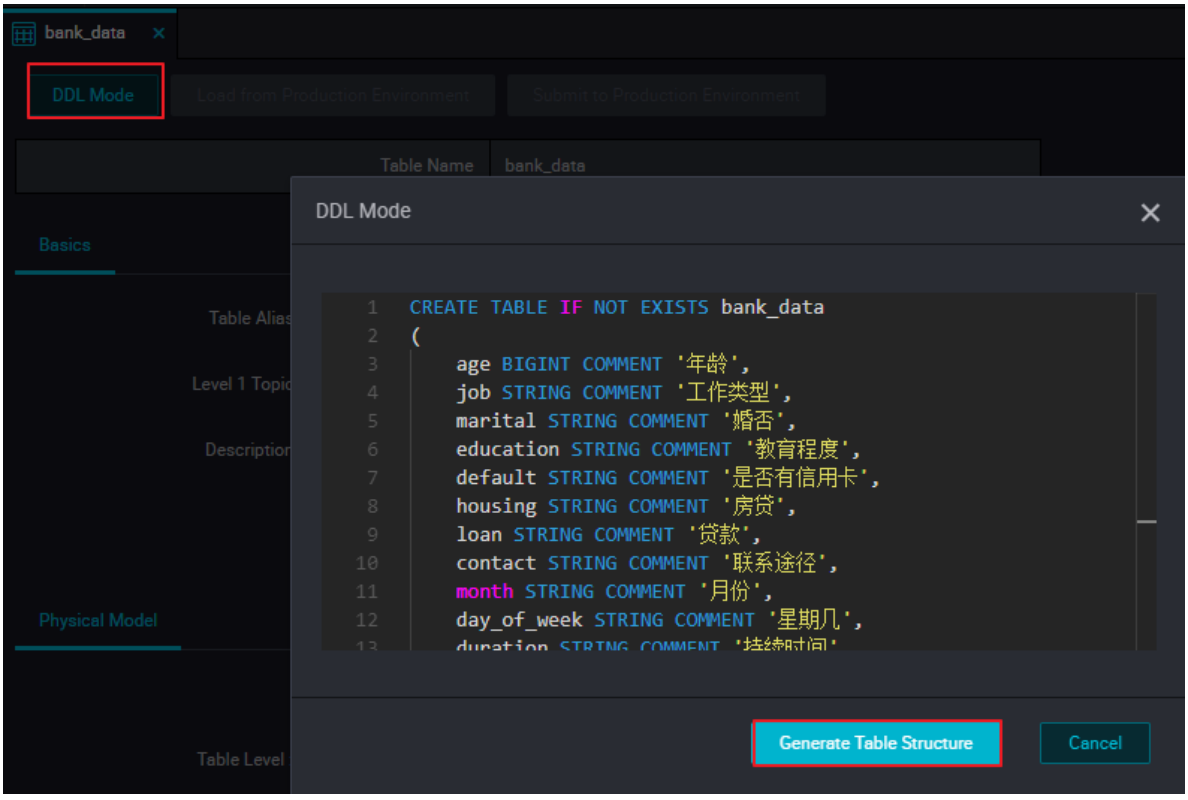


3. Fill in the name of the table in the **new table** dialog box.



4. Click **Submit**.
5. Enter the **new table** page, and select the **DDL mode**.
6. Enter the table creation statement in the **DDL schema** dialog box, and click **build table structure**.

   For more SQL syntax for creating tables, see *creating/viewing/deleting tables*.

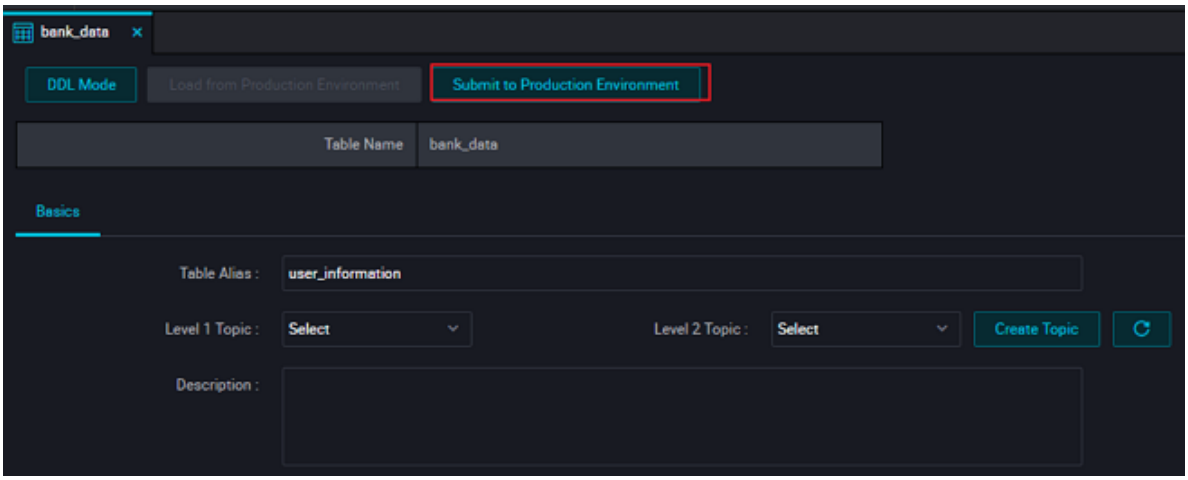


The statements used for table creation in this example are as follows:

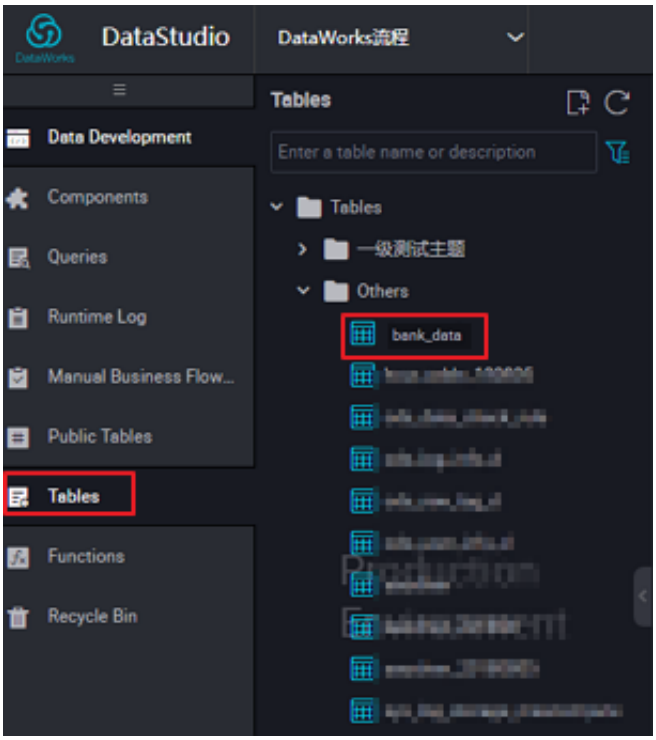
```
CREATE TABLE IF NOT EXISTS bank_data
(
 age             BIGINT COMMENT 'age',
 job             STRING COMMENT 'job type',
 marital         STRING COMMENT 'marital status',
```

```
 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'
);
```

7. After the table structure is generated, enter the Chinese name of the table and click **Submit to development environment**.

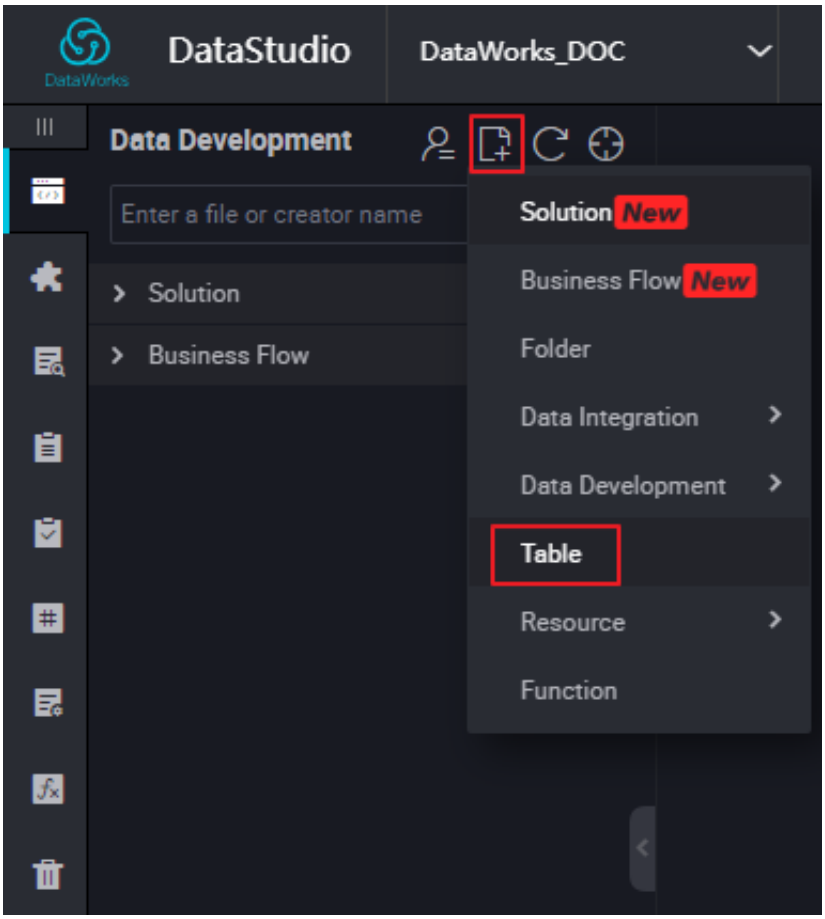


8. After the creation is successful, you can search it by entering the table name in the left-hand navigation **table management**, view table information.



Create result_table

1. Go to the **DataStudio** page and select **new** > **table**.



2. Fill in the name of the table in the **new table** dialog box and click **Submit**.
3. Enter the **new table** page, and select the **DDL mode**.
4. Enter the build TABLE statement in the **DDL schema** dialog box, and click **build table structure**. An example of table creation is as follows:

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

5. After the creation is successful, you can search it by entering the table name in the left-hand navigation **table management**, view table information.

Upload local data to bank_data

Dataworks supports the following actions:

- Uploading the data in locally stored text files to the table in the workspace.
- Using the data integration module to import business data from various data sources to the workspace.

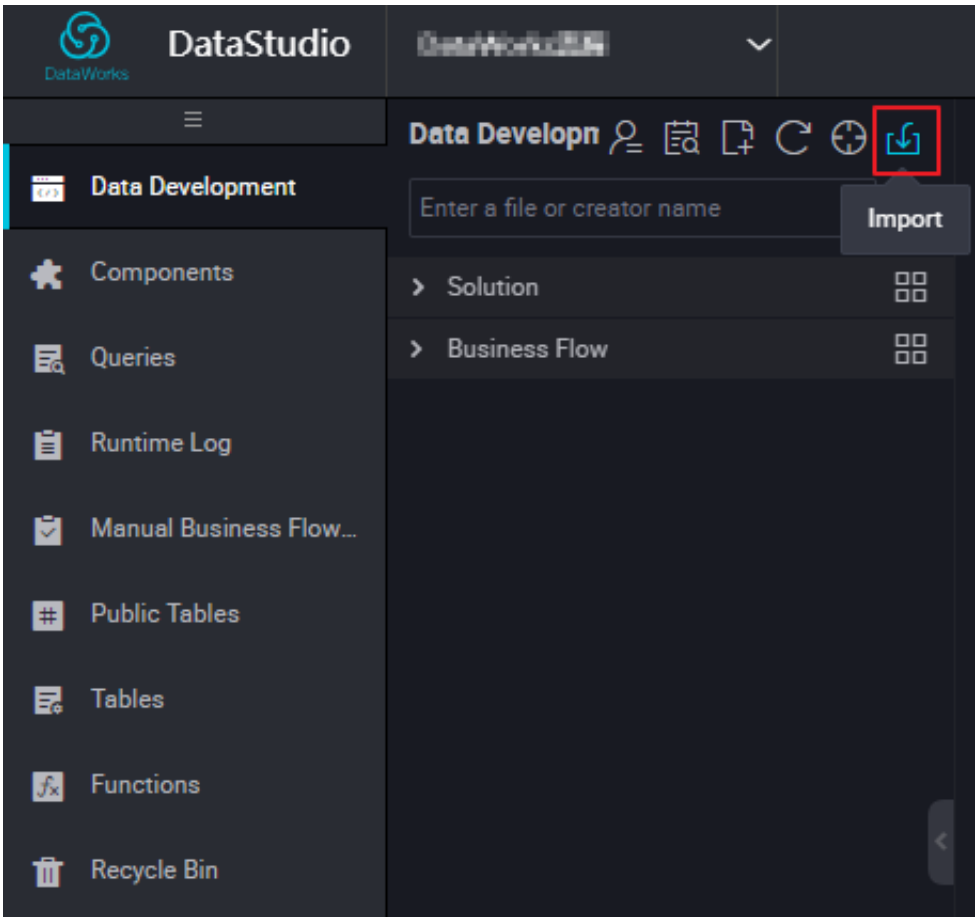
Note:

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

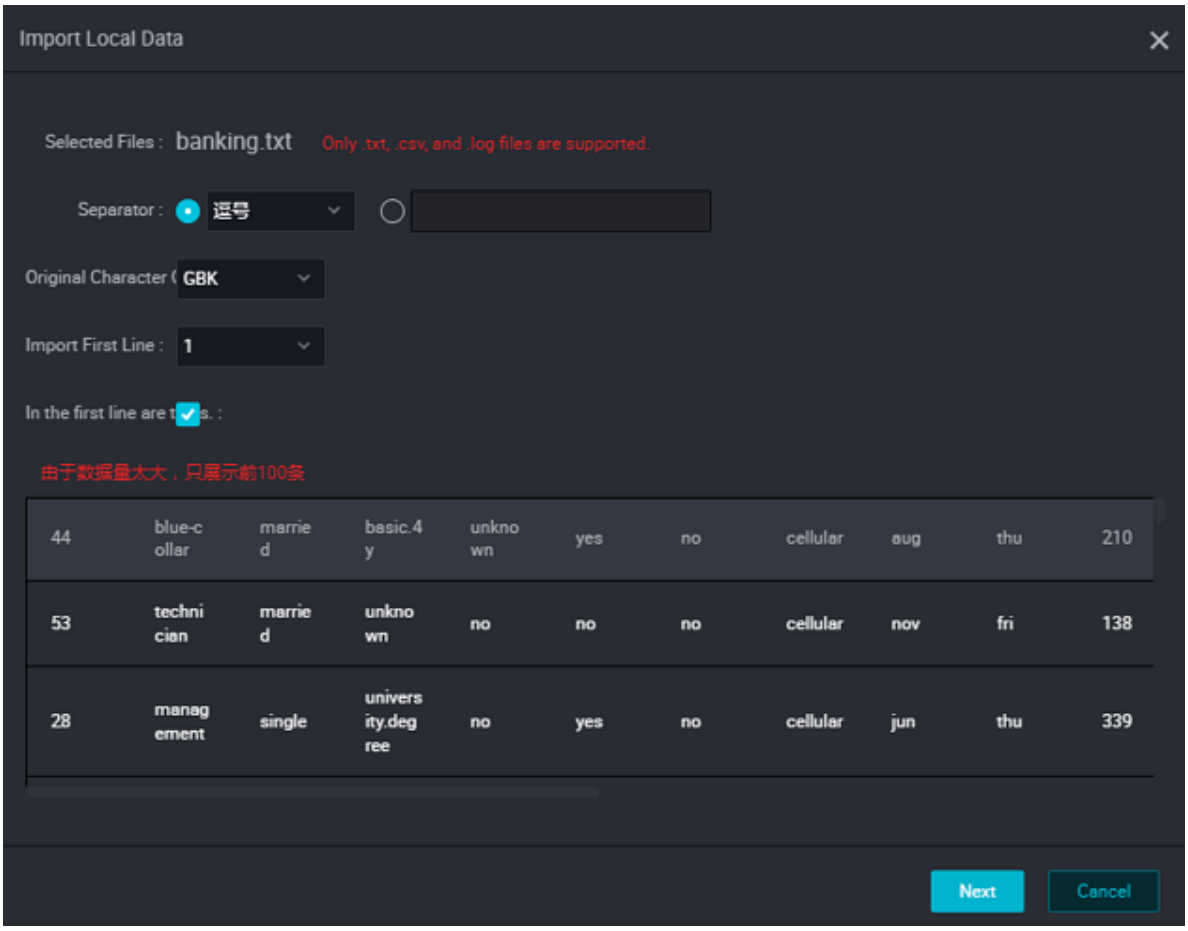
- File type: Only .txt and .csv files are supported.
- File Size: Not exceeding 10 M.
- Operation objects: Partition and non-partition tables can be imported, but Chinese partition values are not supported.

For example, import local file *banking.txt*to DataWorks, the operation is as follows:

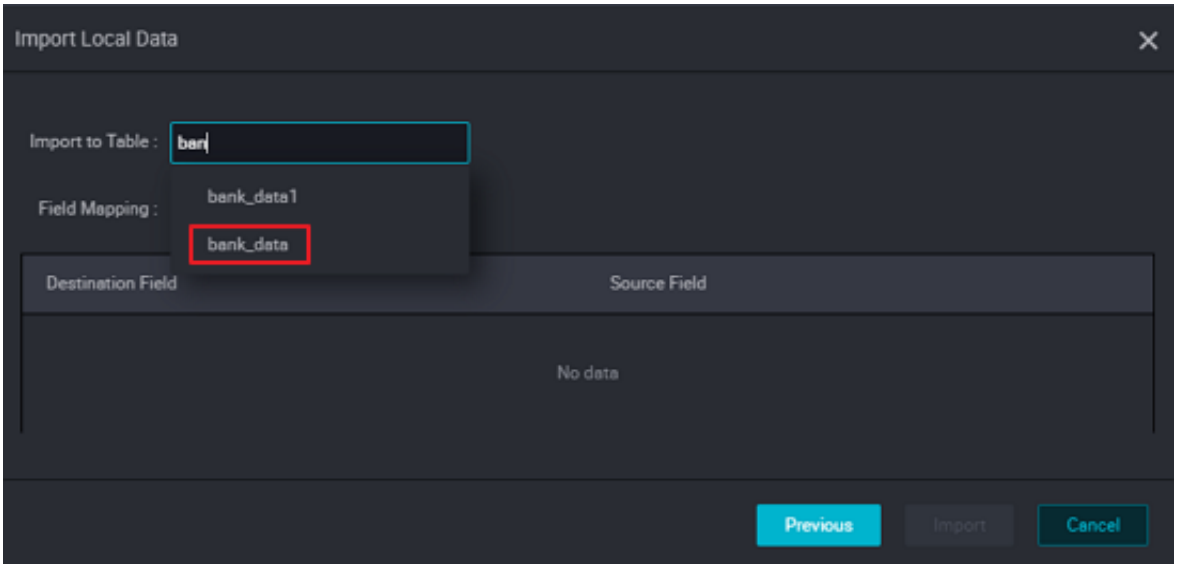
1. Click **Import** to select **import local data**.



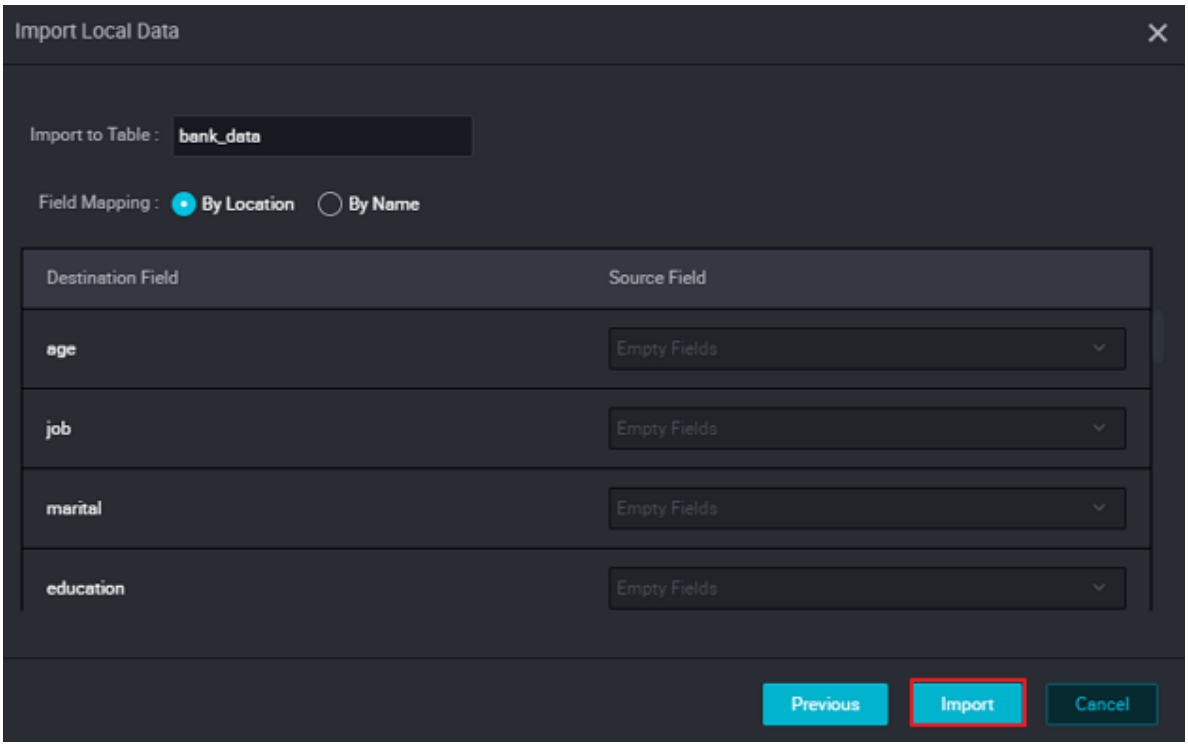
2. Select a local data file, configure the import information, and click **Next**,



3. 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.



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



After the file is imported, the system returns the number of lines that were successful in your data import or an exception that failed.

Other data import methods

- Create a Data Synchronization task

  This method applies to saving RDS, MySQL, SQL Server, PostgreSQL, MaxCompute, OSS , DRDs, OSS data from a variety of data sources such as, Oracle, FTP, DM, HDFS, and MongoDB.

  For details on creating a data synchronization task with Dataworks, see *creating a data synchronization task*.

- Local file uploads

  Ci fang shi yong yu wen jian great&small bu chao guo 10m、 wen jian lei xing wei .txt he .csv data, the target supports partition tables and non-partition tables, but does not support Chinese as a partition.

  For local file upload via DataWorks, see local data upload to bank_data above for details.

- Upload files using tunnel command

  This method applies to local files and other resource files more than 10 m in size.

Upload and download the data through tunnel commands provided by the *MaxCompute client*, when local data files need to be uploaded to the partition table, they can be uploaded using the client tunnel command. See *Tunnel command actions* for details.

Next steps

You have learned know how to create a table and upload data now. You can go to the next tutorial which will show you how to create a flow for further data analysis and computing in the project space. For more information, see *creating a business process*.

3 Step 2: Create a Business Flow

This article will take the creation of business flows as an example, describes how to create nodes and configure dependencies in your business flow, to facilitate the design and presentation of the steps and sequences of data analysis. This article briefly explains how to use the data development function to further analyze and calculate the workspace data.

DataWorks data development features support visual drag-and-drop in the business flow to complete inter-node dependency settings. The flowing and interdependencies of data are implemented in the form of operational business flows. Multiple Task types such as ODPS SQL, data synchronization, open_mr, shell, machine learning, and virtual nodes are currently supported, for specific usage methods for each task type, see*Introduction of Node Type*.

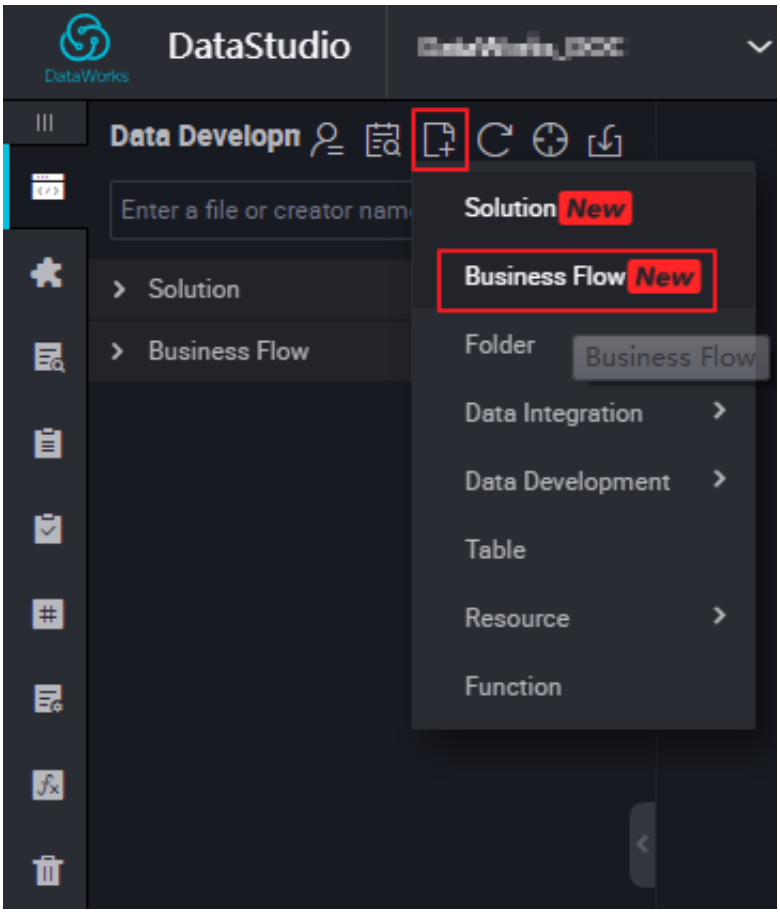
Prerequisites

Make sure that you have *built the table and uploaded the data*, prepare the business data table bank_data and the data in it in the workspace, as well as the result table.

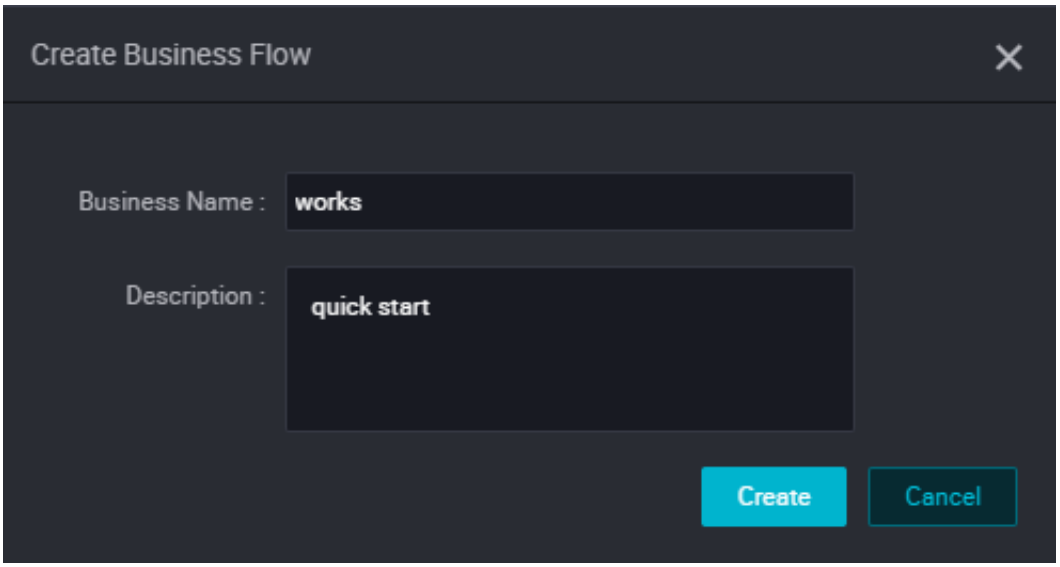
Procedure

Create a Business Flow

1. After *Create a project*, click **Enter workspace** in the corresponding project.
2. Go to the **DataStudio** page and select **create** > **business flow**.



3. Enter the name and description of the business flow.



Create a node and dependency on the flow canvas

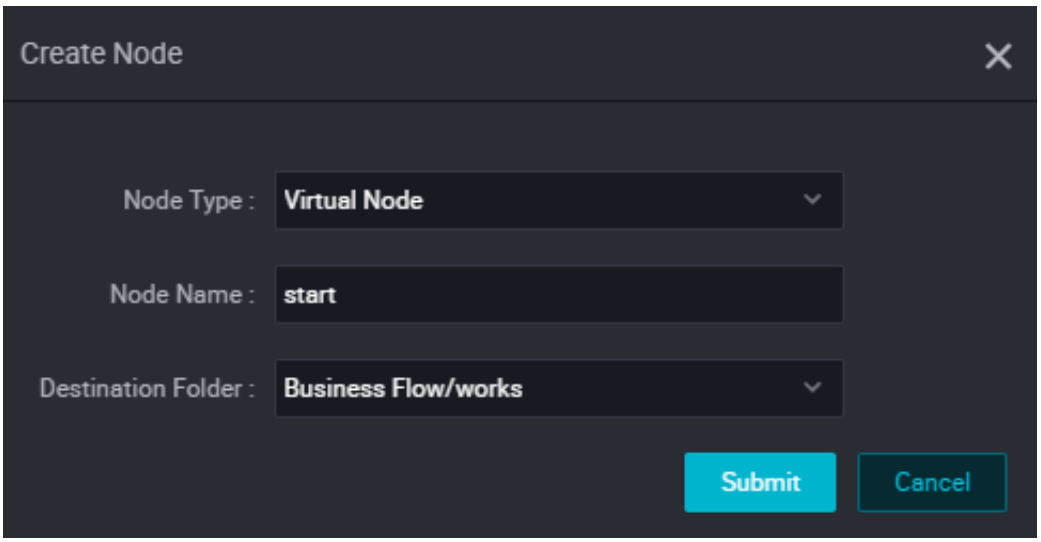
This section shows how to create a virtual node “start” and an ODPS SQL node “insert_data”, and to configure “insert_data” to depend on “start”.

Note:

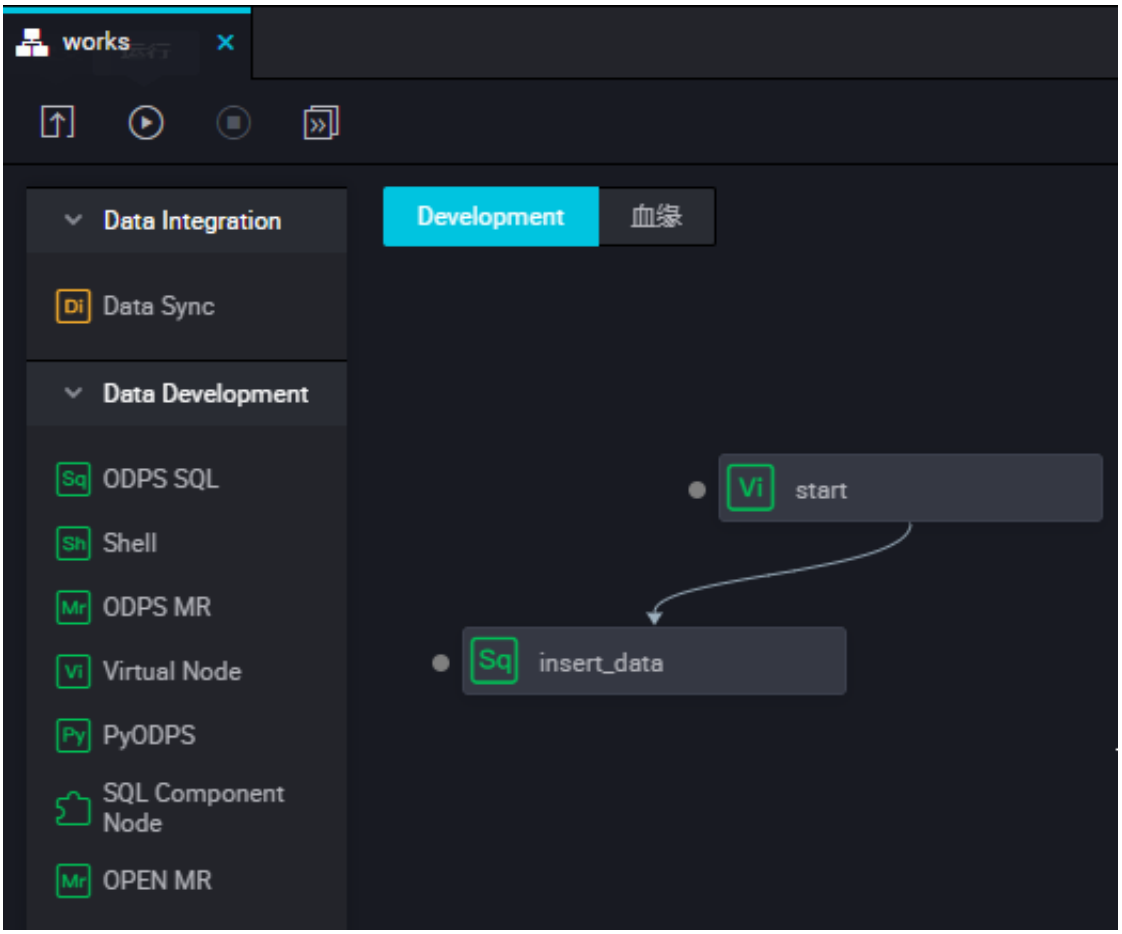
- As a control-type node, the virtual node does not affect the data during flow operation and is only used for O&M control of downstream nodes..
- When a virtual node depends on the other nodes and its status is manually set to failure by the O&M personnel, its downstream nodes that have not run yet, cannot be triggered. This prevents further propagation of erroneous upstream data during the O&M flow. For more information, see the section on virtual nodes in *Introduction of Node Type*.
- The upstream task of a virtual node in a business flow is typically set as the root node of the project, the format of the Project root node is: Project name _ root.

We recommend that you create a virtual node as the root node to control the whole flow when designing a flow.

1. Double-click the virtual node and enter the node name start.



2. Double-click **ODPS SQL** to enter the node name “insert_data”.
3. Click the start note, and draw a line between start and insert_data to make insert_data dependent on start, as shown in the following figure:



Editing code in the ODPS SQL Node

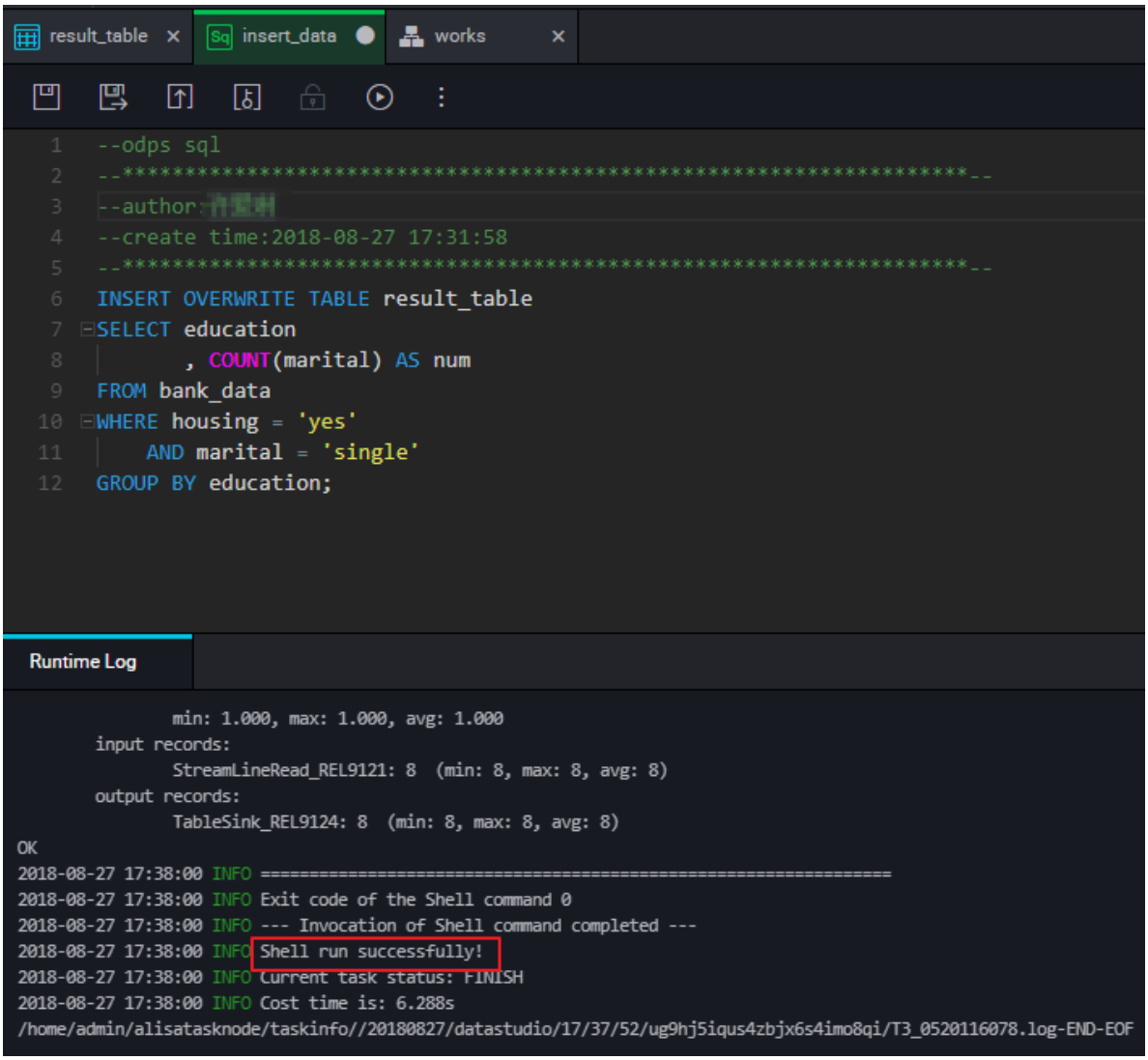
This section describes how to use SQL code in the ODPS SQL node **insert_data** to query the quantity of mortgages available for individuals having different educational background and save results for analysis or display by the following nodes.

The SQL statements are as follows. For more information about the syntax, see *MaxCompute SQL*.

```
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
```

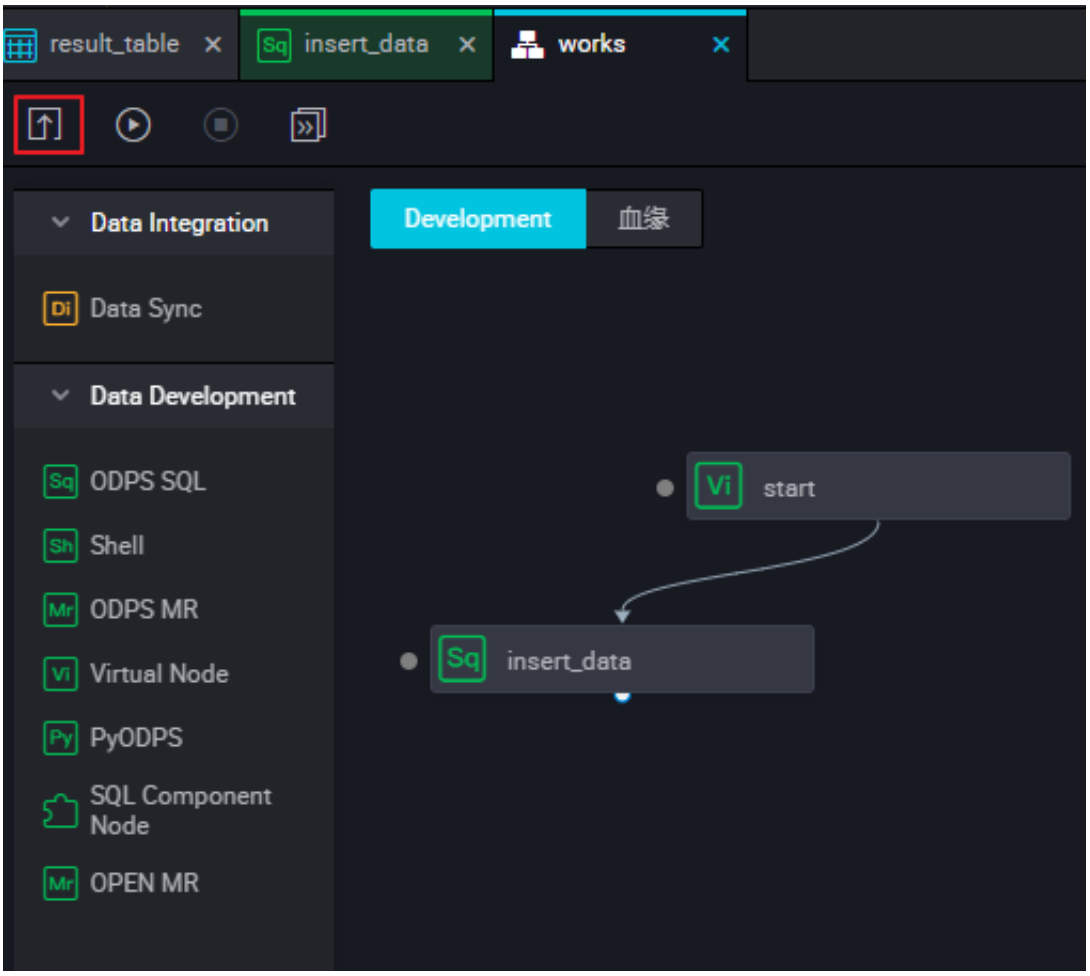
Run and debug ODPS SQL

1. After editing the SQL statements in the insert_data node, click **Save** to prevent code loss.
2. Click **Run** to view the operations logs and results,



Save and submit business flows

After running and debugging the ODPS SQL node “insert_data”, return to the flow page. Click **Save** and **Submit** the whole flow.



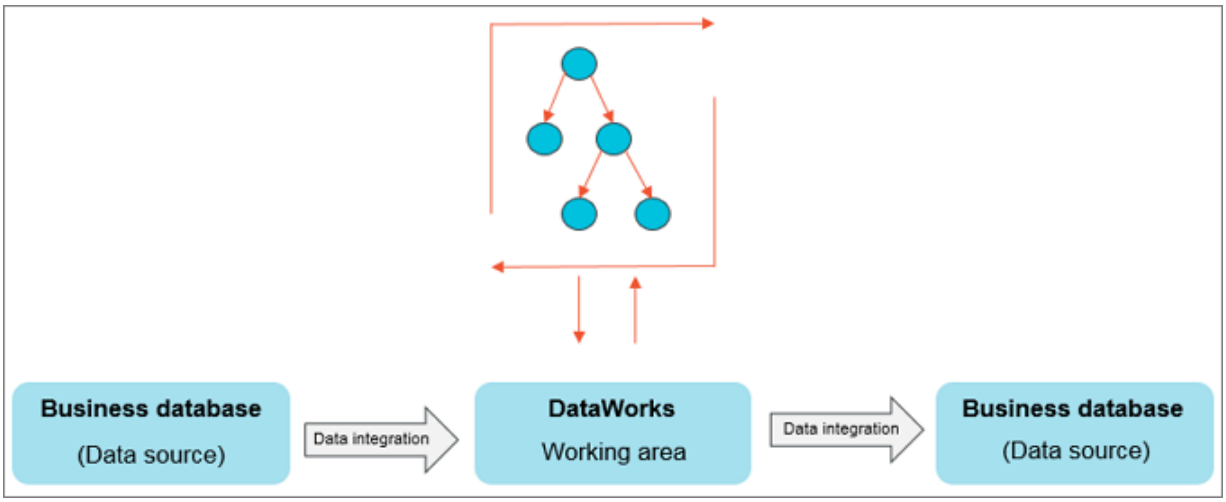
Subsequent steps

Now you have learned how to create, save, and submit the flow. You can proceed with the next tutorial which shows how to create a synchronization task to export data to the diffrent types of the data sources. For more information, see *creating synchronization task export results*.

4 Step 3: Create a synchronization task

This article will take MySQL Data sources as an example, showing how to export data from DataWorks to a MySQL data source through the data integration feature.

In DataWorks, data integration is typically used to periodically import the business data generated in your system into the workspace, after the calculation of the SQL task, the calculation results are periodically exported to the data source that you specify, for further presentation or running usage.



Currently, data from the following data sources can be imported to or exported from the workspace through the data integration function: RDS, MySQL, SQL Server, PostgreSQL, MaxCompute, ApsaraDB for Memcache, DRDS, OSS, Oracle, FTP, DM, Hdfs, MongoDB, and so on. For more information, see *Supported data sources*.

Prerequisites

- If you are using a self-built database on ECS, you need to *add security groups* to your ECS.
- If you are using data sources such as RDS/MongoDB, you need to *add a white list* to a console such as RDS/MongoDB.

> **Note:**
> If you use a custom resource group to schedule the RDS data synchronization task, you must add the IP address of the computer hosting the custom resource group to the RDS whitelist.

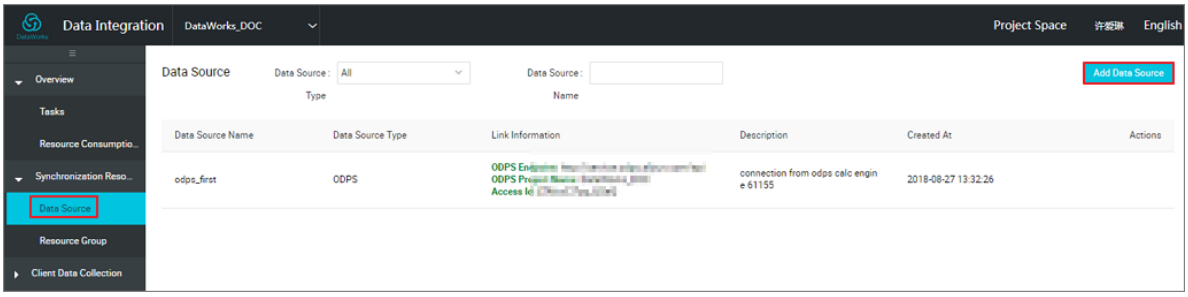
Procedure

Add a data source

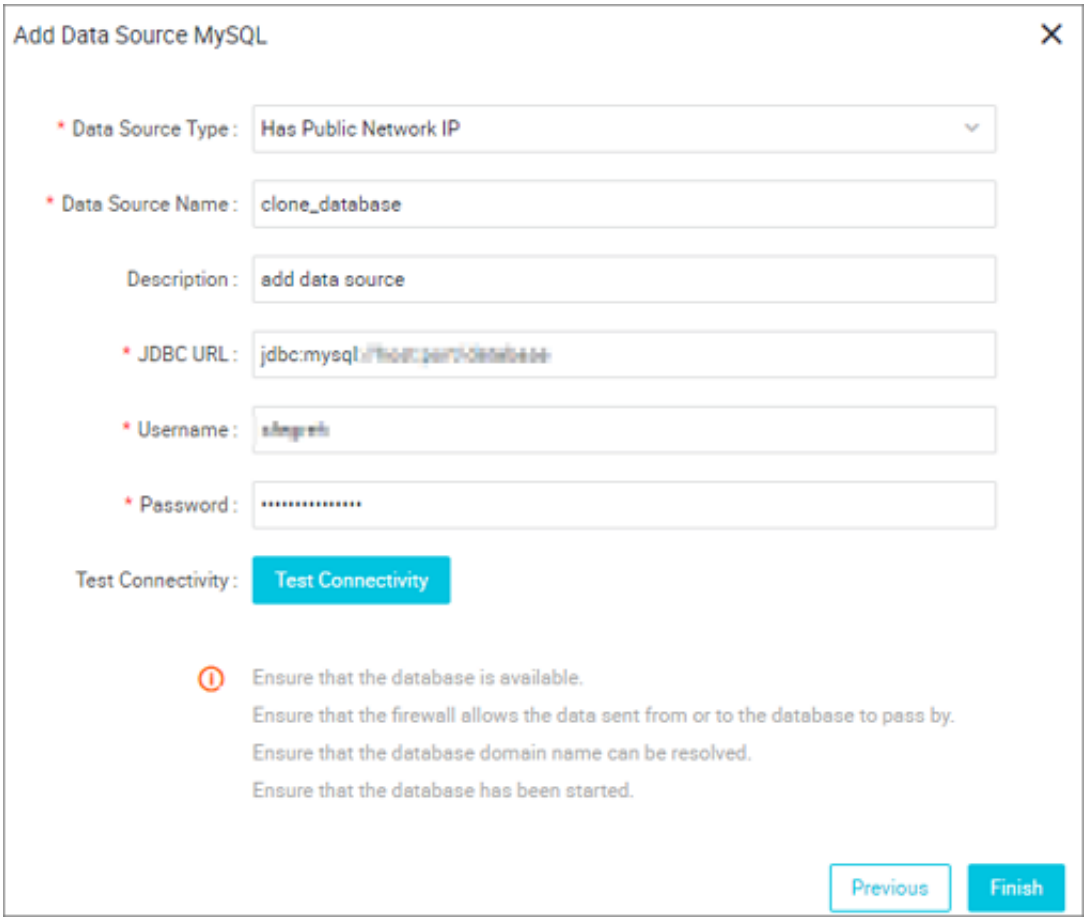
> **Note:**

Only the Project Administrator role can create new data sources, and members of other roles can view data sources only.

1. Log on to the *DataWorks management console* as the Project Administrator.
2. Select **enter workspace** in the corresponding item actions column under the **list of items**.
3. Click **data integration** in the top menu bar.
4. Click **data sources** in the left-hand navigation bar.
5. Click **Add data source** in the upper-right corner.



6. Fill in each configuration item in the **Add Data Source** dialog box.



- Data Source Type: With a public IP address.

- Data source name: The name must contain letters, numbers, and underlines, but cannot begin with a number or underline, For example, abc_1123.
- Data source description: The description cannot exceed 80 characters.
- JDBC URL: `jdbc:mysql://host:port/database`.
- User name/Password: The user name and password used to connect to the database.

For configuration instructions for different data source types, see *Data source configuration*.

7. (Optional).Click **Test Connectivity** after entering all the required information in the relevant fields.
8. If the test connectivity is successful, click **Finish**.

Note:
Make sure that the target MySQL database contains tables.

Create the table odps_result in the MySQL database. The statements used for table creation are as follows:

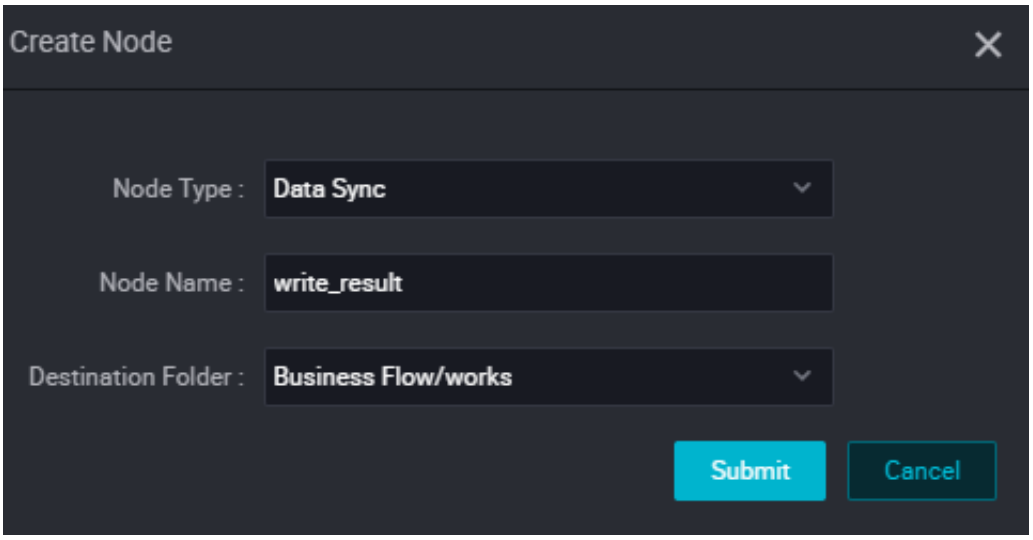
```
CREATE TABLE `ODPS_RESULT` (
`education`  varchar(255) NULL ,
`num`  int(10) NULL
)
```

After the table has been built, you can execute the `desc odps_result;` to view the table details.

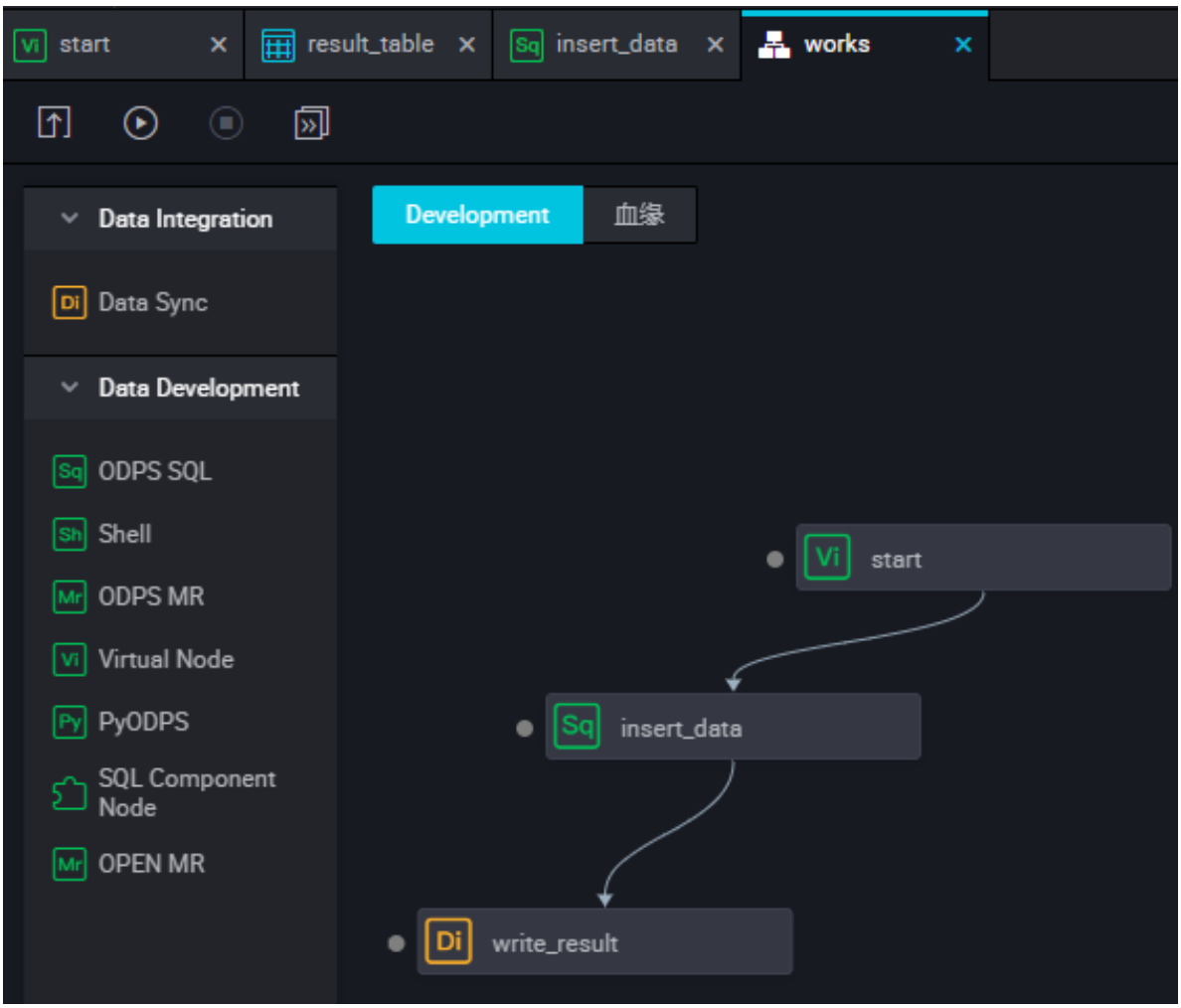
Creating and configuring synchronization node

This section shows how to create and configure the synchronization node **write_result**, and write data from result_table to the MySQL database. The specific steps are as follows.

1. Create the node write_result, as shown in the following figure.

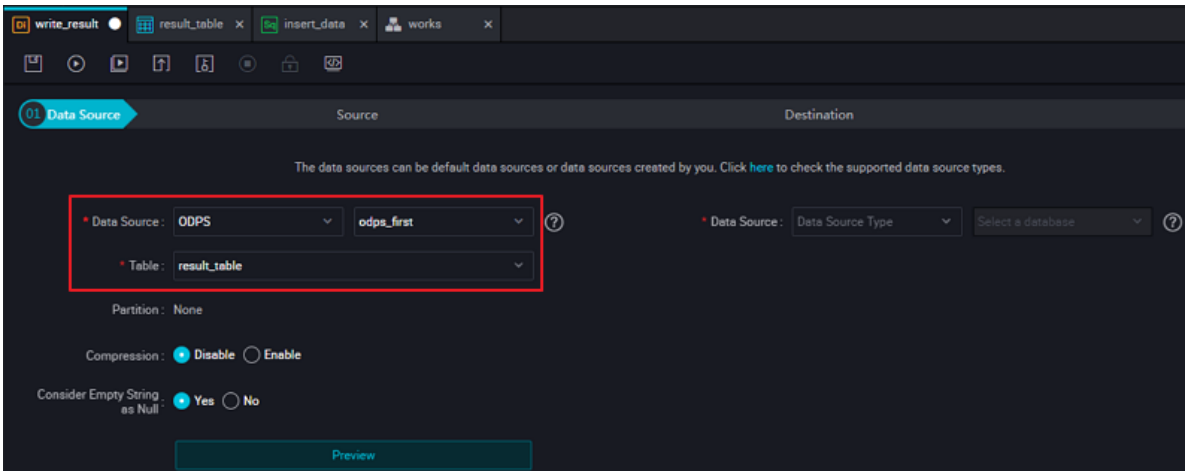


2. Sets the dependencies between nodes so that the write_result node is dependent on the insert_data node.



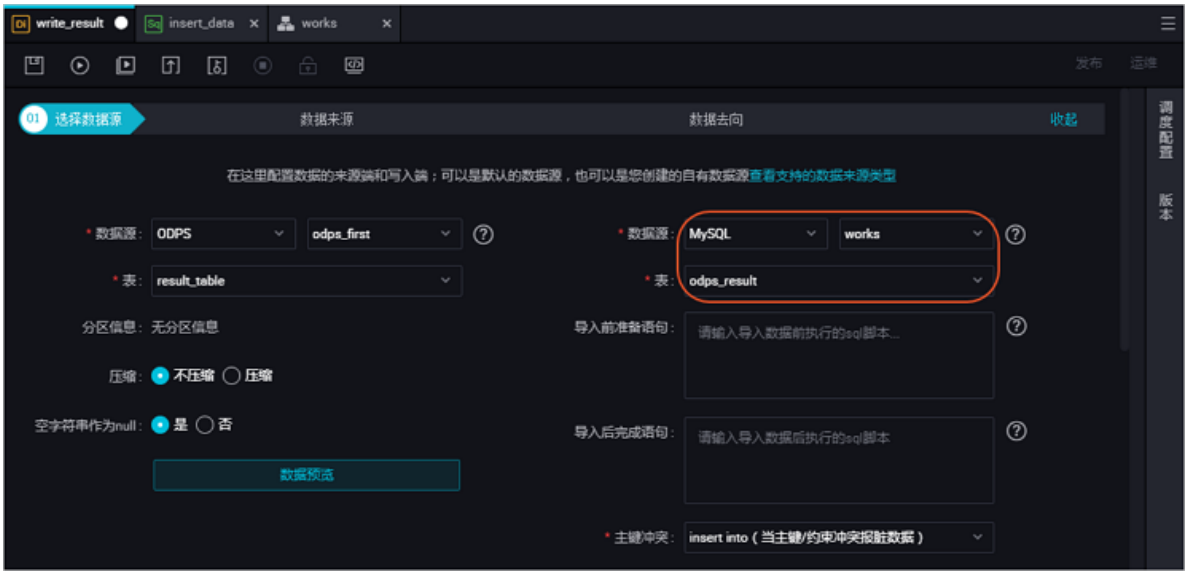
3. Select the source.

   Select the MaxCompute data source and the source table result_table and click **Next**.



4. Select a Target.

   Select the MySQL data source and target table ODPS _result, and click **Next**.

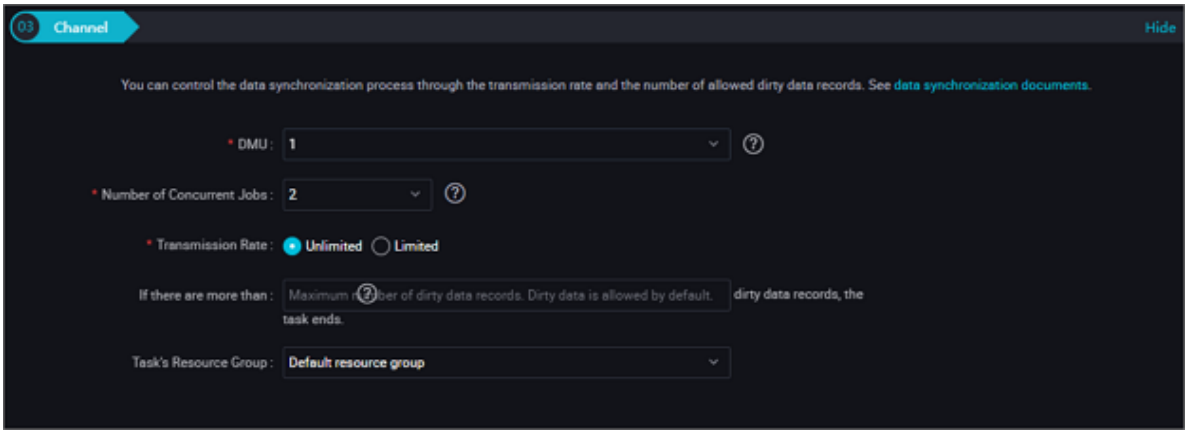


5. Map the fields.

   Select the mapping between fields. 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.

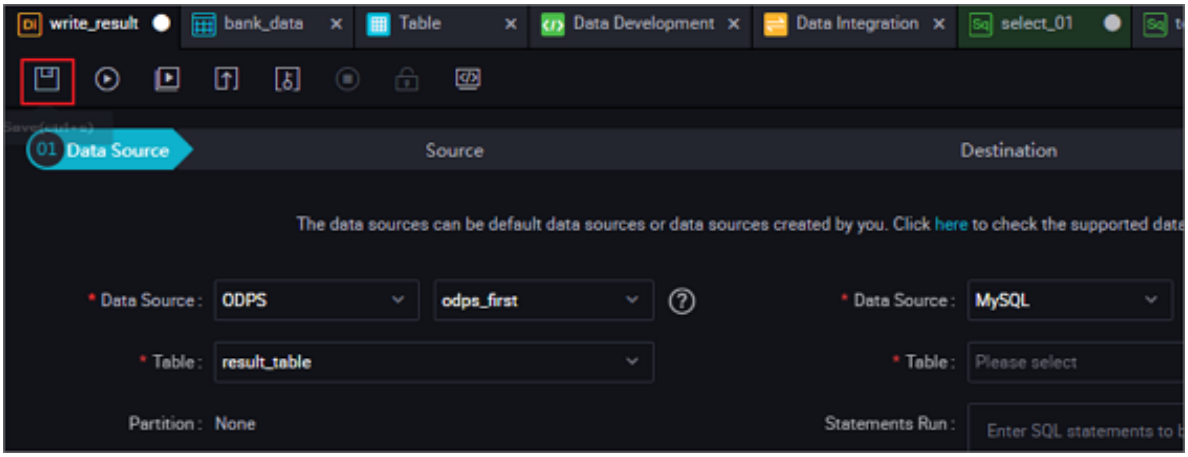
6. Control the channel.

   Click **Next** to configure the maximum job rate and dirty data check rules.



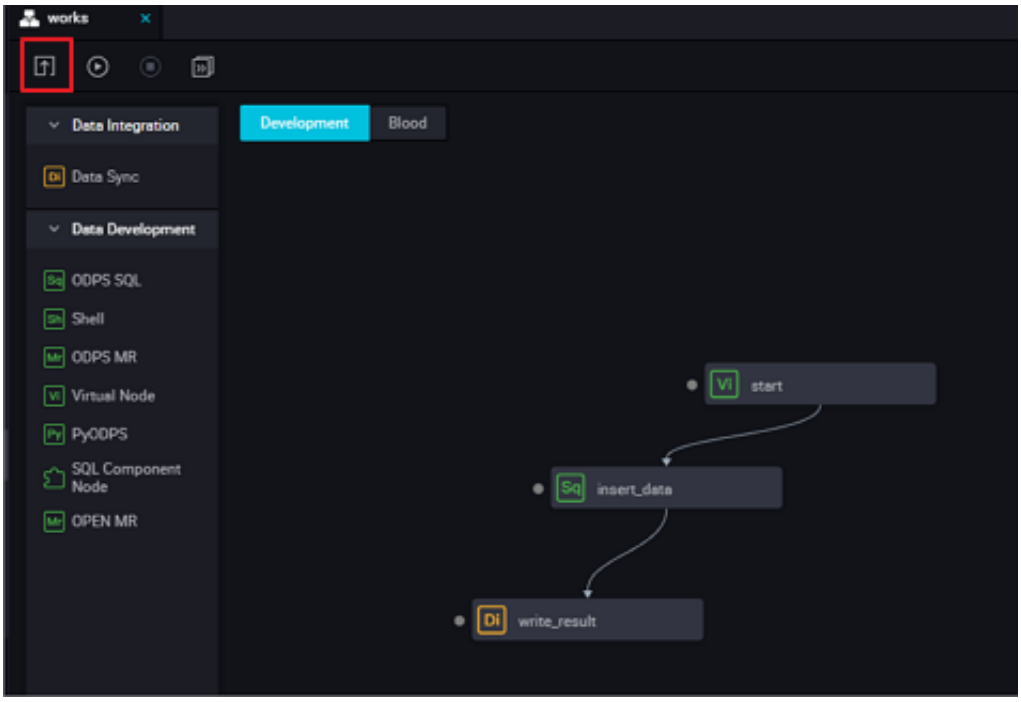
7. Preview and store.

   After completing the above configuration, scroll the mouse up and down to view the task configuration, and if it is not, click **Save**.



Submit a data synchronization task

Once you save a synchronization task click **Submit**, and the synchronization task is submitted to the scheduling system. The scheduling system automatically and periodically runs the task from the second day according to the configuration attributes.



Subsequent steps

Now, you know how to create a synchronization task and export data to data sources of different types. Continue to the next tutorial for further study. This tutorial shows you how to set the scheduling attribute and dependency for a synchronization task. For more information, see *setting schedule properties and dependencies* for tasks.

5 Step 4: Scheduling and dependence settings

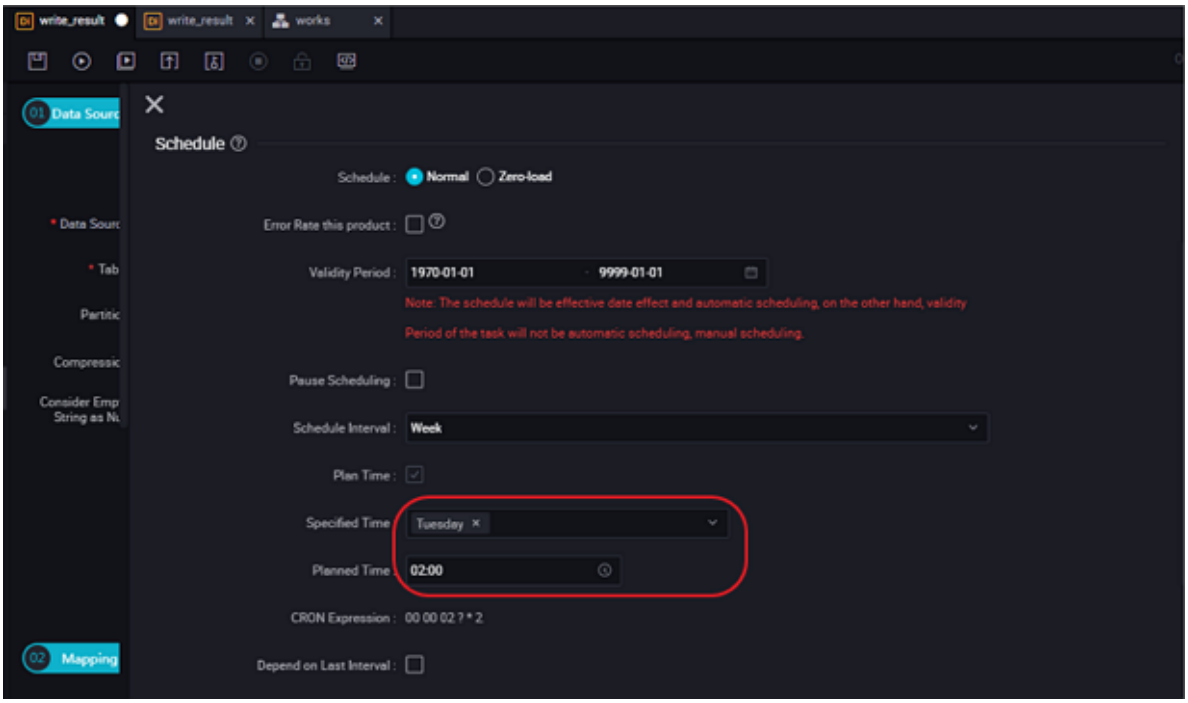
This article takes the "write_result" created in *creating synchronization tasks* as an example, configure its scheduling cycle as weekly scheduling, introduces the scheduling configuration and task operations features of dataworks.

DataWorks provides powerful scheduling capabilities including time-based or dependency-based task trigger functions to perform **tens of millions** of tasks accurately and timely each day, based on DAG relationships. It supports scheduling by minute, hour, day, week, and month. For more information, see *Create a synchronization task*.

Procedure

Configure the scheduling attribute of a synchronization task

1. Select the **data development** > **task Development** page.
2. Double-click the synchronization task (write_result) that you want to configure).
3. Click **schedule configuration** on the right to configure scheduling properties for the task.



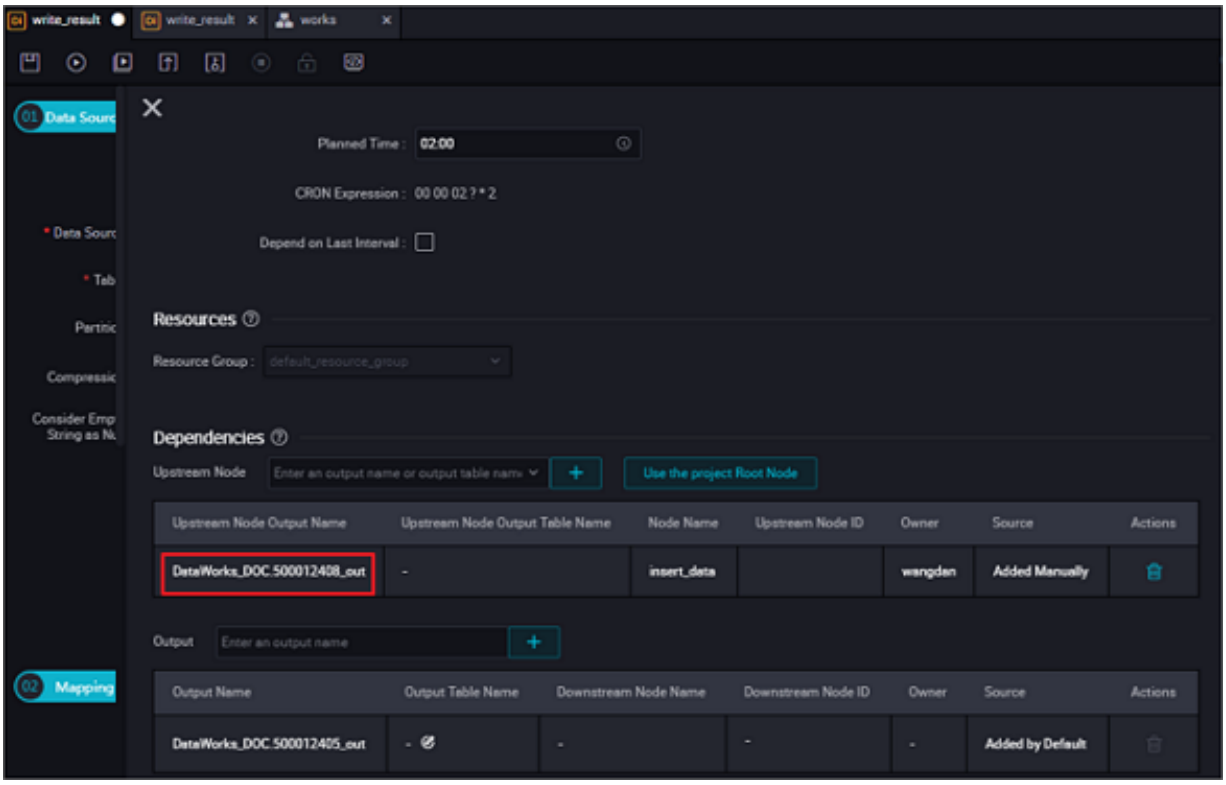
Parameters:

- 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 every Tuesday.

Configure dependency properties for a synchronization task

After the schedule properties configuration for the synchronization task is completed, you can configure its deployment dependency properties.



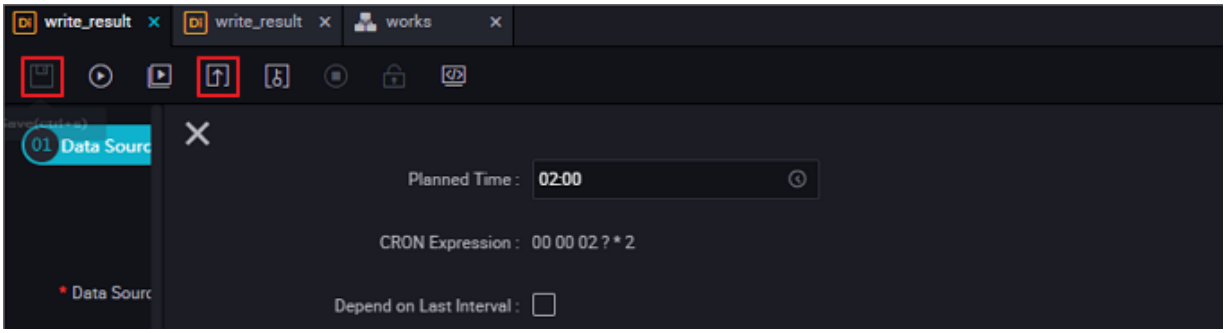
You can configure an upstream dependency for a task. In this way, even if the scheduled time of an instance of the current task is reached, the task can run only after the instance of its upstream task is completed.

The configuration in the preceding figure indicates that instances of the current task are 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 then, by default the current task is triggered by the project . 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 data synchronization task

Save the synchronization task **write_result** and click **Submit** to submit it to the scheduling system.



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

Note:
If the task is submitted after 23: 30, the scheduling system will automatically cycle-generate instances from the third day and run on time.

Subsequent steps

Now you know how to set the scheduling attribute and dependency of a synchronization task. Continue to the next tutorial. This tutorial will tell you how to perform periodic O&M for submitted tasks and view the log troubleshooting results. For more information, see *cycle care operations and check for log ranking errors*.

6 Step 5: O&M and view log troubleshooting results

This article will show you how to implement operations for tasks.

In the previous operations, you have set a synchronization task to run at 02:00 every Tuesday. After the task is submitted, you can view the automatic operation results in the scheduling system from the next day.

Now, how can we check whether the instance schedule and dependency are as expected?
To work this out, DataWorks provides three triggering methods: test run, data population, and periodic running, which are described 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 successful submission, the scheduling system automatically generates task instances at different time points starting from 00:00 of the next day. It checks whether upstream instances of each instance have run successfully according to 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 to both manual and automatic triggering modes.

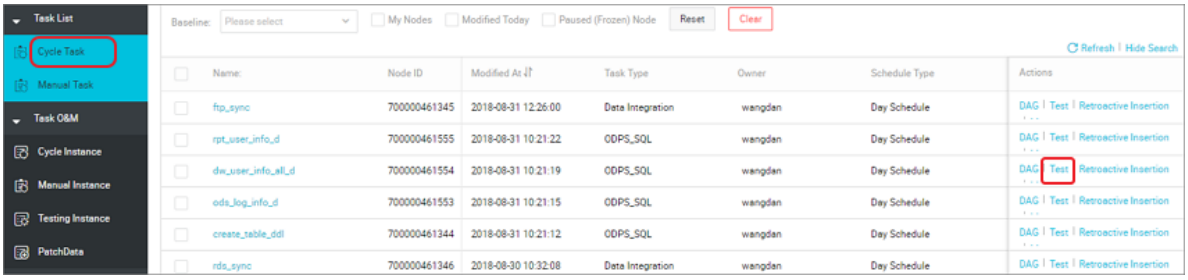
- The period can be set to monthly, weekly, daily, hourly, or even by minute. The scheduling system always generates an instance for the task on a specified day or at a specified time.
- The scheduling system regularly runs the instance on a specified date and generates operation logs.
- Instances rather than on a specified date does not run, and their statuses are directly changed to “Successful” if the running conditions are met. Therefore, no running logs are generated.

For more operational and functional instructions, see *Task operations*.

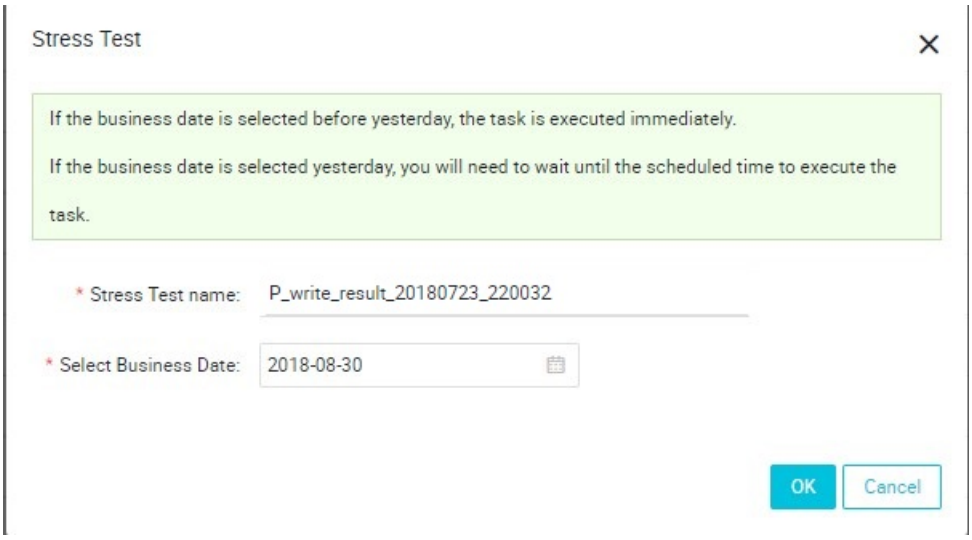
Test

Manually trigger a test

1. On the **Cycle Task** page, locate the task that you want to run, and click **Test**.



2. Enter the business Date and click **OK**.



3. Go to the **Basic information** page to view the task run status.

View the information and operation logs of the test instance

You can see the instance DAG graph by selecting the appropriate task instance in the **test instance** page and clicking.

- Right-click an instance, you can view the dependencies and details of this instance and perform specific actions such as stopping, rerunning, and so on..

- Double-click an instance to pop up task properties, run log, operation log, code, and so on.

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.
- The task write_result, Which is configured to run every Tuesday morning, is based on the instance generation rules described earlier in the article, the business date selected by the test Runtime is Monday (business date = run date-1), the instance will actually run at 2. If it is not Monday, the instance is converted to a successful state at 2 points, and there is no log generation.

replenishment data operation

Manually trigger data population

If you need to confirm the timing and interdependencies of multiple tasks, or you need to re-perform the data analysis calculation from a root task, you can select the **O&M center** > **task list** > **cycle task** page and click the **replenishment data** after the task, to fill multiple tasks for a certain period of time.

1. Select the **O&center** > **cycle task** page and enter the task name.
2. Click **replenishment data**after the query results.

3. Set the business date for the replenishment data as "to", select the write_result node task, and click **OK**.
4. Click to **view the replenishment data results**.

View the information and operation logs of the data population instance

You can see the instance DAG graph by selecting the appropriate task instance.

- Right-click an instance, you can view the dependencies and details of this instance and perform specific actions such as stopping, rerunning, and so on..
- Double-click an instance to pop up task properties, run log, operation log, code, and so on.

Note:

- 2017-09-18 15:56:30. 919 [job-51109647] In the figure above is the job ID.
- The task in the figure above failed because the source does not have this partition value in the synchronized table, so the read failed.
- The instance of a replenishment data task is day-to-day, for example, the task from 2017-09-15 to 2017-09-18 during this period, if the instance of number 15 fails, an instance of number 16 also does not run.
- The task write_result, which is configured to run every Tuesday morning, is based on the instance generation rules described earlier in the article. The business date selected by the replenishment data Runtime is Monday (business date = run date-1). The instance will actually run at 2 AM. If it is not Monday, the instance is converted to a successful state at 2 AM, and there is no log generation.

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. You can view the instance information and operation logs by using either of the following methods.

- Select the parameters such as the business date or the running date on the **O&M center** > **cycle instance** page, search for the instance that corresponds to the write_result task, and then right-click on the instance information and the run log.

- You can see the instance DAG graph by selecting the appropriate task instance in the **cycle instance** page and clicking.
  - Right-click an instance, you can view the dependencies and details of this instance and perform specific actions such as stopping, rerunning, and so on..
  - Double-click an instance to pop up task properties, run log, operation log, code, and so on.

Note:

- The task is not running because the upstream task is not running.
- If the initial state of an instance of a task is “Not Run”, when the scheduled time arrives , the scheduling system checks that all upstream instances of this instance are running successfully.
- The instance will be 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.