ALIBABA CLOUD

Alibaba Cloud

DataWorks

Quick Start

Document Version: 20201026

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 because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should 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 this document based on the "status quo", "being defective", and "existing functions" of its products and services. 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 take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, 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 contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, 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 this document 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 directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice 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.
Warning	A warning notice 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 restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	Notice: If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.Overview

Quick Start guides you through a complete process of data analytics and O&M.

DataWorks Data analytics O&M

> Note
>
> - If you are using DataWorks for the first time, make sure that you have completed preparations. For example, you must get your account ready and set members and roles for your workspace. After completing the preparations, you can log on to the DataWorks console, find the target workspace, and click **Data Analytics** in the Actions column to start data analytics.
> - This document describes the data analytics and O&M operations in a workspace in standard mode. The operations in a workspace in basic mode are basically the same as those in a workspace in standard mode. The only difference is that you can only commit nodes to the production environment in a workspace in basic mode.

Generally, you can complete the following data analytics and O&M operations in a workspace of DataWorks:

1. Create tables and import data
2. Create a workflow
3. Create a batch synchronization node
4. Configure recurrence and dependencies for a node
5. Run a node and troubleshoot errors
6. Optional: Use an ad hoc query node to run SQL statements

The following figure shows the basic process of data analytics and O&M.

2.Create tables and import data

This topic takes the bank_data and result_table tables as an example to describe how to create tables and import data in the DataWorks console.

Prerequisites

A MaxCompute compute engine is bound to the workspace where you want to create tables. The MaxCompute service is available in a workspace only after you bind a MaxCompute compute engine to the workspace on the **Workspace Management** page. For more information, see Configure a workspace.

Context

The bank_data table stores business data and the result_table table stores data analytics results.

Create the bank_data table

1. Go to the **DataStudio** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. In the top navigation bar, select the region where your workspace resides, find the workspace, and then click **Data Analytics** in the Actions column.
2. On the Data Development tab, move the pointer over the icon and choose **MaxCompute > Table**. Alternatively, you can click a workflow in the Business process section, right-click **MaxCompute**, and then choose **New > Table**.
3. In the **New table** dialog box, set **Table name** to bank_data and click **Submit**.

   > **Notice**
   > - The table name must be 1 to 64 characters in length. It must start with a letter and cannot contain special characters.
   > - If multiple MaxCompute compute engines are bound to the current workspace, you must select one from the Please select an Engine type drop-down list.

4. On the table configuration tab, click **DDL mode**.
5. In the **DDL mode** dialog box, enter the following statement and click **Generate table structure**:

```
CREATE TABLE IF NOT EXISTS bank_data
(
 age           BIGINT COMMENT 'age',
 job           STRING COMMENT 'job type',
 marital       STRING COMMENT 'marital status',
 education     STRING COMMENT 'education level',
 default       STRING COMMENT 'credit card',
 housing       STRING COMMENT 'mortgage',
 loan          STRING COMMENT 'loan',
 contact       STRING COMMENT 'contact',
 month         STRING COMMENT 'month',
 day_of_week   STRING COMMENT 'day in a week',
 duration      STRING COMMENT 'duration',
 campaign      BIGINT COMMENT 'number of contacts during the campaign',
 pdays         DOUBLE COMMENT 'interval from the last contact',
 previous      DOUBLE COMMENT 'number of contacts with the customer',
 poutcome      STRING COMMENT 'result of the previous marketing campaign',
 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 'whether time deposit is available'
);
```

For more information about the SQL syntax for creating tables, see Create and view a table.

6. In the **Confirm operation** message, click **OK**.
7. Set the **Chinese name** parameter in the **Basic properties** section and click **Submit to development environment** and **Submit to production environment**.

> **Note** This topic uses a workspace in standard mode as an example. If you are using a workspace in basic mode, you only need to click **Submit to production environment**.

8. In the left-side navigation submenu, click the **Table Management** icon.
9. On the **Table Management** tab, double-click the name of the created table to view the table information.

Create the result_table table

1. On the **Data Development** tab, move the pointer over the icon and choose **MaxCompute > Table**. Alternatively, you can click a workflow in the Business process section, right-click **MaxCompute**, and then choose **New > Table**.
2. In the **New table** dialog box, set **Table name** to result_table and click **Submit**.

3. On the table configuration tab, click DDL mode. In the **DDL mode** dialog box, enter the following statement and click **Generate table structure**:

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

4. In the **Confirm operation** message, click **OK**.
5. Set the **Chinese name** parameter in the **Basic properties** section and click **Submit to development environment** and **Submit to production environment**.
6. In the left-side navigation submenu, click the **Table Management** icon.
7. On the **Table Management** tab, double-click the name of the created table to view the table information.

Upload a local file to import its data to the bank_data table

You can perform the following operations in the DataWorks console:

- Upload a local text file to import its data to a table in a workspace.
- Use Data Integration to import business data from different data stores to a workspace.

> **Note** Comply with the following rules when you upload a local file:
> - File format: The file must be in the .txt, .csv, or .log format.
> - File size: The size of the file cannot exceed 30 MB.
> - Destination object: The destination object can be a partitioned table or a non-partitioned table. The partition key value cannot be in Chinese.

To upload the local file banking.txt to DataWorks, perform the following steps:

1. Click the icon on the **Data Development** tab.
2. In the **Data import wizard** dialog box, enter at least three letters to search for tables, select the table to which you want to import data, and then click **Next Step**.
3. In the dialog box that appears, set the **Select data import method** parameter to **Upload local files** and click **Browse** next to **Select File**. Select the local file that you want to upload and specify other parameters.

Parameter	Description
Select data import method	The method of importing data. Default value: **Upload local files**.
Select File	The file to upload. To upload a file, click **Browse** and select the local file to upload.

Parameter	Description
Select separator	The delimiter of fields in the file. Valid values: **Comma**, **Tab**, **SEMICOLON**, **Space**, **|**, **#**, and **&**. In this example, select **Comma**.
Original character set	The character set of the file. Valid values: **GBK**, **UTF-8**, **CP936**, and **ISO-8859**. In this example, select **GBK**.
Import start row	The row from which data is to be imported. In this example, select **1**.
First behavior title	Specifies whether to use the first row as the header row. In this example, do not select **First behavior title**.
Data preview	The preview of the data to be imported. **Note** If the data volume is large, only the data in the first 100 rows and 50 columns appears.

4. Click **Next Step**.
5. Select a matching mode for the fields in the source file and destination table. In this example, select **Match by location**.
6. Click **Import data**.

What to do next

Now you have learned how to create tables and import data. You can proceed with the next tutorial. In the next tutorial, you will learn how to create, configure, and commit a workflow and then use the Data Analytics feature to further compute and analyze data in the workspace. For more information, see Create a workflow.

3.Create a workflow

This topic describes how to create a workflow, create nodes in the workflow, and configure the dependencies among the nodes. After the configuration is completed, you can use the Data Analytics feature to further compute and analyze data in the workspace.

Prerequisites

The bank_data table for storing business data and the result_table table for storing data analytics results are created in a workspace. Data is imported to the bank_data table. For more information, see Create tables and import data.

Context

The Data Analytics feature of DataWorks allows you to drag nodes in a workflow and configure dependencies among the nodes. You can process data and configure dependencies in the data based on the workflow. You can create multiple workflows in a workspace. For more information, see Manage workflows.

Create a workflow

1. Log on to the DataWorks console.
2. In the left-side navigation pane, click **Workspaces**.
3. In the top navigation bar, select the region where the target workspace resides. Find the target workspace and click **Data Analytics** in the Actions column.
4. On the **Data Development** tab, move the pointer over the icon and select **Business process**.
5. In the **New business process** dialog box, set the **Business Name** and **Description** parameters.

   > **Notice** The workflow name must be 1 to 128 characters in length and can contain letters, digits, underscores (_), and periods (.). It is not case-sensitive.

6. Click **New**.

Create nodes and configure dependencies among the nodes

This section describes how to create a zero load node named start and an ODPS SQL node named insert_data in the workflow, and configure the insert_data node to depend on the start node.

> **Notice**
>
> - A zero load node is a control node that is used to maintain and control its descendant nodes in a workflow. A zero load node does not generate data.
> - If other nodes depend on a zero load node and the zero load node is set to Failed by an O&M expert, the pending descendant nodes cannot be triggered. During the O&M process, an O&M expert can disable a zero load node to prevent errors of ancestor nodes from being further expanded.
> - Typically, the root node of the workspace is used as the ancestor node of a zero load node in a workflow. The root node of a workspace is named in the `Workspace name_root` format.

We recommend that you create a zero load node as the root node of a workflow to control the entire workflow. To create nodes and configure dependencies among the nodes in a workflow, perform the following steps:

1. Double-click the name of the created workflow. On the configuration tab of the workflow, click **Virtual node** under Universal in the left-side navigation tree.

   

2. In the **New node** dialog box, set **Node name** to start and click **Submit**.

   > **Notice** The node name must be 1 to 128 characters in length and can contain letters, digits, underscores (_), and periods (.). It is not case-sensitive.

3. Use the same method to create an ODPS SQL node named insert_data.
4. Drag a directed line to configure the start node as the parent node of the insert_data node.

Configure the parent node of the zero load node

In a workflow, a zero load node is often used to control the entire workflow and serves as the ancestor node of all nodes in the workflow.

Generally, a zero load node in a workflow depends on the root node of the workspace. To configure the root node of a workspace as the parent node of a zero load node, perform the following steps:

1. Double-click the name of the zero load node.
2. On the configuration tab of the zero load node, click **Scheduling configuration** in the right-side navigation pane.
3. In the **Scheduling dependency** section, click **Use workspace root node**.
4. Click the  icon in the toolbar.

Configure and run the ODPS SQL node

This section provides a sample SQL statement that is used to query and save the number of singles with different education levels who loan to buy houses in the ODPS SQL node named insert_data. The query result can be analyzed by and presented in descendant nodes of insert_data.

1. Go to the configuration tab of the ODPS SQL node and enter the following SQL statement.For

more information about the SQL syntax, see SQL overview.

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

2. Right-click bank_data in the code and select **Cut**.

3. Click the  icon in the toolbar.

4. Click the  icon in the toolbar.After the node is run, you can view the operational log and running result in the lower part of the tab.

Commit the workflow

1. After you run and debug the ODPS SQL node insert_data, return to the configuration tab of the workflow.
2. Click the  icon in the toolbar.
3. In the **Submit** dialog box, select the nodes to be committed, enter your comments in the **Change description** field, and then select **Ignore alarms with inconsistent input and output**.
4. Click **Submit**.

What to do next

Now you have learned how to create and commit a workflow. You can proceed with the next tutorial. In the next tutorial, you will learn how to create a sync node to export data to different types of data stores. For more information, see Create a sync node.

4.Create a sync node

This topic describes how to create a sync node to export data from MaxCompute to a MySQL data store.

Prerequisites

An ApsaraDB RDS for MySQL instance is created. The ID of the ApsaraDB RDS for MySQL instance is obtained. A whitelist is configured for the instance in the ApsaraDB for RDS console. The address information about the resource group on which the sync node to create will be run is added to the whitelist. For more information, see Create an ApsaraDB RDS for MySQL instance.

> **Note** If you use a custom resource group to run the sync node, you must add the IP addresses of the servers in the custom resource group to the whitelist of the ApsaraDB RDS for MySQL instance.

Context

You can use Data Integration in DataWorks to periodically transfer the business data generated in a business system to a workspace. After the data is computed in SQL nodes, Data Integration periodically exports the computing results to your specified data store for further display or use.

Data Integration can import data from and export data to various data stores, such as Relational Database Service (RDS), MySQL, SQL Server, PostgreSQL, MaxCompute, Memcache, Distribute Relational Database Service (DRDS), Object Storage Service (OSS), Oracle, FTP, Dameng, Hadoop Distributed File System (HDFS), and MongoDB. For more information about the data stores, see Supported data stores and plug-ins.

Add a connection

> **Note** Only the workspace administrator can add connections, and members of other roles can only view the connections.

1. Go to the **Data Source** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. In the top navigation bar, select the region where the target workspace resides. Find the target workspace and click **Data Integration** in the Actions column.
   iv. On the page that appears, click **Connection** in the left-side navigation pane. The **Data Source** page appears.
2. On the **Data Source** page, click **Add a Connection** in the upper-right corner.
3. In the **Add data source** dialog box, click **MySQL** in the Relational Database section.
4. In the **Add MySQL data source** dialog box, set the parameters as required.In this example, set Data source type to **Alibaba Cloud instance mode**. Then, set other parameters as described in the following table.

5. Click **Test Connection** in the Actions column of each resource group. A sync node only uses one resource group. Therefore, you must test the connectivity of all the resource groups for Data Integration that your sync nodes use to connect to the data store so that sync nodes can run properly. For more information, see Test data store connectivity.
6. After the connection passes the connectivity test, click **Complete**.

Verify that a table exists in the destination MySQL data store

Run the following statement to create the odps_result table in the MySQL data store:

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

After the table is created, run the `desc odps_result;` statement to view the table details.

Create and configure a sync node

This section describes how to create and configure a sync node named write_result and use the node to export data in the result_table table to your MySQL data store. To create and configure a sync node, perform the following steps:

1. Go to the **DataStudio** page and create a sync node named write_result.
2. Configure the insert_data node as the parent node of the write_result node.
3. In the Select data source section, set **Data source** and **Table** below Data source to **ODPS** > **odps_first** and result_table, respectively.
4. Set Data source below Data destination to MySQL > odps_result.
5. In the Filed Mapping section, configure field mappings. Make sure that fields in the source table on the left have a one-to-one mapping with fields in the destination table on the right.
6. In the **Channel control** section, configure the maximum transmission rate and dirty data check rules.

Parameter	Description
Maximum number of concurrent tasks expected	The maximum number of concurrent threads that the batch sync node uses to read data from the source data store and write data to the destination data store. You can configure the concurrency for the sync node on the codeless user interface (UI).

Parameter	Description
Synchronization rate	Specifies whether to enable bandwidth throttling. You can enable bandwidth throttling and set a maximum transmission rate to avoid heavy read workload of the source. We recommend that you enable bandwidth throttling and set the maximum transmission rate to a proper value.
The number of error records exceeds	The maximum number of dirty data records allowed.

7. Preview and save the configuration. After you complete the configuration, scroll up and down to view the node configuration. After you confirm that the configuration is correct, click the icon in the toolbar.

Commit the sync node

Return to the workflow after you save the sync node. Click the icon in the toolbar to commit the sync node to the scheduling system. The scheduling system automatically runs the node at the scheduled time from the next day based on your settings.

What to do next

Now you have learned how to create a sync node to export data to a specific data store. You can proceed with the next tutorial. In the next tutorial, you will learn how to configure recurrence and dependencies for a sync node. For more information, see Configure recurrence and dependencies for a node.

5.Configure recurrence and dependencies for a node

This topic describes how to configure recurrence and dependencies for a node in DataWorks. The sync node write_result that is scheduled by week is used as an example.

Prerequisites

The sync node write_result is created. For more information, see Create a sync node.

Context

DataWorks has a powerful scheduling engine to trigger nodes based on the recurrence and dependencies of the nodes. DataWorks ensures that tens of millions of nodes run accurately and punctually per day based on directed acyclic graphs (DAGs). In the DataWorks console, you can set the recurrence to minutely, hourly, daily, weekly, or monthly. For more information, see Time properties.

Configure recurrence for the sync node

1. Go to the **DataStudio** page.
   i. Log on to the DataWorks console.
   ii. In the left-side navigation pane, click **Workspaces**.
   iii. In the top navigation bar, select the region where your workspace resides, find the workspace, and then click **Data Analytics** in the Actions column.
2. Find the workflow to which the sync node write_result belongs and double-click the sync node.
3. On the configuration tab of the node, click **Scheduling configuration** in the right-side navigation pane.

   > **Note** In a manually triggered workflow, all nodes must be manually triggered, and cannot be automatically scheduled by DataWorks.

4. In the **Time attribute** section, set the parameters as required.

Parameter	Description
How to generate an instance	The time to generate the first instance. Valid values: **T +1 generated the next day** and **Generate immediately after publishing**.
Time attribute	The mode in which the node is run. Valid values: **Normal Scheduling** and **Empty run scheduling**.
Rerun attribute	Specifies whether to allow the node to be rerun. Valid values: **Run again after success or failure**, **Do not re-run after successful operation, and re-run after failed operation**, and **Do not rerun after successful or failed operation**.

Parameter	Description
Error automatic rerun	Specifies whether to automatically rerun the node when an error occurs. This parameter appears only if the **Rerun attribute** parameter is set to **Run again after success or failure** or **Do not re-run after successful operation, and re-run after failed operation**. After you select this check box, the node is automatically rerun when an error occurs. This parameter does not appear if you set the Rerun attribute parameter to **Do not rerun after successful or failed operation**. In this case, the node is not rerun when an error occurs.
Effective Date	The validity period of the node. Specify the start and end dates of the validity period as required.
Suspend scheduling	Specifies whether to skip execution of the node.
Scheduling cycle	The recurrence of the node. Valid values: Minutes, Hours, Day, Week, and Month. In this example, set the value to Week.
Timing scheduling	Specifies whether to periodically schedule the node. This check box is selected by default.
Specify time	The time when the node is run. For example, you can configure the node to run at 02:00 every Tuesday.
cron expression	The CRON expression of the time you specified, which cannot be changed.
Rely on previous cycle	Specifies whether the node depends on the result of the last cycle.

Configure dependencies for the sync node

After you configure the recurrence for the sync node write_result, you can continue to configure dependencies for the sync node.

You can configure the parent node on which the sync node depends. After that, the scheduling system triggers the sync node only after the instance of the parent node is run.

For example, the instance of the sync node is not triggered until the instance of its parent node insert_data is run.

By default, the scheduling system creates a node named in the format of Workspace name_root for each workspace as the root node. If no parent node is configured for the sync node, the sync node depends on the root node.

Commit the sync node

1. On the configuration tab of the write_result node, click the icon in the toolbar.
2. Commit the node.

> **Notice** You must set the **Rerun attribute** and **Dependent upstream node** parameters before you can commit the node.

i. Click the icon in the toolbar.

ii. In the **Submit New version** dialog box, enter your comments in the **Change description** field.

iii. Click **OK**.

In a workspace in standard mode, you must click **Publish** in the upper-right corner after you commit the sync node.

A node must be committed to the scheduling system so that the scheduling system can automatically generate and run instances for the node. The scheduling system runs these instances at the specified time from the next day based on the recurrence settings.

> **Note** If you commit a node after 23:30, the scheduling system automatically generates and runs instances for the node from the third day.

What to do next

Now you have learned how to configure recurrence and dependencies for a sync node. You can proceed with the next tutorial. In the next tutorial, you will learn how to perform O&M on the committed node and troubleshoot errors based on the operational logs. For more information, see Run a node and troubleshoot errors.

6.Run a node and troubleshoot errors

This topic describes how to run and maintain a node, and troubleshoot errors based on logs.

When you configure recurrence and dependencies for the sync node write_result, you have configured the sync node to run at 02:00 every Tuesday. After you commit this node, you need to wait until the next day to view the automatic execution result of this node. DataWorks allows you to run nodes in the following modes: test run, retroactive run, and periodic run. This helps you confirm the run time of each node instance, dependencies among node instances, and whether generated data is as expected.

- Test run: Nodes are manually triggered. We recommend that you use this mode if you need to check the run time and running of only one node.
- Retroactive run: Nodes are manually triggered. We recommend that you use this mode if you need to check the run time of multiple nodes and dependencies among them, or if you need to reperform data analysis and computing from the specific root node.
- Periodic run: Nodes are automatically triggered. After you commit a node, the scheduling system automatically generates and runs instances for the node from 00:00 the next day. When the scheduled time of each instance arrives, the scheduling system checks whether the ancestor instances of the instance have been run. If all the ancestor instances have been run, the scheduling system automatically triggers the instance without manual intervention.

> **Note** The scheduling system generates instances for manually triggered nodes and auto triggered nodes based on the same rules.
>
> - The scheduling system generates instances of a node for each date within the validity period of a node, regardless whether the recurrence of the node is set to minutely, hourly, daily, weekly, or monthly.
> - The scheduling system runs the instances generated for the specified run dates only when the scheduled time arrives and generates operational logs for the instances.
> - The scheduling system does not run the instances generated for other dates. Instead, it changes the status of the instances to successful when the running conditions are met.

Test run

1. On the DataStudio page, click the icon in the upper-left corner and choose **All Products > Operation Center** to go to the **Operation Center** page.
2. In the left-side navigation pane, choose **Cycle Task Maintenance > Cycle Task**.
3. Find the node that you want to test and click **Test** in the Actions column.
4. In the **Smoke Test** dialog box, set the **Smoke Test Instance Name** and **Data Timestamp** parameters and click **OK**.
5. On the **Test Instance** page, click the name of the generated instance. The directed acyclic graph (DAG) of the instance appears on the right.

   Right-click the instance node in the DAG to view its dependencies and details, and stop or rerun this instance.

> **Note**
>
> - In test run mode, a node is manually triggered. When the scheduled time arrives, the scheduling system runs the corresponding instance immediately, no matter whether the ancestor instances have been run.
> - The sync node write_result is configured to run at 02:00 every Tuesday. Based on the instance generation rules described earlier in this topic, if the data timestamp, which is one day before the run date, is set to Monday for a test run, the scheduling system runs the instance for the sync node write_result at 02:00 on Tuesday. If the data timestamp is not set to Monday for the test run, the scheduling system changes the status of the instance to successful at 02:00 on Tuesday with no operational logs generated.

Retroactive run

A retroactive run is recommended if you need to check the run time of multiple nodes and dependencies among them, or if you need to reperform data analysis and computing from the specific root node.

1. On the **Operation Center** page, choose **Cycle Task Maintenance** > **Cycle Task** in the left-side navigation pane.
2. Find the node for which you want to generate retroactive data and choose **Patch Data** > **Current Node Retroactively** in the Actions column.
3. In the **Patch Data** dialog box, set the parameters and click **OK**.

Parameter	Description
Retroactive Instance Name	The name of the retroactive instance.
Data Timestamp	The data timestamp of the retroactive instance. The retroactive instance is run on the next day of the specified timestamp.
Node	The node for which retroactive data will be generated. The default value is the current node, which cannot be changed.
Parallelism	Specifies whether to concurrently run the node with other nodes. Select **Disable** or specify several nodes to run concurrently.

4. On the **Patch Data** page, click the name of the generated retroactive instance to view the DAG of the instance.

   Right-click the instance node in the DAG to view its dependencies and details, and stop or rerun this instance.

> Note
>
> - In retroactive run mode, the running of an instance requires the instance running result of the previous day. For example, in the scenario in which you configure retroactive instances to run from September 15, 2017 to September 18, 2017, if the instance on September 15 fails to run, the instance on September 16 cannot be run.
> - The sync node write_result is configured to run at 02:00 every Tuesday. Based on the instance generation rules described earlier in this topic, if the data timestamp, which is one day before the run date, is set to Monday for a retroactive instance, the scheduling system runs the instance for the sync node write_result at 02:00 on Tuesday. If the data timestamp is not set to Monday for the retroactive instance, the scheduling system changes the status of the instance to successful at 02:00 on Tuesday with no operational logs generated.

Periodic run

In periodic run mode, the scheduling system automatically triggers instances for all nodes based on the scheduling configuration. No menu item is provided for you to control the periodic run on the Operation Center page. You can view the instance information and operational logs of a node, for example, write_result, by using one of the following methods:

- On the **Operation Center** page, choose **Cycle Task Maintenance > Cycle Instance** in the left-side navigation pane. On the page that appears, set parameters such as the data timestamp or run date to search for a specific instance of the node. Then, right-click the instance node in the DAG to view the instance information and operational logs.
- On the **Cycle Instance** page, click an instance of the node. The DAG of the instance appears.

  Right-click the instance node in the DAG to view its dependencies and details, and stop or rerun this instance.

  > Note
  >
  > - If an ancestor node has not been run, its descendant nodes are not run either.
  > - If the initial status of an instance is pending, the scheduling system checks whether all its ancestor instances have been run when the scheduled time arrives.
  > - The instance can be triggered and run only after all its ancestor instances have been run and the scheduled time arrives.
  > - If an instance is pending, check whether all its ancestor instances have been run and whether the scheduled time arrives.

7.Optional: Use an ad hoc query node to run SQL statements

You can use the ad hoc query feature provided by DataStudio to quickly run SQL statements in the MaxCompute project associated with your DataWorks workspace.

For more information about the **ad hoc query** feature, see Create an ad hoc query.

Create an ad hoc query node

1. Log on to the DataWorks console. In the left-side navigation pane, click Workspaces. On the Workspaces page, find the target workspace and click **Data Analytics** in the Actions column.
2. Click the **Ad-Hoc Query** icon in the left-side navigation pane. The **Ad-Hoc Query** tab appears.
3. Right-click **Ad-Hoc Query** and choose **Create Node > ODPS SQL**.
4. In the **Create Node** dialog box that appears, set **Node Name** and **Location**.
5. Click **Commit**.

Run SQL statements

After committing the ad hoc query node, you can run SQL statements supported by MaxCompute in the node. For more information, see SQL statement overview.

For example, to create a table, enter the following statement and click :

```
create table if not exists sale_detail
(
shop_name     string,
customer_id   string,
total_price   double
)
partitioned by (sale_date string,region string);
-- Create a partitioned table named sale_detail.
```

In the Expense Estimate dialog box that appears, check the estimated expense of running the SQL statement that you enter, and click **Run**.

View the running status and results in the Runtime Log section. If the SQL statement is run successfully, the result is shown as **OK**.

You can run SQL query statements in the same way.