Data Lake Insight

User Guide

Issue	01
Date	2022-12-06

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 DLI Introduction

What Is DLI?

Data Lake Insight (DLI) is a serverless data processing and analysis service fully compatible with **Apache Spark** and **Apache Flink** ecosystems. It frees you from managing any server. DLI supports standard SQL and is compatible with Spark and Flink SQL. It also supports multiple access modes, and is compatible with mainstream data formats. DLI supports SQL statements and Spark applications for heterogeneous data sources, including CloudTable, RDS, DWS, CSS, OBS, custom databases on ECSs, and offline databases.

Advantages

- Full SQL compatibility

  You do not require a background in big data to do big data analysis. You only need to know SQL, and you are good to go. The SQL syntax is fully compatible with the standard ANSI SQL 2003.

- Storage-compute decoupling

  DLI decouples storage from computing so that you can use lower costs while improving resource utilization.

- O&M-free and high availability

  DLI frees you from the hassle of complicated O&M and upgrade operations while you enjoy high data availability with dual-AZ deployment.

DLI Core Engine: Spark+Flink

- Spark is a unified analysis engine that is ideal for large-scale data processing. It focuses on query, compute, and analysis. DLI optimizes performance and reconstructs services based on open-source Spark. It is compatible with the Apache Spark ecosystem and interfaces, and improves performance by 2.5x when compared with open-source Spark. In this way, DLI enables you to perform query and analysis of EB's of data within hours.
- Flink is a distributed compute engine that is ideal for batch processing, that is, for processing static data sets and historical data sets. You can also use it for stream processing, that is, processing real-time data streams and generating data results in real time. DLI enhances features and security based on the open-source Flink and provides the Stream SQL feature required for data processing.

Serverless Architecture

DLI is a serverless big data query and analysis service. It has the following advantages:

- Auto scaling: DLI ensures you always have enough capacity on hand to deal with any traffic spikes.

2 Getting Started

2.1 Creating and Submitting a Spark SQL Job

You can use DLI to submit a Spark SQL job to query data. The general procedure is as follows:

Step 1: Logging in to the Cloud Platform

Step 2: Uploading Data to OBS

Step 3: Logging In to the DLI Management Console

Step 4: Creating a Queue

Step 5: Creating a Database

Step 6: Creating a Table

Step 7: Querying Data

The following illustrates how to query OBS data using DLI. Operations to query DLI data are similar.

Step 1: Logging in to the Cloud Platform

1. Open the DLI homepage.
2. On the login page, enter the username and password, and click **Log In**.

Step 2: Uploading Data to OBS

DLI allows you to query data stored on OBS. Before querying the data, you need to upload data to OBS.

1. In the services displayed, click **Object Storage Service (OBS)** in **Storage**.
2. The OBS console page is displayed.
3. Create a bucket. The bucket name must be globally unique. In this example, assume that the bucket name is **obs1**.
    a. Click **Create Bucket** in the upper right corner.

b. On the displayed **Create Bucket** page, enter the **Bucket Name**. Retain the default values for other parameters or set them as required.

c. Click **Create Now**.

4. Click **obs1** to switch to the **Overview** page.
5. In the left navigation pane, click **Objects**. Click **Upload Object**. In the displayed dialog box, drag files or folders to upload or add file, for example, **sampledata.csv** to the file upload box. Then, click **Upload**.

   You can create a **sampledata.txt** file, copy the following content separated by commas (,), and save the file as **sampledata.csv**.

   ```
   12,test
   ```

   After the file is uploaded successfully, the file path is **obs://obs1/sampledata.csv**.

   For more information about OBS operations, see the *Object Storage Service Console Operation Guide*.

   For more information about the tool, see the *OBS Tool Guide*.

   NOTE

   You are advised to use an OBS tool, such as OBS Browser+, to upload large files because OBS Console has restrictions on the file size and quantity.

   - OBS Browser+ is a graphical tool that provides complete functions for managing your buckets and objects in OBS.

Step 3: Logging In to the DLI Management Console

1. .
2. The DLI management console page is displayed. If you log in to the DLI management console for the first time, you need to be authorized to access OBS.

   Note: For this guide, you need at least the **Tenant Administrator (Global service)** permission.

Step 4: Creating a Queue

A queue is the basis for using DLI. Before executing an SQL job, you need to create a queue.

- An available queue **default** is preset in DLI.
- You can also create queues as needed.

  a. On the DLI management console, click **SQL Editor** in the navigation pane on the left. The **SQL Editor** page is displayed.

  b. On the left pane, select the **Queues** tab, and click ⊕ next to **Queues**.

  For details, see Creating a Queue.

Step 5: Creating a Database

Before querying data, create a database, for example, **db1**.

NOTE

The **default** database is a built-in database. You cannot create the database named **default**.

1. On the DLI management console, click **SQL Editor** in the navigation pane on the left. The **SQL Editor** page is displayed.
2. In the editing window on the right of the **SQL Editor** page, enter the following SQL statement and click **Execute**. Read and agree to the privacy agreement, and click **OK**.

```
create database db1;
```

   After database **db1** is successfully created, **db1** will be displayed in the **Database** list.

   NOTE

   When you execute a query on the DLI management console for the first time, you need to read the privacy agreement. You can perform operations only after you agree to the agreement. For later queries, you will not need to read the privacy agreement again.

Step 6: Creating a Table

After database **db1** is created, create a table (for example, **table1**) containing data in the sample file **obs://obs1/sampledata.csv** stored on OBS in **db1**.

1. In the SQL editing window of the **SQL Editor** page, select the **default** queue and database **db1**.
2. Enter the following SQL statement in the job editor window and click **Execute**:

```
create table table1 (id int, name string) using csv options (path 'obs://obs1/sampledata.csv');
```

   After the table is created, click the **Databases** tab and then select **db1**. The created table **table1** is displayed in the **Table** area.

Step 7: Querying Data

After performing the preceding steps, you can start querying data.

1. In the **Table** tab on the **SQL Editor** page, double-click the created table **table1**. The SQL statement is automatically displayed in the SQL job editing window in the right pane. Run following statement to query 1,000 records in the **table1** table:

```
select * from db1.table1 limit 1000;
```

2. Click **Execute**. The system starts the query.

   After the SQL statement is executed successfully, you can view the query result in **View Result** under the SQL job editing window.

2.2 Developing and Submitting a Spark SQL Job Using the TPC-H Sample Template

To facilitate SQL operations, DLI allows you to customize query templates or save the SQL statements in use as templates. After templates are saved, you do not need to write SQL statements. You can directly perform the SQL operations using the templates.

The current system provides various standard TPC-H query statement templates. You can select a template as needed. This example shows how to use a TPC-H template to develop and submit a Spark SQL job.

Step 1: Logging in to the Cloud

Step 2: Logging In to the DLI Management Console

Step 3: Execute the TPC-H Sample Template and View the Result

For details about the templates, see SQL Template Management.

Step 1: Logging in to the Cloud

1. Open the DLI homepage.
2. On the login page, enter the username and password, and click **Log In**.

Step 2: Logging In to the DLI Management Console

1. .
2. The DLI management console page is displayed.

NOTE

You need to perform authorization when accessing the DLI management console for the first time. For details, see Service Authorization.

Step 3: Execute the TPC-H Sample Template and View the Result

1. On the DLI management console, choose **Job Templates** > **SQL Templates**, and click the **Sample Templates** tab. Locate the **Q1_Price_summary_report_query** template under **tpchQuery**, and click **Execute** in the **Operation** column. The **SQL Editor** page is displayed.
2. In the upper part of the editing window, set **Engine** to **spark**, **Queues** to **default**, and **Databases** to **default**, and click **Execute**.
3. View the query result in the **View Result** tab in the lower part of the SQL Editor page.

This example uses the **default** queue and database preset in the system as an example. You can also run query statements on a self-created queue and database.

For details about how to create a queue, see Creating a Queue. For details about how to create a database, see Creating a Database.

2.3 Creating and Submitting a Spark Jar Job

You can use DLI to submit Spark jobs for real-time computing. The general procedure is as follows:

Step 1: Logging in to the Cloud

Step 2: Uploading Data to OBS

Step 3: Logging In to the DLI Management Console

Step 4: Creating a Queue

Step 5: Creating a Package

Step 6: Submitting a Spark Job

Step 1: Logging in to the Cloud

To use DLI, you need to log in to the cloud.

1. Open the DLI home page.
2. On the login page, enter the username and password, and click **Log In**.

Step 2: Uploading Data to OBS

Write a Spark Jar job program , and compile and package it as **spark-examples.jar**. Perform the following steps to submit the job:

Before submitting Spark Jar jobs, upload data files to OBS.

1. In the services displayed, click **Object Storage Service (OBS)** in **Storage**.
2. The OBS console page is displayed.
3. Create a bucket. The bucket name must be globally unique. In this example, assume that the bucket name is **dli-test-obs01**.
   a. Click **Create Bucket**.
   b. On the displayed **Create Bucket** page, enter the **Bucket Name**. Retain the default values for other parameters or set them as required.
   c. Click **Create Now**.
4. Click **dli-test-obs01** to switch to the **Overview** page.
5. From the left navigation tree, click **Objects**. Click **Upload Object**. In the displayed dialog box, drag files or folders to upload or add file to the file upload box, for example, **spark-examples.jar**. Then, click **Upload**.

   After the file is uploaded successfully, the file path is **obs://dli-test-obs01/spark-examples.jar**.

   For more information about OBS operations, see the *Object Storage Service Console Operation Guide*.

   For more information about the tool, see the *OBS Tool Guide*.

   NOTE

   You are advised to use an OBS tool, such as OBS Browser+, to upload large files because OBS Console has restrictions on the file size and quantity.

   - OBS Browser+ is a graphical tool that provides complete functions for managing your buckets and objects in OBS. You are advised to use this tool to create buckets or upload objects.

Step 3: Logging In to the DLI Management Console

To submit Spark jobs, you need to enter the Spark job creation page first.

1. In the list, choose .
2. The DLI management console page is displayed.

NOTE

If you log in to the DLI management console for the first time, you need to be authorized to access OBS. For details, see "Service Authorization".

Step 4: Creating a Queue

If you submit a Spark job for the first time, you need to create a queue first. For example, create a queue named **sparktest** and set **Queue Type** to **General Queue**.

1. In the navigation pane of the DLI management console, choose **Queue Management**.
2. In the upper right corner of the **Queue Management** page, click **Create Queue** to create a queue.
3. Create a queue, name it **sparktest**, and set the queue usage to for general purpose. For details, see Creating a Queue.
4. Click **Create Now** to create a queue.

Step 5: Creating a Package

Before submitting a Spark job, you need to create a package, for example, **spark-examples.jar**.

1. In the navigation pane on the left of the DLI console, choose **Data Management** > **Package Management**.
2. On the **Package Management** page, click **Create** in the upper right corner to create a package.
3. In the **Create Package** dialog box, set **Type** to **JAR**, **OBS Path** to the path of the spark-examples.jar package in **Step 2: Uploading Data to OBS**, and **Group** to **Do not use**.
4. Click **OK**.

   You can view and select the package on the **Package Management** page.

For details about how to create a package, see "Creating a Package".

Step 6: Submitting a Spark Job

1. On the DLI management console, choose **Job Management > Spark Jobs** in the navigation pane on the left. On the displayed page, click **Create Job**.
2. On the Spark job editing page, set **Queues** to the queue created in **Step 4: Creating a Queue** and **Application** to the package created in **Step 5: Creating a Package**.

   For details about other parameters, see the description of the Spark job editing page in "Creating a Spark Job".
3. Click **Execute** in the upper right corner of the Spark job editing window, read and agree to the privacy agreement, and click **OK**. Submit the job. A message is displayed, indicating that the job is submitted.
4. (Optional) Switch to the **Job Management > Spark Jobs** page to view the status and logs of the submitted Spark job.

NOTE

When you click **Execute** on the DLI management console for the first time, you need to read the privacy agreement. Once agreed to the agreement, you will not receive any privacy agreement messages for subsequent operations.

2.4 Creating and Submitting a Flink SQL Job

You can use DLI to submit Flink SQL jobs for real-time computing. The general procedure is as follows:

Step 1: Log In to the Cloud Platform

Step 2: Prepare Data Sources and Data Output Channels

Step 3: Create an OBS Bucket for Saving Outputs

Step 4: Logging In to the DLI Management Console

Step 5: Creating a Queue

Step 6: Creating an Enhanced Datasource Connection

Step 7: Creating a Datasource Authentication

Step 8: Configuring Security Group Rules and Testing Address Connectivity

Step 9: Creating a Flink SQL Job

The Flink SQL job you will create has an input and an output stream. The input stream is used to read data from DIS, and the output stream is used to write data to Kafka.

Step 1: Log In to the Cloud Platform

To use DLI, you need to log in to the cloud platform.

1. Open the DLI home page.
2. On the login page, enter the username and password, and click **Log In**.

Step 2: Prepare Data Sources and Data Output Channels

DLI Flink jobs allow other services to function as data sources and data output channels. For details, see"Preparing Data".

In this example, assume that the job name is **JobSample** and DIS is used as the data source. To enable DIS, see "Creating a DIS Stream" in the *Data Ingestion Service User Guide*. To use Kafka as the data output channel, you need to create a Kafka platinum instance. For details, see **Buying an Instance** in the *Distributed Message Service for Kafka User Guide*.

- Create a DIS stream for the job input stream.
  a. Log in to the DIS console.
  b. In the upper left corner of the management console, select the target region and project.
  c. On the **Overview** page, click **Buy Stream** and configure stream parameters. The channel information is as follows:

        - **Region**: Select the region where DLI is located.
        - **Stream Name**: **csinput**
        - **Stream Type**: **Common**
        - **Partitions**: **1**
        - **Data Retention (hours)**: **24**
        - **Source Data Type**: **BLOB**
        - **Auto Scaling**: **disabled**
        - **Enterprise project**: **default**
        - **Advanced Settings**: Skip it.

    d. Click **Buy Now**. The **Details** page is displayed.

    e. Click **Submit**.

- Create a Kafka platinum instance for the job output stream.

    a. Before creating a Kafka instance, ensure the availability of resources, including a virtual private cloud (VPC), subnet, security group, and security group rules.

        - For details about how to create a VPC and subnet, see "Creating a VPC and Subnet" in the *Virtual Private Cloud User Guide*. For details about how to create and use a subnet in an existing VPC, see "Create a subnet for the VPC" in the *Virtual Private Cloud User Guide*.

            NOTE

            - The created VPC and the Kafka instance that will be created must be in the same region.
            - Retain the default settings unless otherwise specified.

        - For details about how to create a security group, see "Creating a Security Group" in the *Virtual Private Cloud User Guide*. For details about how to add rules to a security group, see "Creating a Subnet for the VPC" in the *Virtual Private Cloud User Guide*.

        For more information, see **Managing Kafka Premium Instances** > **Preparing the Environment** in the *Distributed Message Service User Guide*.

    b. Log in to the DMS (for Kafka) console.

    c. Select a region in the upper left corner.

    d. On the **Kafka Instances** page, click **Buy Instance** in the upper right corner and set related parameters. The instance information is as follows:

        - **Region**: Select the region where DLI is located.
        - **Project**: Keep the default value.
        - **AZ**: Keep the default value.

- **Instance Name**: **kafka-dliflink**
- **Enterprise Project**: **default**
- **Version**: Keep the default value.
- **CPU Architecture**: Keep the default value.
- **Specifications**: Select the specification as needed.
- **Brokers**: Keep the default value.
- **Storage Space**: Keep the default value.
- **Capacity Threshold Policy**: Keep the default value.
- **VPC** and **Subnet**: Select the VPC and subnet created in **a**.
- **Security Group**: Select the security group created in **a**.
- **Manager Username**: dliflink (used to log in to the instance management page)
- **Password**: **** (The system cannot detect your password.)
- **Confirm Password**: ****
- **Advanced Settings**: Enable **Kafka SASL_SSL** and configure the username and password for SSL authentication as prompted. You do not need to set other parameters.

e. Click **Buy**. The confirmation page is displayed.

f. Click **Submit**.

g. On the DMS for Kafka console, click **Kafka Premium** and click the name of the Kafka instance, for example, **kafka-dliflink**. The instance details page is displayed.

h. Locate the SSL certificate in **Basic Information** > **Advanced Settings**, and click **Download**. Download the package to the local PC and decompress it to obtain the client certificate file **client.truststore.jks**.

Step 3: Create an OBS Bucket for Saving Outputs

In this example, you need to enable OBS for job JobSample to provide DLI Flink jobs with the functions of checkpoint, saving job logs, and commissioning test data.

For details about how to create a bucket, see "Creating a Bucket" in the *Object Storage Service Console Operation Guide*.

1. In the navigation pane on the OBS management console, select **Object Storage**.
2. In the upper right corner of the page, click **Create Bucket** and set bucket parameters.
   - **Region**: Select the region where DLI is located.

   - **Bucket Name**: Enter a bucket name.
   - **Default Storage Class**: **Standard**
   - **Bucket Policy**: **Private**
   - **Default Encryption**: **Do not enable**
   - **Direct Reading**: **Do not enable**
   - **Enterprise project**: **default**
   - **Tags**: Leave it blank.
3. Click **Create Now**.

Step 4: Logging In to the DLI Management Console

1. .
2. The DLI management console page is displayed. If you log in to the DLI management console for the first time, you need to be authorized to access OBS.

Step 5: Creating a Queue

You cannot create a DLI Flink SQL job on the existing default queue. Instead, you need to create a queue, for example, a queue named **Flinktest**. For details, see Creating a Queue.

1. On the **Overview** page of the DLI management console, click **Buy Queue** in the upper right corner.
2. Configure the following parameters:
   - **Name**: **Flinktest**
   - **Queue Usage**: Select **For general purpose** and **Dedicated Resource Mode**.
   - **CU Specifications**: **16 CUs**
   - **Enterprise project**: **default**
   - **Description**: Leave it blank.
   - **Advanced Settings**: **Custom**
   - **CIDR Block**: The configured CIDR block cannot conflict with the Kafka subnet CIDR block.
3. Click **Buy** to confirm the configuration.
4. Confirm the configuration and submit the request.

Step 6: Creating an Enhanced Datasource Connection

You need to create an enhanced datasource connection for the Flink job. For details, see "Creating an Enhanced Datasource Connection".

NOTE

- The CIDR block of the DLI queue bound with a datasource connection cannot overlap with the CIDR block of the data source.
- Datasource connections cannot be created for the **default** queue.
- To access a table across data sources, you need to use a queue bound to a datasource connection.

1. In the navigation pane of the DLI management console, choose **Datasource Connections**.
2. Click the **Enhanced** tab and click **Create**. Set the following parameters:
   - **Connection Name**: diskafka
   - **Bind Queue**: flinktest
   - **VPC**: vpc-dli
   - **Subnet**: dli-subnet

     NOTE

     The VPC and subnet must be the same as those of the Kafka instance.
3. Click **OK**.
4. On the **Enhanced** tab page, click the created connection **diskafka** to view its **VPC Peering ID** and **Connection Status**. If the status is **Active**, the connection is successful.

Step 7: Creating a Datasource Authentication

For details about how to create data source authentication, see "Datasource Authentication".

1. Upload the Kafka authentication file **client.truststore.jks** obtained in **Step 2: Prepare Data Sources and Data Output Channels** to the OBS bucket **smoke-test** created in **Step 3: Create an OBS Bucket for Saving Outputs**.
2. On the DLI management console, click **Datasource Connections**.
3. On the **Datasource Authentication** tab page, click **Create** to add an authentication information. Set the following parameters:
   - **Authentication Certificate**: Flink
   - **Type**: Kafka_SSL
   - **Truststore Path**: **obs://smoke-test/client.truststore.jks**
   - **Truststore password**: dms@kafka

   You do not need to set other parameters.
4. Click **OK**.

Step 8: Configuring Security Group Rules and Testing Address Connectivity

1. On the DLI management console, click **Resources** > **Queue Management**, select the bound queue, and click the arrow next to the queue name to view the network segment information of the queue.
2. Log in to the DMS for Kafka console, click **Kafka Premium**, and click the name of the created Kafka instance, for example, **kafka-dliflink**. The instance information page is displayed.
3. On the **Basic Information** page, obtain the Kafka connection address and port number in the **Connection Address** area.
4. On the **Basic Information** page, click the security group name in the **Security Group** area.

5. On the security group configuration page of the Kafka instance, choose **Inbound Rules** > **Add Rule**. Set **Protocol** to **TCP**, **Port** to **9093**, and **Source** to the CIDR block of the DLI queue. Click **OK**.
6. Log in to the DLI management console and choose **Resources** > **Queue Management**. In the row of the Flink queue, choose **More** > **Test Address Connectivity**. In the **Address** text box, enter the Kafka connection address and port number in the format of IP address:port number, and click **Test**. The subsequent operations can be performed only when the address is reachable. Note that multiple addresses must be tested separately.

Step 9: Creating a Flink SQL Job

After the data source and data output channel are prepared, you can create a Flink SQL job.

1. In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.
2. In the upper right corner of the **Flink Jobs** page, click **Create Job**. Set the following parameters:
   - **Type**: **Flink SQL**
   - **Name**: **DIS-Flink-Kafka**
   - **Description**: Leave it blank.
   - **Template Name**: Do not select any template.
3. Click **OK** to enter the editing page.
4. Edit the Flink SQL job

   Enter SQL statements in the editing window. The example statements are as follows. Note that the values of the parameters in bold must be changed according to the comments.

```
CREATE SOURCE STREAM car_info (
  a1 string,
  a2 string,
  a3 string,
  a4 INT
)
WITH (
  type = "dis",
  region = "xxx",// Region where the current DLI queue is located
  channel = "csinput",
  encode = "csv",
  FIELD_DELIMITER = ";"
);

 CREATE SINK STREAM kafka_sink (
  a1 string,
  a2 string,
  a3 string,
  a4 INT
) // Output field

WITH (
  type="kafka",
Change kafka_bootstrap_servers = "192.x.x.x:9093, 192.x.x.x:9093, 192.x.x.x:9093",// Connection address of the Kafka instance
kafka_topic = "testflink", // Topic to be written to Kafka. Log in to the Kafka console, click the name of the created Kafka instance, and view the topic name on the Topic Management page.
  encode = "csv", // Encoding format, which can be JSON or CSV.
  kafka_certificate_name = "Flink",// The value is the name of the Kafka datasource
authentication created in Step 7.
  kafka_properties_delimiter = ",",
  // Replace xxx in username and password in kafka_properties with the username and password
for creating SSL authentication for Kafka in step 2.
  kafka_properties = "sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule
required username=\"xxx\" password=\"xxx\";,sasl.mechanism=PLAIN,security.protocol=SASL_SSL"
);

INSERT INTO kafka_sink
SELECT * FROM car_info;

CREATE sink STREAM car_info1 (
  a1 string,
  a2 string,
  a3 string,
  a4 INT
)
WITH (
  type = "dis",
  region = "xxx",// Region where the current DLI queue is located
  channel = "csinput",
  encode = "csv",
  FIELD_DELIMITER = ";"
);

insert into car_info1 select 'id','owner','brand',1;
insert into car_info1 select 'id','owner','brand',2;
insert into car_info1 select 'id','owner','brand',3;
insert into car_info1 select 'id','owner','brand',4;
insert into car_info1 select 'id','owner','brand',5;
insert into car_info1 select 'id','owner','brand',6;
insert into car_info1 select 'id','owner','brand',7;
insert into car_info1 select 'id','owner','brand',8;
insert into car_info1 select 'id','owner','brand',9;
insert into car_info1 select 'id','owner','brand',10;
```

5. Click **Check Semantics**.
6. Set job running parameters. The mandatory parameters are as follows:
   - **Queue**: Flinktest
   - **CUs**: 2
   - **Job Manager CUs**: 1
   - **Parallelism**: 1
   - **Save Job Log**: selected
   - **OBS Bucket**: Select the OBS bucket for storing job logs. You need the permissions to access this bucket.

   You do not need to set other parameters.
7. Click **Save**.
8. Click **Start**. On the displayed **Start Flink Job** page, confirm the job specifications and the price, and click **Start Now** to start the job.

   After the job is started, the system automatically switches to the **Flink Jobs** page, and the created job is displayed in the job list. You can view the job status in the **Status** column. After a job is successfully submitted, **Status** of the job will change from **Submitting** to **Running**.

   If **Status** of a job is **Submission failed** or **Running exception**, the job fails to be submitted or fails to run. In this case, you can move the cursor over the status icon to view the error details. You can click ❐ to copy these details. After handling the fault based on the provided information, resubmit the job.

9. After the job is complete, you can log in to the management console of Distributed Message Service for Kafka to view the corresponding Kafka premium instance. Click the instance name, click the **Message Query** tab, select the Kafka topic written in the Flink SQL job, and click **Search**. In the **Operation** column, click **View Message Body** to view the written message content.

3 DLI Console Overview

Overview

Log in to DLI and go to the **Overview** page.

The following table describes the functional areas of the **Overview** page.

Table 3-1 Areas

No.	Area	Description
1	Navigation pane	The navigation bar on the DLI management console, including **Overview**, **SQL Editor**, **Job Management**, **Queue Management**, **Data Management**, **Job Templates**, **Datasource Connections**, and **Global Configuration**.
2	DLI introduction	Data Lake Insight (DLI) is a fully hosted big data processing and analysis service. It frees you from managing any server. DLI supports standard SQL and is compatible with Spark and Flink SQL. It also supports multiple access modes, and is compatible with mainstream data formats.
3	Upper right corner	Click **Create Queue** to create a queue.
4	SQL job introduction	Click **Create Job** in this area to access the **SQL Editor** page. For details about how to create a package, see **SQL Editor**.
5	Flink job introduction	Click **Create Job** in this area to go to the **Flink Jobs** page. For details about how to create a package, see **Flink Job Management**.
6	Spark job introduction	Click **Create Job** in this area to access the **Spark Jobs** page. For details about how to create a Spark job, see **Creating a Spark Job**.

No.	Area	Description
8	Queue usage (hours)	This area displays the usage of all queues of the last day, week, or month.
9	Selected queue usage (hours)	This area displays the usage of a queue in the last week or month.
10	Common links	Including: ● *User Guide* ● *API Reference* ● *SDK Download*
11	Get started in four simple steps	1. Create a queue 2. Preparing data 3. Editing and submitting a job 4. Viewing job status

4 SQL Editor

You can use SQL statements in the SQL job editor to execute data query. DLI supports SQL 2003 and complies with Spark SQL.

On the **Overview** page, click **SQL Editor** in the navigation pane on the left or **Create Job** in the upper right corner of the **SQL Jobs** pane. The **SQL Editor** page is displayed.

A message is displayed, indicating that a temporary DLI data bucket will be created. The created bucket is used to store temporary data generated by DLI, such as job logs. You cannot view job logs if you choose not to create it. You can to periodically delete objects in a bucket or transit objects between different storage classes. The bucket name is set by default.

If you do not need to create a DLI temporary data bucket and do not want to view this message, select **Do not show again** and click **Cancel**.

The **SQL Editor** page is composed of the following parts:

Navigation pane

The navigation pane on the left consists of **Databases**, **Queues**, and **Templates** tabs.

Table 4-1 Navigation pane buttons

No.	Tab/Button	Name	Description
1	Databases	Databases	Displays all the existing databases and tables in these databases. ● Click a database name to view the tables in the database. ● Click a table name to view the metadata in the table. A maximum of 20 metadata records can be displayed. ● After you double-click a table name, a SQL query statement is automatically entered in the editing window.
2	Queues	Queues	Displays existing queues.
3	Templates	Templates	Click the drop-down button to view 22 built-in standard TPC-H query templates and custom templates.
4	⊕	Create	Click the button on the related page to create a queue, database, or table. For details, see **Creating a Queue** and **Creating a Database or a Table**.
5	↻	Refresh	Click the button to refresh the existing queue, database, and table lists.
6	🔍	Search	Enter a keyword to search for a database and table.

SQL Editing Window

The SQL job editing window is displayed in the upper right corner. The following describes the components of the window.

Table 4-2 Components of the SQL job editing window

No.	Button	Description
2	Queues	Select a queue from the drop-down list box. If no queue is available, the **default** queue is displayed. Refer to **Creating a Queue** and create a queue. SQL jobs can be executed only on SQL queues.

No.	Button	Description
3	Database	Select a database from the drop-down list box. If no database is available, the **default** database is displayed. For details about how to create a database, see section **Creating a Database or a Table**. **NOTE** If you specify the database in the SQL statements, the database you choose from the drop-down list will not be used.
4	Execute	Click this button to run the SQL statements in the job editing window.
5	Format	Click this button to format the SQL statements.
6	Syntax Reference	Click this button to view the *Data Lake Insight SQL Syntax Reference*.
7	Settings	Add parameters and tags. **Parameter Settings**: Set parameters in key/value format for SQL jobs. **Tags**: Set tags in key/value format for SQL jobs.
8	More	The drop-down list includes the following options: ● Click **Verify Syntax** to check whether the SQL statements are correct. ● Click **Set as Template** to set SQL statements as a template. For details, see **SQL Template Management**. ● Click **Change Theme** to switch between dark and light modes.
9	SQL statement editing area	Area for you to write and edit SQL statements.
10	Cursor position	The line and column where the cursor is in the job editing area.
11	Shortcuts	For details, see **Table 4-3**.

Table 4-3 Keyboard shortcut description

Shortcut	Description
Ctrl+Enter	Execute SQL statements. You can run SQL statements by pressing **Ctrl+R** or **Ctrl + Enter** on the keyboard.
Ctrl+F	Search for SQL statements. You can press Ctrl+F to search for a required SQL statement.

Shortcut	Description
Shift+Alt+F	Format SQL statements. You can press **Shift + Alt + F** to format a SQL statement.
Ctrl+Q	Syntax verification. You can press **Ctrl + Q** to verify the syntax of SQL statements.
F11	Full screen. You can press **F11** to display the SQL Job Editor window in full screen. Press **F11** again to leave the full screen.

Execution Queries and View Result

The **Executed Queries (Last Day)** and **View Result** tab pages are displayed in the lower part of the page.

- Executed Queries

Table 4-4 Area description

Area	Description
Executed Queries (Last Day)	Displays the latest daily information about the submitted jobs. ● **Queues**: Queue name ● **Username**: User who executes the SQL statements ● **Type**: Type of the SQL job ● **Status**: Execution status of the SQL job ● **Query** ● **Created** ● **Operation** – **Edit**: Edit the SQL statement. – **SparkUI**: Switch to the SparkUI page to view the SQL statement execution process. **NOTE** When you execute a job on a created queue, the cluster is restarted. It takes about 10 minutes. If you click **SparkUI** before the cluster is created, an empty **projectID** will be cached. The SparkUI page cannot be displayed. You are advised to use a dedicated queue so that the cluster will not be released. Alternatively, wait for a while after the job is submitted (the cluster is created), and then check **SparkUI**. Currently, only the latest 100 job information records are displayed on the SparkUI of DLI. This function is not supported for synchronization jobs and jobs running on the default queue. – **More**: The following operations vary depending on the SQL job types and running status. **Cancel**: Cancel a SQL job that is running or being submitted. **Re-execute**: Execute the SQL statement again. **View Result**: View the execution result of a QUERY job. **Export Result**: Export the execution results of a QUERY job to a specified OBS path. **View Log**: View the OBS path for storing SQL statement execution logs. **Export Log**: Export SQL statement execution logs. **NOTE** To export the logs, you need to obtain the permission to create an OBS bucket. **View Log** and **Export Log** buttons are not available for synchronization jobs and jobs running on the default queue.

You can filter the execution history in the following ways:

  - In the search box in the upper right corner of the **Executed Queries (Last Day)** pane, select a queue name or enter an execution statement in the search box.
  - In the list, click the icon next to **Created** and choose **Ascending** or **Descending**.
  - Select a job status from the **Status** list.

- View Result

Table 4-5 Function description

No.	Button	Description
1	Clear All	Clear the displayed SQL statement query results.
2	📊	Display the query result in a graph or table.
3	↗	Exports the query result to OBS. For details, see **Exporting Query Results**. A maximum of 1000 records can be displayed in the query result on the console. To view more or all data, you can click **Export Result** to export the data to OBS.

SQL Query Procedure

1. Log in to the Data Lake Insight management console, choose **Job Management** > **SQL Jobs**. In the displayed page, click **Create Job** to go to the **SQL Editor** page.
2. After you enter the **SQL Editor** page, the system prompts you to create a temporary OBS bucket to store temporary data generated by DLI, such as job logs. You cannot view job logs if you choose not to create it. Click **OK**. The system displays a message indicating that the temporary bucket is created successfully.
3. Select a queue from the queue list in the upper left corner of the SQL job editing window. For details about how to create a queue, see **Creating a Queue**.
4. In the upper right corner of the SQL job editing window, select a database, for example, **qw**, from the **Databases** drop-down list.
5. Create a table, for example, **qw**. For details about how to create a database and table, see **Creating a Database or a Table**.
6. In the SQL job editing window, enter the following SQL statement:

   ```
   SELECT * FROM qw.qw LIMIT 10;
   ```

   Alternatively, you can double-click the table name **qw**. The query statement is automatically entered in the SQL job editing window.
7. On top of the editing window, click **More** > **Verify Syntax** to check whether the SQL statement is correct.

a. If the verification fails, check the SQL statement syntax by referring to *Data Lake Insight SQL Syntax Reference*.

b. If the syntax verification is successful, click **Execute**. Read and agree to the privacy agreement. Click **OK** to execute the SQL statement.

c. After the execution is complete, you can view the execution result in the area under the SQL job editing window.

8. (Optional) A maximum of 1000 records can be displayed in the query result on the current console. To view more or all data, click ⧉ to export the data to OBS.

9. (Optional) In the **View Result** tab page, click 📊 in the upper right corner to display the result in a graph. Click 📊 to display the query result in a table.

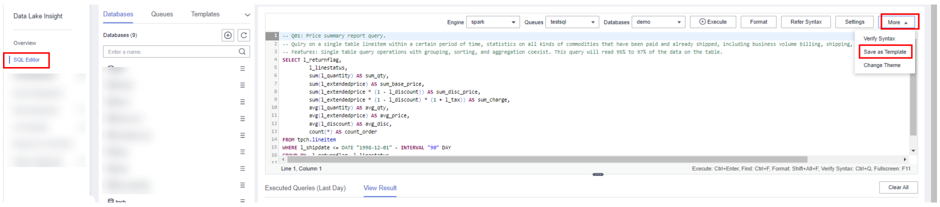
NOTE

- If no column of the numeric type is displayed in the execution result, the result cannot be represented in charts.
- You can view the data in a bar chart, line chart, or fan chart.
- In the bar chart and line chart, the X axis can be any column, while the Y axis can only be columns of the numeric type. The fan chart displays the corresponding legends and indicators.

Operation Skills in the Job Editing Window

- A simple method for importing SQL statements in the SQL job editing window:
  - Double-click a table name in the navigation pane on the left to import the query statement of the selected table into the SQL statement editing window, and then click **Execute** to query.
  - You can click **More** and choose **Save as Template** to save the SQL statement as a template for future use.

    To use the SQL statement template, click **Templates** from the left pane of the SQL editor page. Double-click the required template in the template list, and click **Execute** or modify it as required before executing the SQL statements.



Engine spark
Queues testsql
Databases demo

```
-- Q01: Price summary report query.
-- Quiry on a single table lineitem within a certain period of time, statistics on all kinds of commodities that have been paid and already shipped, including business volume billing, shipping,
-- Features: Single table query operations with grouping, sorting, and aggregation coexist. This query will read 95% to 97% of the data on the table.
SELECT l_returnflag,
       l_linestatus,
       sum(l_quantity) AS sum_qty,
       sum(l_extendedprice) AS sum_base_price,
       sum(l_extendedprice * (1 - l_discount)) AS sum_disc_price,
       sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)) AS sum_charge,
       avg(l_quantity) AS avg_qty,
       avg(l_extendedprice) AS avg_price,
       avg(l_discount) AS avg_disc,
       count(*) AS count_order
FROM tpch.lineitem
WHERE l_shipdate <= DATE "1998-12-01" - INTERVAL "90" DAY
```

Line 1, Column 1

Execute: Ctrl+Enter, Find: Ctrl+F, Format: Shift+Alt+F, Verify Syntax: Ctrl+Q, Fullscreen: F11

  - SQL statements can be executed in batches.
- Features of the job editing window

  The job editing window has the following three features:
  - Highlighting: Statement keywords are highlighted.
  - Comment support: You can write comments in a single line or multiple lines. In each line, the comments follow two consecutive hyphens (--).
  - Cursor location display: After you click in a line in the job editing window, the location of your cursor in the job editing window, including the

column number and line number, will be automatically displayed in the lower left corner of the window.

5 Job Management

5.1 SQL Job Management

SQL jobs allow you to execute SQL statements entered in the **SQL job editing window**, import data, and export data.

SQL job management provides the following functions:

- **Searching for Jobs**: Search for jobs that meet the search criteria.
- **Viewing Job Details**: Display job details.
- **Terminating a Job**: Stop a job in the **Submitting** or **Running** status.
- **Exporting Query Results**: A maximum of 1000 records can be displayed in the query result on the console. To view more or all data, you can export the data to OBS.

SQL Jobs page

On the **Overview** page of the DLI console, click **SQL Jobs** to go to the SQL job management page. Alternatively, you can click **Job Management** > **SQL Jobs**. The job list displays all SQL jobs. If there are a large number of jobs, they will be displayed on multiple pages. You can switch to the specified page as needed. DLI allows you to view jobs in all statuses. By default, jobs in the job list are displayed in descending order of the job creation time.

Table 5-1 SQL Job management parameters

Parameter	Description
Queues	Name of the queue to which a job belongs
Username	Name of the user who executed the job.

Parameter	Description
Type	Job type. The following types are supported: ● **IMPORT**: A job that imports data to DLI ● **EXPORT**: A job that exports data from DLI ● **DCL**: Conventional DCLs and operations related to queue permissions ● **DDL**:Conventional DDLs, including creating and deleting databases and tables ● **QUERY**: A job that querys data by running SQL statements ● **INSERT**: A job that inserts data by running SQL statements ● **UPDATE**: A job that updates data. ● **DELETE**: A job that deletes a SQL job. ● **DATA_MIGRATION**: A job that migrates data. ● **RESTART_QUEUE**: A job that restarts a queue. ● **SCALE_QUEUE**: A job that changes queue specifications, including sale-out and scale-in.
Status	Job status. Possible values are as follows: ● **Submitting** ● **Running** ● **Finished** ● **Canceled** ● **Failed** ● **Scaling**
Query	SQL statements for operations such as exporting and creating tables You can click ⧉ to copy the query statement.
Duration	Running duration of a job
Created	Time when a job is created. Jobs can be displayed in ascending or descending order of the job creation time.

Parameter	Description
Operation	- **Edit**: Edit the job. - **Cancel** – You can terminate a job only when the job is in **Submitting** or **Running** status. – A job whose status is **Finished**, **Failed**, or **Canceled** cannot be terminated. – If the **Cancel** button is gray, you are not allowed to perform this operation. - **Re-execute**: Execute the job again. - **SparkUI**: Display the Spark job execution page. NOTE - When you execute a job on a created queue, the cluster is restarted. It takes about 10 minutes. If you click **SparkUI** before the cluster is created, an empty **projectID** will be cached. The SparkUI page cannot be displayed. You are advised to use a dedicated queue so that the cluster will not be released. Alternatively, wait for a while after the job is submitted (the cluster is created), and then check **SparkUI**. - Currently, only the latest 100 job information records are displayed on the SparkUI of DLI. - In addition to the preceding operations, the following operations are available for QUERY jobs and asynchronous DDL jobs. – **View Result**: View the job running result. – **Export Result**: Export the job running result to the created OBS bucket. For details, see section **Exporting Query Results**. - In addition to the preceding operations, the EXPORT job also includes the following operations: – **Download** - **View Log**: Save job logs to the temporary OBS bucket created by DLI. - **Export Log**: Export logs to the created OBS bucket. If the job is in the **Running** state, logs cannot be exported. NOTE To export the logs, you need to obtain the permission to create an OBS bucket. Log archiving and export are not available for synchronization jobs and jobs running on the default queue.

Searching for a Job

On the **SQL Jobs** page, you can search jobs with any of the following operations.

- Select a queue name.

- Set the date range.
- Enter a username, statement, or job ID.
- Select the creation time in ascending or descending order.
- Select a job type.
- Select a job status.
- Select the job execution duration in ascending or descending order.

Viewing Job Details

On the **SQL Jobs** page, you can click ⌄ in front of a job record to view details about the job.

The following describes how to load data, create a table, and select a job. The job details vary depending on the job types, status, and configuration options.

- **Load data** (job type: IMPORT) include the following information: queue, job ID, username, type, status, execution statement, running duration, creation time, end time, parameter settings, label, number of results, scanned data, number of scanned data, number of error records, storage path, data format, database, table, table header, separator, reference character, escape character, date format, and timestamp format.
- **Create table** (job type: DDL) include the following information: queue, job ID, username, type, status, execution statement, running duration, creation time, end time, parameter settings, tags, number of results, scanned data, and database.
- **Select** (job type: QUERY) include the following information: queue, job ID, username, type, status, execution statement, running duration, creation time, end time, parameter setting, label, number of results (results of successful executions can be exported), and scanned data, username, result status (results of successful tasks can be viewed. Failure causes of failed tasks are displayed), and database.

Terminating a Job

On the **SQL Jobs** page, you can click **Terminate** in the **Operation** column to stop a submitting or running job.

Exporting Query Results

A maximum of 1000 records can be displayed in the query result on the console. To view more or all data, you can export the data to OBS. The procedure is as follows:

You can export results on the **SQL Jobs** page or the **SQL Editor** page.

- On the **Job Management** > **SQL Jobs** page, you can click **More** > **Export Result** in the **Operation** column to export the query result.
- After the query statements are successfully executed on the **SQL Editor** page, click ↗ on the right of the **View Result** tab page to export the query result.

NOTE

If no column of the numeric type is displayed in the query result, the result cannot be exported.

Table 5-2 Exporting parameters

Parameter	Description
Data Format	Format of the exported query result file. Formats other than JSON will be supported in later versions.
Queues	The queue where the jobs are excuted. SQL jobs can be executed only in SQL queues. For details about how to create a queue, see **Creating a Queue**.
Compression Format	Compression format of the data to be exported. The following options are supported: ● none ● bzip2 ● deflate ● gzip
Storage Path	OBS path to store the result. **NOTE** ● After selecting an OBS bucket, enter a name for the folder. If the folder does not exist, it will be created in OBS. ● The folder name cannot contain the special characters of \ / : * ? "< > |, and cannot start or end with a dot (.).
Export Mode	Mode for saving the exported query result data. ● **New OBS directory**: If the specified export directory exists, an error is reported and the export operation cannot be performed. ● **Existing OBS directory (Overwritten)**: If you create a file in the specified directory, the existing file will be overwritten.
Number of Results	Number of exported query results. If no value is entered or the value is **0**, all results are exported.
Table Header	Whether the data to be exported contains table headers.

5.2 Flink Job Management

5.2.1 Overview

On the Job Management page of Flink jobs, you can submit a Flink job. Currently, the following job types are supported:

- **Flink SQL** uses SQL statements to define jobs and can be submitted to any general purpose queue.
- **Flink Jar** customizes a JAR package job based on Flink APIs. It runs on dedicated queues.

Flink job management provides the following functions:

- **Managing Flink Job Permissions**
- **Creating a Flink SQL Job**
- **Creating a Flink Jar Job**
- **Debugging a Job**
- **Editing a job**
- **Starting a Job**
- **Stopping a Job**
- **Deleting a Job**
- **Exporting a Job**
- **Importing a Job**
- **Modifying Name and Description**
- **Importing to a Savepoint**
- **Triggering a Savepoint**
- **Runtime Configuration**
- **Job Details**

Assigning Agency Permissions

Agencies are required for DLI to execute Flink jobs. You can set the agency when logging in to the management console for the first time or go to **Global Configurations** > **Service Authorization** to modify the agencies.

The permissions are as follows:

- **Tenant Administrator (global)** permissions are required to access data from OBS to execute Flink jobs on DLI, for example, obtaining OBS/DWS data sources, log dump (including bucket authorization), checkpointing enabling, and job import and export.

  NOTE

  Due to cloud service cache differences, permission setting operations require about 60 minutes to take effect.
- **DIS Administrator** permissions are required to use DIS data as the data source of DLI Flink jobs.

  NOTE

  Due to cloud service cache differences, permission setting operations require about 30 minutes to take effect.
- To use CloudTable data as the data source of DLI Flink jobs, **CloudTable Administrator** permissions are required.

NOTE

Due to cloud service cache differences, permission setting operations require about 3 minutes to take effect.

- **Tenant Administrator (Project-level)** permissions are required if you use services that can run only with these permissions.

NOTE

Due to cloud service cache differences, permission setting operations require about 3 minutes to take effect.

Flink Jobs Page

On the **Overview** page, click **Flink Jobs** to go to the Flink job management page. Alternatively, you can choose **Job Management** > **Flink Jobs** from the navigation pane on the left. The page displays all Flink jobs. If there are a large number of jobs, they will be displayed on multiple pages. DLI allows you to view jobs in all statuses.

Table 5-3 Job management parameters

Parameter	Description
ID	ID of a submitted Flink job, which is generated by the system by default.
Name	Name of the submitted Flink job.
Type	Type of the submitted Flink job. Including: ● Flink SQL: Flink SQL jobs ● Flink Jar: Flink Jar jobs
Status	Job statuses, including: ● **Draft** ● **Submitting** ● **Submission failed** ● **Running**: After the job is submitted, a normal result is returned. ● **Running exception**: The job stops running due to an exception. ● **Downloading** ● **Idle** ● **Stopping** ● **Stopped** ● **Stopping failed** ● **Creating the savepoint** ● **Completed**
Description	Description of the submitted Flink job.
Username	Name of the user who submits a job.

Parameter	Description
Created	Time when a job is created.
Started	Time when a Flink job starts to run.
Duration	Time consumed by job running.
Operation	● **Edit**: Edit a created job. For details, see **Editing a Job**. ● **Start**: Start and run a job. For details, see **Starting a Job**. ● **More** – **FlinkUI**: After you click this button, the Flink job execution page is displayed. NOTE When you execute a job on a created queue, the cluster is restarted. It takes about 10 minutes. If you click **FlinkUI** before the cluster is created, an empty **projectID** will be cached. The **FlinkUI** page cannot be displayed. You are advised to use a dedicated queue so that the cluster will not be released. Alternatively, wait for a while after the job is submitted (the cluster is created), and then check **FlinkUI**. – **Stop**: Stop a Flink job. If this function is unavailable, jobs in the current status cannot be stopped. – **Delete**: Delete a job. NOTE A deleted job cannot be restored. – **Modify Name and Description**: You can modify the name and description of a job. For details, see **Modifying Name and Description**. – **Import Savepoint**: Import the data exported from the original CS job. For details, see **Importing to a Savepoint**. – **Trigger Savepoint**: You can click this button for jobs in the **Running** status to save the job status. For details, see **Triggering a Savepoint**. – **Permissions**: You can view the user permissions corresponding to the job and grant permissions to other users. For details, see **Managing Flink Job Permissions**. – **Runtime Configuration**: You can enable **Alarm Generation upon Job Exception** and **Auto Restart upon Exception**. For details, see **Runtime Configuration**.

5.2.2 Managing Flink Job Permissions

Scenarios

- You can isolate Flink jobs allocated to different users by setting permissions to ensure data query performance.
- The administrator and job creator have all permissions, which cannot be set or modified by other users.

Flink Job Permission Operations

1. On the left of the DLI management console, choose **Job Management** > **Flink Jobs**.
2. Select the job to be configured and choose **More** > **Permissions** in the **Operation** column. The **User Permissions** area displays the list of users who have permissions on the job.

   You can assign queue permissions to new users, modify permissions for users who have some permissions of a queue, and revoke all permissions of a user on a queue.

   – Assign permissions to a new user.

     A new user does not have permissions on the job.

     i. Click **Grant Permission** on the right of **User Permissions** page. The **Grant Permission** dialog box is displayed.
     ii. Specify **Username** and select corresponding permissions.
     iii. Click **OK**.

     Table 5-4 describes the related parameters.

Table 5-4 Permission parameters

Parameter	Description
Username	Name of the user you want to grant permissions to. **NOTE** The username is the name of an existing IAM user. In addition, the user can perform authorization operations only after logging in to the platform.
Permissions to be granted to the user	• **View Job Details**: This permission allows you to view the job details. • **Modify Job**: This permission allows you to modify the job. • **Delete Job**: This permission allows you to delete the job. • **Start Job**: This permission allows you to start the job. • **Stop Job**: This permission allows you to stop the job. • **Export Job**: This permission allows you to export the job. • **Grant Permission**: This permission allows you to grant job permissions to other users. • **Revoke Permission**: This permission allows you to revoke the job permissions that other users have but cannot revoke the job creator's permissions. • **View Other User's Permissions**: This permission allows you to view the job permissions of other users.

  - To assign or revoke permissions of a user who has some permissions on the job, perform the following steps:

    i. In the list under **User Permissions** for a job, select the user whose permissions need to be modified and click **Set Permission** in the **Operation** column.

    ii. In the displayed **Set Permission** dialog box, modify the permissions of the current user. **Table 5-4** lists the detailed permission descriptions.

       If all options under **Set Permission** are gray, you are not allowed to change permissions on this job. You can apply to the administrator, job creator, or other authorized users for job permission granting and revoking.

    iii. Click **OK**.

  - To revoke all permissions of a user on a job, perform the following steps:

    In the list under **User Permissions** for a job, locate the user whose permissions need to be revoked, click **Revoke Permission** in the **Operation** column, and click **Yes**. After this operation, the user does not have any permission on the job.

Flink Job Permissions

- **View Job Details**
  - Tenants and the admin user can view and operate all jobs.
  - Subusers and users with the read-only permission can only view their own jobs.

    NOTE

    If another user grants any permission other than the job viewing permission to a subuser, the job is displayed in the job list, but the details cannot be viewed by the subuser.
- **Start Job**
  - To use an exclusive queue, you must have the permission to submit and start jobs.
  - To use a shared queue, you only need to have the permission to start jobs.
- **Stop Job**
  - To use an exclusive queue, you must have the permission to stop jobs and queues.
  - To use a shared queue, you only need to have the permission to stop jobs.
- **Delete Job**
  - If a job can be deleted, you can delete the job if you were granted this permission.
  - If a job cannot be deleted, the system stops the job before you delete it. For details about how to stop a job, see **Stop Job**. In addition, you must have the permission to delete the job.

- **Create Job**
  - By default, sub-users cannot create jobs.
  - To create a job, you must have this permission. Currently, only the admin user has the permission to create jobs. In addition, the user must have the permission of the related package group or package used by the job.
- **Modify Job**

  When modifying a job, you need to have the permission to update the job and the permission to the package group or package used by the job belongs.

5.2.3 Preparing Flink Job Data

To create a Flink job, you need to enter the data source and data output channel, that is, source and sink. To use another service as the source or sink stream, you need to apply for the service first.

Flink jobs support the following data sources and output channels:

- DIS as the data input and output channel

  To use DIS as the data source and output channel, you need to enable DIS first.

  For details about how to create a DIS stream, see **Creating a DIS Stream** in the *Data Ingestion Service User Guide*.

  After applying for a DIS stream, you can upload local data to DIS to provide data sources for Flink jobs in real time. For details, see **Sending Data to DIS** in the *Data Ingestion Service User Guide*.

  An example is provided as follows:

  ```
  1,lilei,bmw320i,28
  2,hanmeimei,audia4,27
  ```

- OBS as the data source

  To use OBS as the data source, enable OBS first. For details about how to enable OBS, see **Enabling OBS** in the *Object Storage Service Console Operation Guide*.

  After you enable OBS, upload local files to OBS using the Internet. For detailed operations, see **Uploading a File** in the *Object Storage Service Console Operation Guide*.

- RDS as the output channel

  To use RDS as the output channel, create an RDS instance. For details, see **Creating a DB Instance** in the *Relational Database Service User Guide*.

- SMN as the output channel

  To use SMN as the output channel, create an SMN topic to obtain the URN resource ID and then add topic subscription. For detailed operations, see **Getting Started** in the *Simple Message Notification User Guide*.

- Kafka as the data input and output channel

  If Kafka serves as both the source and sink streams, create an enhanced datasource connection between Flink jobs and Kafka. For details, see **Enhanced Datasource Connections**.

  If the port of the Kafka server is listened on by the host name, you need to add the mapping between the host name and IP address of the Kafka Broker node to the datasource connection.

- CloudTable as the data input and output channel

  To use CloudTable as the data input and output channel, create a cluster in CloudTable and obtain the cluster ID.

- CSS as the output channel

  To use CSS as the data output channel, create a cluster in CSS and obtain the cluster's private network address. For details, see **Getting Started** in the *Cloud Search Service User Guide*.

- DCS as the output channel

  To use DCS as the output channel, create a Redis cache instance in DCS and obtain the address used for Flink jobs to connect to the Redis instance. For detailed operations, see **Getting Started** in the *Distributed Cache Service User Guide*.

5.2.4 Creating a Flink SQL job

This section describes how to create a Flink SQL job. You can use Flink SQLs to develop jobs to meet your service requirements. Using SQL statements simplifies logic implementation. You can edit Flink SQL statements for your job in the DLI SQL editor. This section describes how to use the SQL editor to write Flink SQL statements.

Prerequisites

- You have prepared the data input and data output channels. For details, see **Preparing Flink Job Data**.
- When you use a Flink SQL job to access other external data sources, such as OpenTSDB, HBase, Kafka, DWS, RDS, CSS, CloudTable, DCS Redis, and DDS MongoDB, you need to create a cross-source connection to connect the job running queue to the external data source.
  - For details about the external data sources that can be accessed by Flink jobs, see **Datasource Connection and Cross-Source Analysis**.
  - For details about how to create a datasource connection, see **Enhanced Datasource Connections**. After a datasource connection is created, you can choose **More** > **Test Address Connectivity** in the **Operation** column on the **Queue Management** page to check whether the network connection between the queue and the external data source is normal. For details, see **Testing Address Connectivity**.

Creating a Flink SQL Job

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 In the upper right corner of the **Flink Jobs** page, click **Create Job**.

Step 3 Specify job parameters.

Table 5-5 Job configuration information

Parameter	Description
Type	Set **Type** to **Flink SQL**. You will need to rewrite SQL statements to start the job.
Name	Name of a job. Enter 1 to 57 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. **NOTE** The job name must be globally unique.
Description	Description of a job. It can contain up to 512 characters.
Template Name	You can select a sample template or a custom job template. For details about templates, see **Flink Template Management**.

Step 4 Click **OK** to enter the **Edit** page.

Step 5 Edit a Flink SQL job.

Enter details SQL statements in the statement editing area. For details about SQL syntax, see the *Data Lake Insight SQL Syntax Reference*.

Step 6 Click **Check Semantics**.

- You can **Debug** or **Start** a job only after the semantic verification is successful.
- If verification is successful, the message "The SQL semantic verification is complete. No error." will be displayed.
- If verification fails, a red "X" mark will be displayed in front of each SQL statement that produced an error. You can move the cursor to the "X" mark to view error details and change the SQL statement as prompted.

Step 7 Set job running parameters.

Table 5-6 Running parameters

Parameter	Description
Queue	A shared queue is selected by default. You can select a custom queue as needed. **NOTE** ● During job creation, a sub-user can only select a queue that has been allocated to the user. ● If the remaining capacity of the selected queue cannot meet the job requirements, the system automatically scales up the capacity. When a queue is idle, the system automatically scales in the queue.

Parameter	Description
UDF Jar	If you selected custom queues, you need to configure this parameter. You can customize a UDF Jar file. Before you select a JAR file, upload the corresponding JAR package to the OBS bucket and choose **Data Management > Package Management** to create a package. For details, see **Creating a Package**. In SQL, you can call a user-defined function that is inserted into a JAR file.
CUs	Sum of the number of compute units and job manager CUs of DLI. One CU equals one vCPU and 4 GB. The configured number of CUs is the number of CUs required for job running and cannot exceed the number of CUs in the bound queue.
Job Manager CUs	Number of CUs of the management unit.
Parallelism	Number of Flink SQL jobs that run at the same time Properly increasing the number of parallel threads improves the overall computing capability of the job. However, the switchover overhead caused by the increase of threads must be considered. **NOTE** ● This value cannot be greater than four times the compute units (number of CUs minus the number of job manager CUs). ● The priority of the number of parallel tasks on this page is lower than that set in the code.
Task Manager Configuration	Whether to set Task Manager resource parameters If this option is selected, you need to set the following parameters: ● **CU(s) per TM**: Number of resources occupied by each Task Manager. ● **Slot(s) per TM**: Number of slots contained in each Task Manager.
OBS Bucket	OBS bucket to store job logs and checkpoint information. If the selected OBS bucket is not authorized, click **Authorize**. **NOTE** If both **Enable Checkpointing** and **Save Job Log** are selected, you only need to authorize OBS once.

Parameter	Description
Save Job Log	Whether to save the job running logs to OBS The logs are saved in the following path: *Bucket name*/**jobs/logs/***Directory starting with the job ID*. To go to this path, go to the job list and click the job name. On the **Run Log** tab page, click the provided OBS link. **CAUTION** You are advised to select this parameter. Otherwise, no run log is generated after the job is executed. If the job is abnormal, the run log cannot be obtained for fault locating. If this option is selected, you need to set the following parameters: **OBS Bucket**: Select an OBS bucket to store user job logs. If the selected OBS bucket is not authorized, click **Authorize**. **NOTE** If both **Enable Checkpointing** and **Save Job Log** are selected, you only need to authorize OBS once.
Alarm Generation upon Job Exception	Whether to report job exceptions, for example, abnormal job running or exceptions due to an insufficient balance, to users via SMS or email If this option is selected, you need to set the following parameters: **SMN Topic** Select a user-defined SMN topic. For details about how to customize SMN topics, see **Creating a Topic** in the *Simple Message Notification User Guide*.
Enable Checkpointing	Whether to enable job snapshots. If this function is enabled, jobs can be restored based on the checkpoints. If this option is selected, you need to set the following parameters: ● **Checkpoint Interval** indicates the interval for creating checkpoints. The value ranges from 1 to 999999, and the default value is **30**. ● **Checkpoint Mode** can be set to either of the following values: – **At least once**: Events are processed at least once. – **Exactly once**: Events are processed only once. ● **OBS Bucket**: Select an OBS bucket to store your checkpoints. If the selected OBS bucket is not authorized, click **Authorize**. The checkpoint path is *Bucket name*/**jobs/checkpoint/** *Directory starting with the job ID*. **NOTE** You only need to authorize OBS once for both **Enable Checkpointing** and **Save Job Log**.

Parameter	Description
Auto Restart upon Exception	Whether to enable automatic restart. If this function is enabled, any job that has become abnormal will be automatically restarted. If this option is selected, you need to set the following parameters: ● **Max. Retry Attempts**: maximum number of retry times upon an exception. The unit is **Times/hour**. – **Unlimited**: The number of retries is unlimited. – **Limited**: The number of retries is user-defined. ● **Restore Job from Checkpoint**: This parameter is available only when **Enable Checkpointing** is selected.
Idle State Retention Time	Defines for how long the state of a key is retained without being updated before it is removed in **GroupBy** or **Window**. The default value is 1 hour.
Dirty Data Policy	Select a policy for processing dirty data. The following policies are supported: **Ignore**, **Trigger a job exception**, and **Save**. **NOTE** **Save** indicates that the dirty data is stored to the OBS bucket selected above.
Dirty Data Dump Address	Set this parameter when **Dirty Data Policy** is set to **Save**. Click the address box to select the OBS path for storing dirty data.

Step 8 (Optional) Debug parameters as required. The job debugging function is used only to verify the SQL logic and does not involve data write operations. For details, see **Debugging a Flink Job**.

Step 9 (Optional) Set the runtime configuration as required. Set Custom Configuration to User-defined.

Step 10 Click **Save**.

Step 11 Click **Start**. On the displayed **Start Flink Jobs** page, confirm the job specifications, and click **Start Now** to start the job.

After the job is started, the system automatically switches to the **Flink Jobs** page, and the created job is displayed in the job list. You can view the job status in the **Status** column. After a job is successfully submitted, the job status will change from **Submitting** to **Running**. After the execution is complete, the message **Completed** is displayed.

If the job status is **Submission failed** or **Running exception**, the job submission failed or the job did not execute successfully. In this case, you can move the cursor over the status icon in the **Status** column of the job list to view the error details. You can click ❐ to copy error information. After handling the fault based on the provided information, resubmit the job.

NOTE

Other available buttons are as follows:

- **Save As**: Save the created job as a new job.
- **Debug**: Perform job debugging. For details, see **Debugging a Flink Job**.
- **Format**: Format the SQL statements in the editing box.
- **Set as Template**: Set the created SQL statements as a job template.
- **Theme Settings**: Set the theme related parameters, including **Font Size**, **Wrap**, and **Page Style**.

----End

5.2.5 Creating a Flink Jar Job

This section describes how to create a Flink Jar job. You can perform secondary development based on Flink APIs, build your own JAR file, and submit the JAR file to DLI queues. DLI is fully compatible with open-source community APIs. To create a custom Flink job, you need to compile and build application JAR files. You must have a certain understanding of Flink secondary development and have high requirements related to stream computing complexity.

Prerequisites

- Ensure that a dedicated queue has been created.
- When creating a Flink Jar job to access other external data sources, such as OpenTSDB, HBase, Kafka, DWS, RDS, CSS, CloudTable, DCS Redis, and DDS MongoDB, you need to create a cross-source connection to connect the job running queue to the external data source.
  - For details about the external data sources that can be accessed by Flink jobs, see **Datasource Connection and Cross-Source Analysis**.
  - For details about how to create a datasource connection, see **Enhanced Datasource Connections**.

    On the **Queue Management** page, locate the queue you have created, and choose **More** > **Test Address Connectivity** in the **Operation** column to check whether the network connection between the queue and the data source is normal. For details, see **Testing Address Connectivity**.
- When running a Flink Jar job, you need to build the secondary development application code into a Jar package and upload the JAR package to the created OBS bucket. Choose **Data Management** > **Package Management** to create a package. For details, see **Creating a Package**.

  NOTE

  DLI does not support the download function. If you need to modify the uploaded data file, please edit the local file and upload it again.
- Flink dependencies have been built in the DLI server and security hardening has been performed based on the open-source community version. To avoid dependency package compatibility issues or log output and dump issues, be careful to exclude the following files when packaging:
  - Built-in dependencies (or set the package dependency scope to **provided** in Maven or SBT)

- Log configuration files (example, **log4j.properties**/**logback.xml**)
- JAR package for log output implementation (example, **log4j**).

Creating a Flink Jar Job

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 In the upper right corner of the **Flink Jobs** page, click **Create Job**.

Step 3 Specify job parameters.

Table 5-7 Job configuration information

Parameter	Description
Type	Select **Flink Jar**.
Name	Name of a job. Enter 1 to 57 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. **NOTE** The job name must be globally unique.
Description	Description of a job. It can be up to 512 characters long.

Step 4 Click **OK** to enter the page for job information editing.

Step 5 Select a queue. Flink Jar jobs can run only on general queues.

NOTE

- A Flink Jar job can run only on a pre-created dedicated queue.
- If no dedicated queue is available in the **Queue** drop-down list, create a dedicated queue and bind it to the current user.

Step 6 Configure Flink Jar job parameters.

Table 5-8 Parameter description

Name	Description
Application	User-defined package. Before selecting a JAR file to be inserted, upload the corresponding JAR file to the OBS bucket and choose **Data Management > Package Management** to create a package. For details, see **Creating a Package**. For details about the built-in dependencies, see **Flink 1.7.2 Dependencies** and **Flink 1.10 Dependencies**.

Name	Description
Main Class	The name of the JAR package to be loaded, for example, **KafkaMessageStreaming**. ● **Default**: Specified based on the **Manifest** file in the JAR package. ● **Manually assign**: You must enter the class name and confirm the class arguments (separate arguments with spaces). **NOTE** When a class belongs to a package, the main class path must contain the complete package path, for example, **packagePath.KafkaMessageStreaming**.
Class Arguments	List of arguments of a specified class. The arguments are separated by spaces. Flink parameters support replacement of non-sensitive global variables. For example, if you add the global variable **windowsize** in **Global Configuration** > **Global Variables**, you can add the -**windowsSize {{windowsize}}** parameter for the Flink Jar job.
JAR Package Dependencies	Select a user-defined package dependency. The dependent program packages are stored in the classpath directory of the cluster. Before selecting a JAR file to be inserted, upload the corresponding JAR file to the OBS bucket and choose **Data Management > Package Management** to create a package. Select **JAR** as the package type. For details, see **Creating a Package**. For details about the built-in dependencies, see **Flink 1.7.2 Dependencies** and **Flink 1.10 Dependencies**.
Other Dependencies	User-defined dependency files. Other dependency files need to be referenced in the code. Before selecting a dependency file, upload the file to the OBS bucket and choose **Data Management > Package Management** to create a package. The package type is not limited. For details, see **Creating a Package**. You can add the following command to the application to access the corresponding dependency file. In the command, **fileName** indicates the name of the file to be accessed, and **ClassName** indicates the name of the class that needs to access the file. `ClassName.class.getClassLoader().getResource("userData/fileName")`
Flink Version	Before selecting a Flink version, you need to select the queue to which the Flink version belongs. Currently, the following versions are supported: 1.10.

Name	Description
Runtime Configuration	User-defined optimization parameters. The parameter format is **key=value**. Flink optimization parameters support replacement non-sensitive global variable. For example, if you create global variable **phase** in **Global Configuration** > **Global Variables**, optimization parameter **table.optimizer.agg-phase.strategy={{phase}}** can be added to the Flink Jar job.

Step 7 Configure job parameters.

Table 5-9 Parameter description

Name	Description
CUs	One CU has one vCPU and 4-GB memory. The number of CUs ranges from 2 to 400.
Job Manager CUs	Set the number of CUs on a management unit. The value ranges from 1 to 4. The default value is 1.
Parallelism	Maximum number of parallel operators in a job. **NOTE** ● The value must be less than or equal to four times the number of compute units (CUs minus the number of job manager CUs). ● You are advised to set this parameter to a value greater than that configured in the code. Otherwise, job submission may fail.
Task Manager Configuration	Whether to set Task Manager resource parameters If this option is selected, you need to set the following parameters: ● **CU(s) per TM**: Number of resources occupied by each Task Manager. ● **Slot(s) per TM**: Number of slots contained in each Task Manager.
Save Job Log	Whether to save the job running logs to the OBS bucket. **CAUTION** You are advised to select this parameter. Otherwise, no run log is generated after the job is executed. If the job is abnormal, the run log cannot be obtained for fault locating. If this option is selected, you need to set the following parameters: **OBS Bucket**: Select an OBS bucket to store job logs. If the selected OBS bucket is not authorized, click **Authorize**.

Name	Description
Alarm Generation upon Job Exception	Whether to report job exceptions, for example, abnormal job running or exceptions due to an insufficient balance, to users via SMS or email If this option is selected, you need to set the following parameters: **SMN Topic** Select a user-defined SMN topic. For details about how to customize SMN topics, see **Creating a Topic** in the *Simple Message Notification User Guide*.
Auto Restart upon Exception	Whether to enable automatic restart. If this function is enabled, any job that has become abnormal will be automatically restarted. If this option is selected, you need to set the following parameters: ● **Max. Retry Attempts**: maximum number of retry times upon an exception. The unit is **Times/hour**. – **Unlimited**: The number of retries is unlimited. – **Limited**: The number of retries is user-defined. ● **Restore Job from Checkpoint**: Restore the job from the saved checkpoint. If you select this parameter, you also need to set **Checkpoint Path**. **Checkpoint Path**: Select the checkpoint saving path. The checkpoint path must be the same as that you set in the application package. Note that the checkpoint path for each job must be unique. Otherwise, the checkpoint cannot be obtained.

Step 8 Click **Save** on the upper right of the page.

Step 9 Click **Start** on the upper right side of the page. On the displayed **Start Flink Job** page, confirm the job specifications, and click **Start Now** to start the job.

After the job is started, the system automatically switches to the **Flink Jobs** page, and the created job is displayed in the job list. You can view the job status in the **Status** column. After a job is successfully submitted, the job status will change from **Submitting** to **Running**. After the execution is complete, the message **Completed** is displayed.

If the job status is **Submission failed** or **Running exception**, the job submission failed or the job did not execute successfully. In this case, you can move the cursor over the status icon in the **Status** column of the job list to view the error details. You can click ❐ to copy these details. After handling the fault based on the provided information, resubmit the job.

NOTE

Other available buttons are as follows:

Save As: Save the created job as a new job.

----End

5.2.6 Debugging a Flink Job

The job debugging function helps you check the logic correctness of your compiled SQL statements before running a job.

NOTE

- Currently, only Flink SQL jobs support this function.
- The job debugging function is used only to verify the SQL logic and does not involve data write operations.

Procedures

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 In the **Operation** column of the created Flink SQL job, click **Edit**. The page for editing the Flink SQL job is displayed.

For a job that is being created, you can debug the job on the editing page.

Step 3 Click **Debug** above the SQL editing box to parse the edited SQL statements. The **Debugging Parameters** tab is displayed on the right of the page.

- **Dump Bucket**: Select an OBS bucket to save debugging logs. If you select an unauthorized OBS bucket, click **Authorize**.
- **Data Input Mode**: You can select CSV data stored in the OBS bucket or manually enter the data.
  - **OBS (CSV)**

    If you select this value, prepare OBS data first before using DLI. For details, see **Preparing Flink Job Data**. OBS data is stored in CSV format, where multiple records are separated by line breaks and different fields in a single record are separated by commas (,). In addition, you need to select a specific object in OBS as the input source data.
  - **Manual typing**

    If you select this value, compile SQL statements as data sources. In this mode, you need to enter the value of each field in a single record.

Step 4 Click **Start Debugging**. Once debugging is complete, the **Debugging Result** page appears.

- If the debugging result meets the expectation, the job is running properly.
- If the debugging result does not meet the expectation, business logic errors may have occurred. In this case, modify SQL statements and conduct debugging again.

----End

5.2.7 Performing Operations on a Flink Job

After a job is created, you can perform operations on the job as required.

- **Editing a Job**

- **Starting a Job**
- **Stopping a Job**
- **Deleting a Job**
- **Exporting a Job**
- **Importing a Job**
- **Modifying Name and Description**
- **Importing to a Savepoint**
- **Triggering a Savepoint**
- **Runtime Configuration**

Editing a Job

You can edit a created job, for example, by modifying the SQL statement, job name, job description, or job configurations.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 In the row where the job you want to edit locates, click **Edit** in the **Operation** column to switch to the **Edit** page.

Step 3 Edit the job as required.

For details about how to edit a Flink SQL job, see Step 5 to Step 7 in **Creating a Flink SQL Job**.

For details about how to edit a user-defined Flink job, see Step 5 to Step 7 in **Creating a Flink Jar Job**.

----End

Starting a Job

You can start a saved or stopped job.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Use either of the following methods to start jobs:

- Starting a single job

  Select a job and click **Start** in the **Operation** column.

  Alternatively, you can select the row where the job you want to start locates and click **Start** in the upper left of the job list.

- Batch starting jobs

  Select the rows where the jobs you want to start locate and click **Start** in the upper left of the job list.

After you click **Start**, the **Start Flink Jobs** page is displayed.

Step 3 On the **Start Flink Jobs** page, confirm the job information. If they are correct, click **Start Now**.

After a job is started, you can view the job execution result in the **Status** column.

----End

Stopping a Job

You can stop a job in the **Running** or **Submitting** state.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Stop a job using either of the following methods:

- Stopping a job

  Locate the row that contains the job to be stopped, click **More** in the **Operation** column, and select **Stop**.

  Alternatively, you can select the row where the job you want to stop locates and click **Stop** in the upper left of the job list.

- Batch stopping jobs

  Locate the rows containing the jobs you want to stop and click **Stop** in the upper left of the job list.

Step 3 In the displayed **Stop Job** dialog box, click **OK** to stop the job.

NOTE

- Before stopping a job, you can trigger a savepoint to save the job status information. When you start the job again, you can choose whether to restore the job from the savepoint.
- If you select **Trigger savepoint**, a savepoint is created. If **Trigger savepoint** is not selected, no savepoint is created. By default, the savepoint function is disabled.
- The lifecycle of a savepoint starts when the savepoint is triggered and stops the job, and ends when the job is restarted. The savepoint is automatically deleted after the job is restarted.

When a job is being stopped, the job status is displayed in the **Status** column of the job list. The details are as follows:

- **Stopping**: indicates that the job is being stopped.
- **Stopped**: indicates that the job is stopped successfully.
- **Stop failed**: indicates that the job failed to be stopped.

----End

Deleting a Job

If you do not need to use a job, perform the following operations to delete it. A deleted job cannot be restored. Therefore, exercise caution when deleting a job.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Perform either of the following methods to delete jobs:

- Deleting a single job

  Locate the row containing the job you want to delete and click **More > Delete** in the **Operation** column.

Alternatively, you can select the row containing the job you want to delete and click **Delete** in the upper left of the job list.

- Deleting jobs in batches

  Select the rows containing the jobs you want to delete and click **Delete** in the upper left of the job list.

Step 3 Click **Yes**.

----End

Exporting a Job

You can export the created Flink jobs to an OBS bucket.

This mode is applicable to the scenario where a large number of jobs need to be created when you switch to another region, project, or user. In this case, you do not need to create a job. You only need to export the original job, log in to the system in a new region or project, or use a new user to import the job.

> NOTE
>
> When switching to another project or user, you need to grant permissions to the new project or user. For details, see **Managing Flink Job Permissions**.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click **Export Job** in the upper right corner. The **Export Job** dialog box is displayed.

Step 3 Select the OBS bucket where the job is stored. Click **Next**.

Step 4 Select job information you want to export.

By default, configurations of all jobs are exported. You can enable the **Custom Export** function to export configurations of the desired jobs.

Step 5 Click **Confirm** to export the job.

----End

Importing a Job

You can import the Flink job configuration file stored in the OBS bucket to the **Flink Jobs** page of DLI.

This mode is applicable to the scenario where a large number of jobs need to be created when you switch to another region, project, or user. In this case, you do not need to create a job. You only need to export the original job, log in to the system in a new region or project, or use a new user to import the job.

If you need to import a self-created job, you are advised to use the job creation function. For details, see **Creating a Flink SQL job**, **Creating a Flink Jar Job**.

NOTE

- When switching to another project or user, you need to grant permissions to the new project or user. For details, see **Managing Flink Job Permissions**.
- Only jobs whose data format is the same as that of Flink jobs exported from DLI can be imported.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click **Import Job** in the upper right corner. The **Import Job** dialog box is displayed.

Step 3 Select the complete OBS path of the job configuration file to be imported. Click **Next**.

Step 4 Configure the same-name job policy and click next. Click **Next**.

- Select **Overwrite job of the same name**. If the name of the job to be imported already exists, the existing job configuration will be overwritten and the job status switches to **Draft**.
- If **Overwrite job of the same name** is not selected and the name of the job to be imported already exists, the job will not be imported.

Step 5 Ensure that **Config File** and **Overwrite Same-Name Job** are correctly configured. Click **Confirm** to export the job.

----End

Modifying Name and Description

You can change the job name and description as required.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 In the **Operation** column of the job whose name and description need to be modified, choose **More > Modify Name and Description**. The **Modify Name and Description** dialog box is displayed. Change the name or modify the description of a job.

Step 3 Click **OK**.

----End

Triggering a Savepoint

You can click **Trigger Savepoint** for jobs in the **Running** status to save the job status.

Importing to a Savepoint

You can import a savepoint to restore the job status. For details about the savepoint, see **Checkpointing** at the official website of Flink.

You need to select the OBS path of the save point.

Runtime Configuration

You can select **Runtime Configuration** to configure job exception alarms and restart options.

NOTE

Flink SQL jobs and Flink Jar jobs are supported.

1. In the **Operation** column of the Flink job, choose **More** > **Runtime Configuration**.
2. In the **Runtime Configuration** diaglog box, set the following parameters:

Table 5-10 Running parameters

Parameter	Description
Alarm Generation upon Job Exception	Whether to report job exceptions, for example, abnormal job running or exceptions due to an insufficient balance, to users via SMS or email. If this option is selected, you need to set the following parameters: **SMN Topic** Select a user-defined SMN topic. For details about how to customize SMN topics, see **Creating a Topic** in the *Simple Message Notification User Guide*.
Auto Restart upon Exception	Whether to enable automatic restart. If this function is enabled, any job that has become abnormal will be automatically restarted. If this option is selected, you need to set the following parameters: ● **Max. Retry Attempts**: maximum number of retry times upon an exception. The unit is times/hour. – **Unlimited**: The number of retries is unlimited. – **Limited**: The number of retries is user-defined. ● **Restore Job from Checkpoint**: Restore the job from the saved checkpoint. NOTE For Flink streaming SQL jobs, you need to select **Enable Checkpoint** on the job editing page before configuring this parameter. If this parameter is selected, you need to set **Checkpoint Path** for Flink Jar jobs. **Checkpoint Path**: Select the checkpoint saving path. The checkpoint path must be the same as that you set in the application package. Note that the checkpoint path for each job must be unique. Otherwise, the checkpoint cannot be obtained.

5.2.8 Flink Job Details

After creating a job, you can view the job details to learn about the following information:

- **Viewing Job Details**
- **Checking the Job Monitoring Information**
- **Viewing the Task List of a Job**
- **Viewing the Job Execution Plan**
- **Viewing Job Submission Logs**
- **Viewing Job Running Logs**

Viewing Job Details

This section describes how to view job details. After you create and save a job, you can click the job name to view job details, including SQL statements and parameter settings. For a Jar job, you can only view its parameter settings.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job to be viewed. The **Job Detail** tab is displayed.

In the **Job Details** tab, you can view SQL statements, configured parameters.

The following uses a Flink SQL job as an example.

Table 5-11 Description

Parameter	Description
Type	Job type, for example, **Flink SQL**
Name	Flink job name
Description	Description of a Flink job
Status	Running status of a job
Running Mode	If your job runs on a shared queue, this parameter is **Shared**. If your job runs on a custom queue with dedicated resources, this parameter is **Exclusive**.
Queue	If the queue where the job runs is a shared queue, the shared queue is displayed. If the queue where the job runs is a custom queue with dedicated resources, the queue name is displayed.
UDF Jar	This parameter is displayed when a non-shared queue is selected for the job and **UDF Jar** is configured.
Runtime Configuration	Displayed when a user-defined parameter is added to a job

Parameter	Description
CUs	Number of CUs configured for a job
Job Manager CUs	Number of job manager CUs configured for a job.
Parallelism	Number of jobs that can be concurrently executed by a Flink job
CU(s) per TM	Number of CUs occupied by each Task Manager configured for a job
Slot(s) per TM	Number of Task Manager slots configured for a job
OBS Bucket	OBS bucket name. After **Enable Checkpointing** and **Save Job Log** are enabled, checkpoints and job logs are saved in this bucket.
Save Job Log	Whether the job running logs are saved to OBS
Alarm Generation upon Job Exception	Whether job exceptions are reported
SMN Topic	Name of the SMN topic. This parameter is displayed when **Alarm Generation upon Job Exception** is enabled.
Auto Restart upon Exception	Whether automatic restart is enabled.
Max. Retry Attempts	Maximum number of retry times upon an exception. **Unlimited** means the number is not limited.
Restore Job from Checkpoint	Whether the job can be restored from a checkpoint
ID	Job ID
Savepoint	OBS path of the savepoint
Enable Checkpointing	Whether checkpointing is enabled
Checkpoint Interval	Interval between storing intermediate job running results to OBS. The unit is second.
Checkpoint Mode	Checkpoint mode. Available values are as follows: ● **At least once**: Events are processed at least once. ● **Exactly once**: Events are processed only once.
Idle State Retention Time	Defines for how long the state of a key is retained without being updated before it is removed in GroupBy or Window.

Parameter	Description
Dirty Data Policy	Policy for processing dirty data. The value is displayed only when there is a dirty data policy. Available values are as follows: **Ignore** **Trigger a job exception** **Save**
Dirty Data Dump Address	OBS path for storing dirty data when **Dirty Data Policy** is set to **Save**.
Created	Time when a job is created
Updated	Time when a job was last updated

----End

Checking the Job Monitoring Information

You can use Cloud Eye to view details about job data input and output.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job you want. The job details are displayed.

Click **Job Monitoring** in the upper right corner of the page to switch to the Cloud Eye console.

The following table describes monitoring metrics related to Flink jobs.

Table 5-12 Monitoring metrics related to Flink jobs

Name	Description
Flink Job Data Read Rate	Displays the data input rate of a Flink job for monitoring and debugging. Unit: record/s.
Flink Job Data Write Rate	Displays the data output rate of a Flink job for monitoring and debugging. Unit: record/s.
Flink Job Total Data Read	Displays the total number of data inputs of a Flink job for monitoring and debugging. Unit: records
Flink Job Total Data Write	Displays the total number of output data records of a Flink job for monitoring and debugging. Unit: records
Flink Job Byte Read Rate	Displays the number of input bytes per second of a Flink job. Unit: byte/s
Flink Job Byte Write Rate	Displays the number of output bytes per second of a Flink job. Unit: byte/s

Name	Description
Flink Job Total Read Byte	Displays the total number of input bytes of a Flink job. Unit: byte
Flink Job Total Write Byte	Displays the total number of output bytes of a Flink job. Unit: byte
Flink Job CPU Usage	Displays the CPU usage of Flink jobs. Unit: %
Flink Job Memory Usage	Displays the memory usage of Flink jobs. Unit: %
Flink Job Max Operator Latency	Displays the maximum operator delay of a Flink job. The unit is **ms**.
Flink Job Maximum Operator Backpressure	Displays the maximum operator backpressure value of a Flink job. A larger value indicates severer backpressure. **0**: OK **50**: low **100**: high

----End

Viewing the Task List of a Job

You can view details about each task running on a job, including the task start time, number of received and transmitted bytes, and running duration.

NOTE

If the value is **0**, no data is received from the data source.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job you want. The job details are displayed.

Step 3 On **Task List** and view the node information about the task.

View the operator task list. The following table describes the task parameters.

Table 5-13 Parameter description

Parameter	Description
Name	Name of an operator.
Duration	Running duration of an operator.
Max Concurrent Jobs	Number of parallel tasks in an operator.

Parameter	Description
Task	Operator tasks are categorized as follows: ● The digit in red indicates the number of failed tasks. ● The digit in light gray indicates the number of canceled tasks. ● The digit in yellow indicates the number of tasks that are being canceled. ● The digit in green indicates the number of finished tasks. ● The digit in blue indicates the number of running tasks. ● The digit in sky blue indicates the number of tasks that are being deployed. ● The digit in dark gray indicates the number of tasks in a queue.
Status	Status of an operator task.
Back Pressure Status	Working load status of an operator. Available options are as follows: ● **OK**: indicates that the operator is in normal working load. ● **LOW**: indicates that the operator is in slightly high working load. DLI processes data quickly. ● **HIGH**: indicates that the operator is in high working load. The data input speed at the source end is slow.
Delay	Duration from the time when source data starts being processed to the time when data reaches the current operator. The unit is millisecond.
Sent Records	Number of data records sent by an operator.
Sent Bytes	Number of bytes sent by an operator.
Received Bytes	Number of bytes received by an operator.
Received Records	Number of data records received by an operator.
Started	Time when an operator starts running.
Ended	Time when an operator stops running.

----End

Viewing the Job Execution Plan

You can view the execution plan to understand the operator stream information about the running job.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job you want. The job details are displayed.

Step 3 Click the **Execution Plan** tab to view the operator flow direction.

Click a node. The corresponding information is displayed on the right of the page.

- Scroll the mouse wheel to zoom in or out.
- The stream diagram displays the operator stream information about the running job in real time.

----End

Viewing Job Submission Logs

You can view the submission logs to locate the fault.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job you want. The job details are displayed.

Step 3 In the **Commit Logs** tab, view the information about the job submission process.

----End

Viewing Job Running Logs

You can view the run logs to locate the faults occurring during job running.

Step 1 In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.

Step 2 Click the name of the job you want. The job details are displayed.

Step 3 On the **Run Log** tab page, you can view the **Job Manager** and **Task Manager** information of the running job.

Information about JobManager and TaskManager is updated every minute. Only run logs of the last minute are displayed by default.

If you select an OBS bucket for saving job logs during the job configuration, you can switch to the OBS bucket and download log files to view more historical logs.

If the job is not running, information on the **Task Manager** page cannot be viewed.

----End

5.3 Spark Job Management

5.3.1 Spark Job Management

Based on the open-source Spark, DLI optimizes performance and reconstructs services to be compatible with the Apache Spark ecosystem and interfaces, and executes batch processing tasks.

DLI also allows you to use Spark jobs to access DLI metadata.

Spark job management provides the following functions:

- **Creating a Spark Job**
- **Re-executing a Job**
- **Searching for a Job**
- **Terminating a Job**

In addition, you can click **Quick Links** to switch to the details on *User Guide*.

Spark Jobs Page

On the **Overview** page, click **Spark Jobs** to go to the SQL job management page. Alternatively, you can click **Job Management** > **Spark Jobs**. The page displays all Spark jobs. If there are a large number of jobs, they will be displayed on multiple pages. DLI allows you to view jobs in all statuses.

Table 5-14 Job management parameters

Parameter	Description
Job ID	ID of a submitted Spark job, which is generated by the system by default.
Name	Name of a submitted Spark job.
Queues	Queue where the submitted Spark job runs
Username	Name of the user who executed the Spark job
Status	Job status. The following values are available: ● **Starting**: The job is being started. ● **Running**: The job is being executed. ● **Failed**: The session has exited. ● **Finished**: The session is successfully executed. ● **Restoring**: The job is being restored.
Created	Time when a job is created. Jobs can be displayed in ascending or descending order of the job creation time.
Last Modified	Time when a job is completed.

Parameter	Description
Operation	● **Edit**: You can modify the current job configuration and re-execute the job. ● **SparkUI**: After you click this button, the Spark job execution page is displayed. NOTE ● The SparkUI page cannot be viewed for jobs in the **Starting** state. ● Currently, only the latest 100 job information records are displayed on the SparkUI of DLI. ● **Terminate Job**: Cancel a job that is being started or running. ● **Re-execute**: Run the job again. ● **Archive Log**: Save job logs to the temporary bucket created by DLI. ● **Export Log**: Export logs to the created OBS bucket. NOTE ● You have the permission to create OBS buckets. ● If the job is in the **Running** state, logs cannot be exported. ● **Commit Log**: View the logs of submitted jobs. ● **Driver Log**: View the logs of running jobs.

Re-executing a Job

On the **Spark Jobs** page, click **Edit** in the **Operation** column of the job. On the Spark job creation page that is displayed, modify parameters as required and execute the job.

Searching for a Job

On the **Spark Jobs** page, select **Status** or **Queues**. The system displays the jobs that meet the filter condition in the job list.

Terminating a Job

On the **Spark Jobs** page, choose **More** > **Terminate Job** in the **Operation** column of the job that you want to stop.

Exporting Logs

On the **Spark Jobs** page, choose **More** > **Export Log** in the Operation column of the corresponding job. In the dialog box that is displayed, enter the path of the created OBS bucket and click **OK**.

5.3.2 Creating a Spark Job

DLI provides fully-managed Spark computing services by allowing you to execute Spark jobs.

On the **Overview** page, click **Create Job** in the upper right corner of the **Spark Jobs** tab or click **Create Job** in the upper right corner of the **Spark Jobs** page. The Spark job editing page is displayed.

On the Spark job editing page, a message is displayed, indicating that a temporary DLI data bucket will be created. The created bucket is used to store temporary data generated by DLI, such as job logs and job results. You cannot view job logs if you choose not to create it. The bucket will be created and the default bucket name is used.

If you do not need to create a DLI temporary data bucket and do not want to receive this message, select **Do not show again** and click **Cancel**.

Prerequisites

- You have uploaded the dependencies to the corresponding OBS bucket on the **Data Management > Package Management** page. For details, see **Creating a Package**.
- Before creating a Spark job to access other external data sources, such as OpenTSDB, HBase, Kafka, DWS, RDS, CSS, CloudTable, DCS Redis, and DDS MongoDB, you need to create a cross-source connection to enable the network between the job running queue and external data sources.
  - For details about the external data sources that can be accessed by Spark jobs, see **Datasource Connection and Cross-Source Analysis**.
  - For details about how to create a datasource connection, see **Enhanced Datasource Connections**.

    On the **Queue Management** page, locate the queue you have created, and choose **More** > **Test Address Connectivity** in the **Operation** column to check whether the network connection between the queue and the data source is normal. For details, see **Testing Address Connectivity**.

GUI Description

- Navigation bar on the left

  On the Spark job creation page, the navigation tree on the left contains the **Queues** and **Packages** tab pages.

Table 5-15 Description of buttons in the left navigation pane

No.	Tab/Button	Tab/Button Name	Description
1	Queues	Queues	Display existing queues.
2	Packages	Packages	Display existing packages.
3	Templates	Templates	Display job templates.
4	⊕	Create	Create a queue or a package.

No.	Tab/ Button	Tab/Button Name	Description
5	↻	Refresh	Refresh the lists of existing queues and packages.
6	🔍	Search for packages by name.	On the **Packages** tab page, enter a package name for search.

- Job editing window

  In the job editing window, you can set parameters in **Fill Form** mode or **Write API** mode.

  The following uses the **Fill Form** as an example. In **Write API** mode, refer to the *Data Lake Insight API Reference* for parameter settings.

  – **Select a Queue**: For details about the parameters, see **Table 5-16**.

  Table 5-16 Queue parameters

Parameter	Description
Queues	Select the target queue from the drop-down list box.

  – **Job Configurations**: Refer to **Table 5-17** for details.

  Table 5-17 Job configuration parameters

Parameter	Description
Job Name	Set a job name.
Application	Select the package to be executed. The value can be **.jar** or **.py**.
Main Class	Enter the name of the main class. When the application type is **.jar**, the main class name cannot be empty.
Application Parameters	User-defined parameters. Separate multiple parameters by **Enter**. These parameters can be replaced by global variables. For example, if you create a global variable **batch_num** on the **Global Configuration** > **Global Variables** page, you can use **{{batch_num}}** to replace a parameter with this variable after the job is submitted.

Parameter	Description
Spark Arguments	Enter a parameter in the format of **key=value**. Press Enter to separate multiple key-value pairs. These parameters can be replaced by global variables. For example, if you create a global variable **custom_class** on the **Global Configuration** > **Global Variables** page, you can use **"spark.sql.catalog"={{custom_class}}** to replace a parameter with this variable after the job is submitted. **NOTE** The JVM garbage collection algorithm cannot be customized for Spark jobs.
Job Type	Type of the Spark image used by a job. This parameter is mandatory when you select a CCE queue. The values are as follows: ● **Basic**: Basic images provided by DLI. Select this option for non-AI jobs. ● **AI-enhanced**: AI images provided by DLI. Select this option for AI jobs. ● **Image**: Custom Spark images. Select an existing image name and version on SWR.
JAR Package Dependencies	JAR file on which the Spark job depends. You can enter the JAR package name or the corresponding OBS path. The format is as follows: **obs://*Bucket name/Folder name/Package name***.
Python File Dependencies	py-files on which the Spark job depends. You can enter the Python file name or the corresponding OBS path of the Python file. The format is as follows: **obs://*Bucket name/Folder name/File name***.
Other Dependencies	Other files on which the Spark job depends. You can enter the name of the dependency file or the corresponding OBS path of the dependency file. The format is as follows: **obs://*Bucket name/Folder name/File name***.
Group Name	If you select a group when creating a package, you can select all the packages and files in the group. For details about how to create a package, see **Creating a Package**.
Access Metadata	Whether to access metadata through Spark jobs. .
Retry upon Failure	Indicates whether to retry a failed job. If you select **Yes**, you need to set the following parameters: **Maximum Retries**: Maximum number of retry times. The maximum value is **100**.

Parameter	Description
Advanced Settings	- Skip - Configure – **Select Dependency Resources**: For details about the parameters, see **Table 5-18**. – **Configure Resources**: For details about the parameters, see **Table 5-19**.

Table 5-18 Parameters for selecting dependency resources

Parameter	Description
Module Name	Dependency modules provided by DLI for executing datasource connection jobs. To access different services, you need to select different modules. - MRS HBase: sys.datasource.hbase - DDS: sys.datasource.mongo - MRS OpenTSDB: sys.datasource.opentsdb - DWS: sys.datasource.dws - RDS MySQL: sys.datasource.rds - RDS PostGre: sys.datasource.rds - DCS: sys.datasource.redis - CSS: sys.datasource.css
Resource Package	JAR package on which the Spark job depends.

Table 5-19 Resource specification parameters

Parameter	Description
Resource Specifications	Select a resource specification from the drop-down list box. The system provides three resource specifications for you to select. The following configuration items in the resource specifications can be modified: ● Executor Memory ● Executor Cores ● Executors ● Driver Cores ● Driver Memory If modified, your modified settings of the items are used.
Executor Memory	Customize the configuration item based on the selected resource specifications.
Executor Cores	Customize the configuration item based on the selected resource specifications.
Executors	Customize the configuration item based on the selected resource specifications.
Driver Cores	Customize the configuration item based on the selected resource specifications.
Driver Memory	Customize the configuration item based on the selected resource specifications.

NOTE

Spark job parameter calculation:

- Number of CUs = Number of driver CPU cores + Number of executors x Number of executor CPU cores

  The cluster management plane and driver use some CU resources. Number of Executors * Number of Executor Cores must be smaller than the number of computing CUs of the queue.
- Memory = Driver memory + (Number of Executors x Executor memory)

Creating a Spark Job

1. In the Spark job editing window, set related parameters. For details, see the description of the Spark job editing window.
2. Click **Execute** in the upper right corner of the Spark job editing window to submit the job. The message "The batch job is submitted." is displayed.
3. (Optional) Go to the **Spark Jobs** page to view the status and logs of the submitted Spark job.

6 Queue Management

6.1 Overview

Queue

Queues in DLI are computing resources, which are the basis for using DLI. All executed jobs require computing resources.

Currently, DLI provides two types of queues, **For SQL** and **For general use**. SQL queues are used to run SQL jobs. General-use queues are compatible with Spark queues of earlier versions and are used to run Spark and Flink jobs.

NOTE

- The SQL queue automatically restarts every 12 hours.
- Queues of different types cannot be directly switched. If you need to change a queue, purchase a new queue.
- The default queue is preset. Do not delete it.

Difference Between Computing and Storage Resources

Table 6-1 Difference between computing and storage resources

Resource	How to Obtain	Function
Compute resource	Create queue on the DLI management console.	Used for executing queries.
Storage resource	DLI has a 5 GB quota.	Used for storing data in the database and DLI tables.

NOTE

- Storage resources are internal storage resources of DLI for storing database and DLI tables and represent the amount of data stored in DLI.
- By default, DLI provides a 5 GB quota for storage resources.
- A queue named **default** is preset in DLI. If you are uncertain about the required queue capacity or have no available queue capacity to run queries, you can execute jobs using this queue.
- The **default** queue is used only for user experience. It may be occupied by multiple users at a time. Therefore, it is possible that you fail to obtain the resource for related operations. You are advised to use a self-built queue to execute jobs.

Dedicated Queue

Resources of a dedicated queue are not released when the queue is idle. That is, resources are reserved regardless of whether the queue is used. Dedicated queues ensure that resources exist when jobs are submitted.

Elastic Scaling

DLI allows you to flexibly scale in or out queues on demand. After a queue with specified specifications is created, you can scale it in and out as required.

To change the queue specifications, see **Elastic Scaling**.

NOTE

Scaling can be performed for a newly created queue only when jobs are running on this queue.

Scheduled Elastic Scaling

DLI allows you to schedule tasks for periodic queue scaling. After creating a queue, the scheduled scaling tasks can be executed.

You can schedule an automatic scale-out/scale-in based on service requirements. The system periodically triggers queue scaling. For details, see **Scheduling CU Changes**.

NOTE

Scaling can be performed for a newly created queue only when jobs are running on this queue.

Automatic Queue Scaling

Flink jobs use queues. DLI can automatically trigger scaling for jobs based on the job size.

NOTE

Scaling can be performed for a newly created queue only when there are jobs running on this queue.

Queue Management Page

Queue Management provides the following functions:

- **Managing Permissions**
- **Creating a Queue**
- **Deleting a Queue**
- **Modifying CIDR Block**
- **Elastic Scaling**
- **Scheduling CU Changes**
- **Testing Address Connectivity**
- **Creating a Topic for Key Event Notifications**

NOTE

To receive notifications when a DLI job fails, **SMN Administrator** permissions are required.

The queue list displays all queues created by you and the **default** queue. Queues are listed in chronological order by default in the queue list, with the most recently created queues displayed at the top.

Table 6-2 Parameter description

Parameter	Description
Name	Name of a queue.
Type	Queue type. ● For SQL ● For general purpose ● Spark queue (compatible with earlier versions)
Specifications	Queue size. Unit: CU CU is the pricing unit of queues. A CU consists of 1 vCPU and 4-GB memory. The computing capabilities of queues vary with queue specifications. The higher the specifications, the stronger the computing capability.
Actual CUs	Actual size of the current queue.
Elastic Scaling	Target CU value for scheduled scaling, or the maximum and minimum CU values of the current specifications.
Username	Queue owner
Description	Description of a queue specified during queue creation. If no description is provided, **--** is displayed.

Parameter	Description
Operation	● **Delete**: Allow you to delete the selected queue. You cannot delete a queue where there are running jobs or jobs are being submitted. ● **Manage Permissions**: You can view the user permissions corresponding to the queue and grant permissions to other users. ● **More** – **Restart**: Forcibly restart a queue. **NOTE** Only the SQL queue has the **Restart** operation. – **Elastic Scaling**: You can select **Scale-out** or **Scale-in** as required. The number of CUs after modification must be an integer multiple of 16. – **Schedule CU Changes**: You can set different queue sizes at different time or in different periods based on the service period or usage. The system automatically performs scale-out or scale-in as scheduled. The **After Modification** value must be an integer multiple of 16. – **Modifying CIDR Block**: When DLI enhanced datasource connection is used, the CIDR block of the DLI queue cannot overlap with that of the data source. You can modify the CIDR block as required. – **Test Address Connectivity**: Test whether the queue is reachable to the specified address. The domain name and IP address are supported. The port can be specified.

6.2 Queue Permission Management

Scenarios

- You can isolate queues allocated to different users by setting permissions to ensure data query performance.
- The administrator and queue owner have all permissions, which cannot be set or modified by other users.

Operations

Step 1 On the top menu bar of the DLI management console, click **Resources** > **Queue Management**.

Step 2 Select the queue to be configured and choose **Manage Permissions** in the **Operation** column. The **User Permission Info** area displays the list of users who have permissions on the queue.

You can assign queue permissions to new users, modify permissions for users who have some permissions of a queue, and revoke all permissions of a user on a queue.

- Assign permissions to a new user.

  A new user does not have permissions on the queue.

  a. Click **Set Permission** on the right of **User Permissions** page. The **Set Permission** dialog box is displayed.
  b. Specify **Username** and select corresponding permissions.
  c. Click **OK**.

     Table 6-3 describes the related parameters.

Table 6-3 Parameter description

Parameter	Description
Username	Name of the authorized user. **NOTE** The username is an existing IAM user name and has logged in to the DLI management console.
Permission Settings	▪ **Delete Queues**: This permission allows you to delete the queue. ▪ **Submit Jobs**: This permission allows you to submit jobs using this queue. ▪ **Terminate Jobs**: This permission allows you to terminate jobs submitted using this queue. ▪ **Grant Permission**: This permission allows you to grant queue permissions to other users. ▪ **Revoke Permission**: This permission allows you to revoke the queue permissions that other users have but cannot revoke the queue owner's permissions. ▪ **View Other User's Permissions**: This permission allows you to view the queue permissions of other users. ▪ **Restart Queues**: This permission allows you to restart queues. ▪ **Modify Queue Specifications**: This permission allows you to modify queue specifications.

- To assign or revoke permissions of a user who has some permissions on the queue, perform the following steps:

  a. In the list under **User Permission Info** for a queue, select the user whose permissions need to be modified and click **Set Permission** in the **Operation** column.
  b. In the displayed **Set Permission** dialog box, modify the permissions of the current user. Table 6-3 lists the detailed permission descriptions.

     If all options under **Set Permission** are gray, you are not allowed to change permissions on this queue. You can apply to the administrator, queue owner, or other authorized users for queue permission granting and revoking.

c. Click **OK**.

- To revoke all permission of a user on the queue, perform the following steps:

  In the user list under **Permission Info**, select the user whose permission needs to be revoked and click **Revoke Permission** under **Operation**. In the **Revoke Permission** dialog box, click **OK**. All permissions on this queue are revoked.

----End

6.3 Creating a Queue

Before executing a job, you need to create a queue.

NOTE

- If you use a sub-account to create a queue for the first time, log in to the DLI management console using the main account and keep records in the DLI database before creating a queue.
- It takes 6 to 10 minutes for a job running on a new queue for the first time.
- After a queue is created, if no job is run within one hour, the system releases the queue.

Procedure

1. You can create a queue on the **Overview**, **SQL Editor**, or **Queue Management** page.
   - In the upper right corner of the **Overview** page, click Create Queue.
   - To create a queue on the **Queue Management** page:
     i. In the navigation pane of the DLI management console, choose **Resources** >**Queue Management**.
     ii. In the upper right corner of the **Queue Management** page, click **Create Queue** to create a queue.
   - To create a queue on the **SQL Editor** page:
     i. In the navigation pane of the DLI management console, click **SQL Editor**.
     ii. On the left pane of the displayed **SQL Editor** page, click ⊕ to the right of **Queues**.
2. In the displayed Purchase QueueCreate Queue dialog box, set related parameters by referring to **Table 6-4**.

Table 6-4 Parameters

Parameter	Description
Name	Name of a queue. ● The queue name can contain only digits, letters, and underscores (_), but cannot contain only digits, start with an underscore (_), or be left unspecified. ● The length of the name cannot exceed 128 characters. **NOTE** The queue name is case-insensitive. Uppercase letters will be automatically converted to lowercase letters.
Type	● **For SQL**: compute resources used for SQL jobs. ● **For general purpose**: compute resources used for Spark and Flink jobs. **NOTE** In **Dedicated Resource Mode**, you can create enhanced datasource connections.
Specifications	Select queue specifications as required. A CU includes one core and 4 GB memory. You can set the total number of CUs on all compute nodes of a queue. DLI automatically allocates the memory and vCPUs for each node. ● Fixed specifications include **16 CUs**, **64 CUs**, **256 CUs**, and **512 CUs**. ● **Custom**: Set the specifications as required.
Description	Description of the queue to be created. The length of the queue name cannot exceed 256 characters.
Advanced Settings	In the **Queue Type** area, select **Dedicated Resource Mode** and then click **Advanced Settings**. ● **Default**: The system automatically configures the parameter. ● **Custom** **CIDR Block**: You can specify the CIDR block. For details, see **Modifying the Queue CIDR Block**. If DLI enhanced datasource connection is used, the CIDR block of the DLI queue cannot overlap with that of the data source. **Queue Type**: When running an AI-related SQL job, select **AI-enhanced**. When running other jobs, select **Basic**.

3. Click **Create Now** to create a queue.

   After a queue is created, you can view and select the queue for use on the **Queue Management** page.

   NOTE

   It takes 6 to 10 minutes for a job running on a new queue for the first time.

6.4 Deleting a Queue

You can delete a queue based on actual conditions.

NOTE

- This operation will fail if there are jobs in the **Submitting** or **Running** state on this queue.
- Deleting a queue does not cause table data loss in your database.

Procedure

Step 1 On the left of the DLI management console, click **Resources** >**Queue Management**.

Step 2 Locate the row where the target queue locates and click **Delete** in the **Operation** column.

NOTE

If **Delete** in the **Operation** column is gray, the current user does not have the permission of deleting the queue. You can apply to the administrator for the permission.

Step 3 In the displayed dialog box, click **OK**.

----End

6.5 Modifying the CIDR Block

If the CIDR block of the DLI queue conflicts with that of the user data source, you can change the CIDR block of the queue.

NOTE

If the queue whose CIDR block is to be modified has jobs that are being submitted or running, or the queue has been bound to enhanced datasource connections, the CIDR block cannot be modified.

Procedure

1. On the left of the DLI management console, click **Resources** >**Queue Management**.
2. Select the queue to be modified and click **Modify CIDR Block** in the **Operation** column.
3. Enter the required CIDR block and click **OK**.

6.6 Elastic Scaling

Prerequisites

Elastic scaling can be performed for a newly created queue only when there were jobs running in this queue.

Precautions

- If **Status of queue xxx is assigning, which is not available** is displayed on the **Elastic Scaling** page, the queue can be scaled only after the queue resources are allocated.

Scaling Out

If the current queue specifications do not meet service requirements, you can add the number of CUs to scale out the queue.

NOTE

Scale-out is time-consuming. After you perform scale-out on the **Elastic Scaling** page of DLI, wait for about 10 minutes. The duration is related to the CU amount to add. After a period of time, refresh the **Queue Management** page and check whether values of **Specifications** and **Actual CUs** are the same to determine whether the scale-out is successful. Alternatively, on the **Job Management** page, check the status of the **SCALE_QUEUE** SQL job. If the job status is **Scaling**, the queue is being scaled out.

The procedure is as follows:

1. On the left of the DLI management console, click **Resources** > **Queue Management**.
2. Select the queue to be scaled out and choose **More > Elastic Scaling** in the **Operation** column.
3. On the displayed page, select **Scale-out** for **Operation** and set the scale-out amount.
4. Click .

Scaling In

If the current queue specifications are too much for your computing service, you can reduce the number of CUs to scale in the queue.

NOTE

- Scale-in is time-consuming. After you perform scale-in on the **Elastic Scaling** page of DLI, wait for about 10 minutes. The duration is related to the CUs amount to reduce. After a period of time, refresh the **Queue Management** page and check whether values of **Specifications** and **Actual CUs** are the same to determine whether the scale-in is successful. Alternatively, on the **Job Management** page, check the status of the **SCALE_QUEUE** SQL job. If the job status is **Scaling**, the queue is being scaled in.
- The system may not fully scale in the queue to the target size. If the current queue is in use or the service volume of the queue is large, the scale-in may fail or only partial specifications may be reduced.
- By default, the minimum number of CUs is **16**. That is, when the queue specifications are **16 CUs**, you cannot scale in the queue.

The procedure is as follows:

1. On the left of the DLI management console, click **Resources** > **Queue Management**.
2. Select the queue to be scaled out and choose **More > Elastic Scaling** in the **Operation** column.

3. On the displayed page, select **Scale-in** for **Operation** and set the scale-in amount.
4. Click .

6.7 Scheduling CU Changes

Scenario

When services are busy, you might need to use more compute resources to process services in a period. After this period, you do not require the same amount of resources. If the purchased queue specifications are small, resources may be insufficient during peak hours. If the queue specifications are large, resources may be wasted.

DLI provides scheduled tasks for elastic scale-in and -out in the preceding scenario. You can set different queue sizes (CUs) at different time or in different periods based on your service period or usage and the existing queue specifications to meet your service requirements and reduce costs.

Precautions

- Periodic scaling can be performed for a newly created queue only when there were jobs running in this queue.
- Scheduled scaling tasks are available only for a queue with more than 64 CUs. That is, the minimum specifications of a queue are 64 CUs.
- A maximum of 12 scheduled tasks can be created for each queue.
- When each scheduled task starts, the actual start time of the specification change has a deviation of 5 minutes. It is recommended that the task start time be at least 20 minutes earlier than the time when the queue is actually used.
- The interval between two scheduled tasks must be at least 2 hours.
- Changing the specifications of a queue is time-consuming. The time required for changing the specifications depends on the difference between the target specifications and the current specifications. You can view the specifications of the current queue on the **Queue Management** page.
- If a job is running in the current queue, the queue may fail to be scaled in to the target CU amount value. Instead, it will be scaled in to a value between the current queue specifications and the target specifications. The system will try to scale in again 1 hour later until the next scheduled task starts.
- If a scheduled task does not scale out or scale in to the target CU amount value, the system triggers the scaling plan again 15 minutes later until the next scheduled task starts.

Creating Periodic Task

- If only scale-out or scale-in is required, you need to create only one task for changing specifications. Set the **Task Name**, **Final CU Count**, and **Executed** parameters. For details, see **Table 6-5**.

- To set both scale-out and scale-in parameters, you need to create two periodic tasks, and set the **Task Name**, **Final CU Count**, and **Executed** parameters. For details, see **Table 6-5**.

The procedure is as follows:

1. On the left of the DLI management console, click **Resources** > **Queue Management**.
2. Locate the queue for which you want to schedule a periodic task for elastic scaling, and choose **More** > **Schedule CU Changes** in the **Operation** column.
3. On the displayed page, click **Create Periodic Task** in the upper right corner.
4. On the **Create Periodic Task** page, set the required parameters. Click **OK**.

Table 6-5 Parameter description

Parameter	Description
Task Name	Enter the name of the periodic task. ● The task name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_) or be left unspecified. ● The name can contain a maximum of 128 characters.
Enable Task	Whether to enable periodic elastic scaling. The task is enabled by default. If disabled, the task will not be triggered on time.
Validity Period	Set the time segment for executing the periodic task, including **Date** and **Time**. If there is no time segment restriction, leave this parameter empty, indicating that the task takes effect permanently. If you need to specify the time segment for the task to take effect, set this parameter based on the service requirements. **NOTE** ● The start time of the **Validity Period** must be later than the current system time. ● If only scale-out is configured, the system does not automatically scale in after the **Validity Period** expires. You need to manually modify or configure a periodic scale-in task. Similarly, if only scale-in is configured, you need to manually modify or configure a periodic scale-out task. That is, a scheduled scaling task is executed at a time. ● If both scale-out and scale-in are configured, the system scales in or out resources based on the configured queue specifications within the validity period. After the validity period expires, the system retains the last configured queue specifications.
Actual CUs	Queue specifications before scale-in or scale-out.

Parameter	Description
Final CUs	Specifications after the queue is scaled in or out. **NOTE** ● By default, the maximum specifications of a queue are **512 CUs**. ● The minimum queue specifications for scheduled scaling are 64 CUs. That is, only when **Actual CUs** are more than 64 CUs, the scheduled scaling can be performed. ● The value of **Actual CUs** must be a multiple of 16.
Repeat	Time when scheduled scale-out or scale-in is repeat. Scheduled tasks can be scheduled by week in **Repeat**. ● By default, this parameter is not configured, indicating that the task is executed only once at the time specified by **Executed**. ● If you select all, the plan is executed every day. ● If you select some options of **Repeat**, the plan is executed once a week at all specified days. **NOTE** ● You do not need to set this parameter if you only need to perform scale-in or scale-out once. ● If you have set scaling, you can set **Repeat** as required. You can also set the repeat period together with the validity period.
Executed	Time when scheduled scale-out or scale-in is performed ● When each scheduled task starts, the actual start time of the specification change has a deviation of 5 minutes. It is recommended that the task start time be at least 20 minutes earlier than the time when the queue is actually used. ● The interval between two scheduled tasks must be at least 2 hours.

After a periodic task is created, you can view the specification change of the current queue and the latest execution time on the page for scheduling CU changes.

Alternatively, on the **Queue Management** page, check whether the **Specifications** change to determine whether the scaling is successful.

You can also go to the **Job Management** page and check the status of the **SCALE_QUEUE** job. If the job status is **Scaling**, the queue is being scaled in or out.

Modifying a Scheduled Task

If the configured scheduled task does not meet service requirements, you can click **Modify** in the **Operation** column of the task on the **Schedule CU Changes** page to modify the task configuration.

Deleting a Scheduled Task

If you do not need a scheduled task for queue scaling, click **Delete** in the **Operation** column of the task on the **Schedule CU Changes** page. In the displayed **Delete Periodic Task** dialog box, click **Yes**.

6.8 Testing Address Connectivity

It can be used to test the connectivity between the DLI queue and the peer IP address specified by the user in common scenarios, or the connectivity between the DLI queue and the peer IP address bound to the datasource connection in cross-source connection scenarios. The operation is as follows:

1. On the **Queue Management** page, locate the row that contains the target queue and choose **More > Test Address Connectivity** in the **Operation** column.
2. On the **Test Address Connectivity** page, enter the address to be tested. The domain name and IP address are supported, and the port number can be specified.
3. Click **Test**.

   If the test address is reachable, a message is displayed on the page, indicating that the address is reachable.

   If the test address is unreachable, the system displays a message indicating that the address is unreachable. Check the network configurations and try again. Network configurations include the VPC peering and the datasource connection. Check whether they have been activated.

6.9 Creating a Message Notification Topic

Once you have created a message notification topic, you can **Add subscription** of the topic on the **Topic Management** page of the Simple Message Notification service. You can select different ways (such as text messages or emails) to subscribe. After the subscription succeeds, any job failure will automatically be sent to your subscription endpoints. The operation is as follows:

1. On the **Queue Management** page, click **Create SMN Topic** on the upper left side. The **Create SMN Topic** dialog box is displayed.
2. Select a queue and click **OK**. The dialog box is displayed.

   NOTE

   - You can select a single queue or all queues. If the subscription endpoints of a single queue are different from those of all queues, when a single queue and all queues are both selected, the SMN of all queues does not include the message of the single queue.
   - After a notification topic is created, you will receive a notification only when a session or a batch job fails to be created.
3. Click **Topic Management** to go to the **Topic Management** page of the SMN service.
4. In the **Operation** column of the corresponding topic in , click **Add Subscription**. Select **Protocol** to determine the subscription mode.

  - If you select **Email**, enter the email address for receiving the confirmation email in the **Endpoint** text box.

5. After you click the link in the email, you will receive a message indicating that the subscription is successful.
6. Return to the **Topic Management** page and click **Subscriptions** on the left. If the subscription status is **Confirmed**, the subscription is successful.

7 Data Management

7.1 Databases and Tables

7.1.1 Overview

DLI database and table management provide the following functions:

- **Database Permission Management**
- **Table Permission Management**
- **Creating a Database or a Table**
- **Deleting a Database or a Table**
- **Changing the Owners of Databases and Tables**
- **Importing Data**
- **Exporting Data**
- **Viewing Metadata**
- **Previewing Data**

Difference Between DLI Tables and OBS Tables

- Data stored in DLI tables is applicable to delay-sensitive services, such as interactive queries.
- Data stored in OBS tables is applicable to delay-insensitive services, such as historical data statistics and analysis.

Databases and Tables Page

The **Databases and Tables** page displays all created databases. You can view the database information, such as the owner and the number of tables.

Table 7-1 Database and table management parameters

Parameter	Description
Database Name	● The database name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). ● The database name is case insensitive and cannot be left unspecified. ● It cannot exceed 128 characters.
Username	Database owner.
Tables	Number of tables in the database.
Description	Description of the database specified during database creation. If no description is provided, **--** is displayed.
Enterprise Project	Enterprise project to which the database belongs. An enterprise project facilitates project-level management and grouping of cloud resources and users. For details about how to set enterprise projects, see .
Operation	● **Permissions**: View the permission information and perform user authorization, permission settings, and user permission revocation. ● **Tables**: View the tables in the corresponding database. For details, see **Table Management Page**. ● **Create Table**: This permission allows you to create a table in the corresponding database. ● **Drop Database**: This permission allows you to delete the selected database. ● **Modify Database**. This permission allows you to change the owner of the database. The username must exist under the same account.

Table Management Page

From the **Data Management** page, click the database name or **Tables** in the **Operation** column to switch to the **Table Management** page.

The displayed page lists all tables created in the current database. You can view the table type, data storage location, and other information. Tables are listed in chronological order by default, with the most recently created tables displayed at the top.

Table 7-2 Table management parameters

Parameter	Description
Table Name	● The table name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). ● The table name is case insensitive and cannot be left unspecified. ● The table name can contain the dollar sign ($). An example value is **$test**. ● It cannot exceed 128 characters.
Table Type	Table type. Available options are as follows: ● **Managed**: Indicates that data is stored in a DLI table. ● **External**: Indicates that data is stored in an OBS table. ● **View**: Indicates the view type. You can only create views using SQL statements. **NOTE** The table or view information contained in the view cannot be modified. If the table or view information is modified, the query may fail.
Owner	User who creates the table.
Storage Location	DLI, OBS, View, CloudTable, and CSS data location
Size	Size of the data in the table. The value is displayed only for tables of the **Managed** type. For tables of other types, **--** is displayed.
Data Source Path	● If **Data Location** is **OBS**, the corresponding OBS path is displayed. ● If **Data Location** is **DLI** and **View**, **--** is displayed. ● When the data storage location is a datasource connection service such as CloudTable and CSS, the corresponding URL is displayed.
Created	Time when the table is created.
Last Accessed	Last time when an operation was performed on the table.

Parameter	Description
Operation	● **Manage Permissions**: This operation allows you to view the permission information and perform user authorization, permission settings, and user permission revocation. ● **More**: – **Delete**: Delete a table from the corresponding database. – **Modify Owner**: Change the owner of a table The username must exist under the same account. – **Import**: Import data stored in an OBS bucket to a DLI or OBS table. – **Export**: Export data from a DLI table to an OBS bucket. – **Properties**: View data in **Metadata** and **Preview** tabs.

7.1.2 Database Permission Management

Operation Scenarios

- You can isolate databases allocated to different users by setting permissions to ensure data query performance.
- The administrator and database owner have all permissions, which cannot be set or modified by other users.

Precautions

- Lower-level objects automatically inherit permissions granted to upper-level objects. The hierarchical relationship is database > table > column.
- The database owner, table owner, and **authorized** users can assign permissions on the database and tables.
- Columns can only inherit the query permission. For details about **Inheritable Permissions**, see **Database Permission Management**.
- The permissions can be revoked only at the initial level to which the permissions are granted. You need to grant and revoke permissions at the same level. You need to grant and revoke permissions at the same level. For example, after you are granted the insertion permission on a database, you can obtain the insertion permission on the tables in the database. Your insertion permission can be revoked only at the database level.
- If you create a database with the same name as a deleted database, the database permissions will not be inherited. In this case, you need to grant the database permissions to users or projects.

  For example, user A is granted with the permission to delete the **testdb** database. Delete the database and create another one with the same name. You need to grant user A the deletion permission of the **testdb** database again.

Viewing Database Permissions

1. On the left of the management console, choose **Data Management** > **Databases and Tables**.
2. Locate the row where the target database resides and click **Manage Permissions** in the **Operation** column.

   Permissions can be granted to new users or projects, modified for users or projects with existing permissions, or revoked from a user or project.

Granting Permissions to a New User or Project

Here, the new user or project refers to a user or a project that does not have permissions on the database.

1. Click a database you need. In the displayed **Database Permission Management** page, click **Grant Permission** in the upper right corner.
2. In the displayed dialog box, select **User** or **Project**, enter the username or select the project to be authorized, and select the required permissions. For details about the permissions, see **Table 7-3**.

Table 7-3 Parameters

Parameter	Description
Authorization Object	Select **User** or **Project**.
Username/ Project Name	● If you select **User**, enter the IAM username when adding a user to the database. **NOTE** The username is an existing IAM user name and has logged in to the DLI management console. ● If you select **Project**, select the project to be authorized in the current region. **NOTE** When **Project** is selected: ● If you set **Non-inheritable Permissions**, you cannot view tables in the corresponding database within the project. ● If you set **Inheritable Permissions**, you can view all tables in the database within the project.

Parameter	Description
Non-Inherited Permissions	Select a permission to grant it to the user, or deselect a permission to revoke it. Non-inherited permissions apply only to the current database. ● The following permissions are applicable to both user and project authorization: – **Drop Database**: This permission allows you to delete the current database. – **Create Table**: This permission allows you to create tables in the current database. – **Create View**: This permission allows you to create views in the current database. – **Execute SQL EXPLAIN**: This permission allows you to execute an EXPLAIN statement and view information about how this database executes a query. – **Create Role**: This permission allows you to create roles in the current database. – **Delete Role**: This permission allows you to delete roles of the current database. – **View Role**: This permission allows you to view the role of the current user. – **Bind Role**: This permission allows you to bind roles to the current database. – **Unbind Role**: This permission allows you to bind roles from the current database. – **View All Binding Relationships**: This permission allows you to view the binding relationships between all roles and users. – **Create Function**: This permission allows you to create a function in the current database. – **Delete Function**: This permission allows you to delete functions from the current database. – **View All Functions**: This permission allows you to view all functions in the current database. – **View Function Details**: This permission allows you to view details about the current function. ● The following permissions can only be granted to users: – **View All Tables**: This permission allows you to view all tables in the current database. **NOTE** If this permission of a specific database is not granted, all tables in the database will not be displayed. – **View Database**: This permission allows you to view the information about the current database.

Parameter	Description
	NOTE If this permission is not granted, the database will not be displayed.

Parameter	Description
Inherited Permissions	Select a permission to grant it to the user, or deselect a permission to revoke it. Inherited permissions are applicable to the current database and all its tables. However, only the query permission is applicable to table columns. The following permissions can be granted to both user and project. ● **Drop Table**: This permission allows you to delete tables in a database. ● **Select Table**: This permission allows you to query data of the current table. ● **View Table Information**: This permission allows you to view information about the current table. ● **Insert**: This permission allows you to insert data into the current table. ● **Add Column**: This permission allows you to add columns to the current table. ● **Overwrite**: This permission allows you to insert data to overwrite the data in the current table. ● **Grant Permission**: This permission allows you to grant database permissions to other users or projects. ● **Revoke Permission**: This permission allows you to revoke the permissions of the database that other users have but cannot revoke the database owner's permissions. ● **Add Partition to Partition Table**: This permission allows you to add a partition to a partition table. ● **Delete Partition from Partition Table**: This permission allows you to delete existing partitions from a partition table. ● **Configure Path for Partition**: This permission allows you to set the path of a partition in a partition table to a specified OBS path. ● **Rename Table Partition**: This permission allows you to rename partitions in a partition table. ● **Rename Table**: This permission allows you to rename tables. ● **Restore Table Partition**: This permission allows you to export partition information from the file system and save the information to metadata. ● **View All Partitions**: This permission allows you to view all partitions in a partition table. ● **View Other Users' Permissions**: This permission allows you to query other users' permission on the current database.

3. Click **OK**.

Modifying Permissions for an Existing User or Project

For a user or project that has some permissions on the database, you can revoke the existing permissions or grant new ones.

> **NOTE**
>
> If the options in **Set Permission** are gray, the corresponding account does not have the permission to modify the database. You can apply to the administrator, database owner, or other authorized users for granting and revoking permissions of databases.

1. In the **User Permission Info** list, find the user whose permission needs to be set.
   - If the user is a sub-user, you can set permissions for it.
   - If the user is already an administrator, you can only view the permissions information.

   In the **Project Permission Info** list, locate the project for which you want to set permissions and click **Set Permission**.
2. In the **Operation** column of the sub-user or project, click **Set Permission**. The **Set Permission** dialog box is displayed.

   For details about the permissions of database users or projects, see **Table 7-3**.
3. Click **OK**.

Revoking All Permissions of a User or Project

Revoke all permissions of a user or a project.

- In the user list under **User Permission Info**, locate the row where the target sub-user resides and click **Revoke Permission** in the **Operation** column. In the displayed dialog box, click **OK**. In this case, the user has no permissions on the database.

  > **NOTE**
  >
  > If a user is an administrator, **Revoke Permission** is gray, indicating that the user's permission cannot be revoked.

- In the **Project Permission Info** area, select the project whose permissions need to be revoked and click **Revoke Permission** in the **Operation** column. After you click **OK**, the project does not have any permissions on the database.

7.1.3 Table Permission Management

Operation Scenario

- You can isolate databases allocated to different users by setting permissions to ensure data query performance.
- The administrator and database owner have all permissions, which cannot be set or modified by other users.

- When setting database permissions for a new user, ensure that the user group to which the user belongs has the **Tenant Guest** permission.

Precautions

- If you create a table with the same name as a deleted table, the table permissions will not be inherited. In this case, you need to grant the table permissions to users or projects.

  For example, user A is granted with the permission to delete the **testTable** table. Delete the table and create another one with the same name. You need to grant user A the deletion permission of the **testTable** table again.

Viewing Table Permissions

1. On the left of the management console, choose **Data Management** > **Databases and Tables**.
2. Click the database name in the table whose authority is to be set. The **Table Management** page of the database is displayed.
3. Locate the row where the target table resides and click **Manage Permissions** in the **Operation** column.

   Permissions can be granted to new users or projects, modified for users or projects with existing permissions, or revoked from a user or project.

Granting Permissions to a New User or a Project

Here, the new user or project refers to a user or a project that does not have permissions on the database.

1. Click the table you need. In the displayed table permissions page, click **Grant Permission** in the upper right corner.
2. In the displayed **Grant Permission** dialog box, select the required permissions.
   - For details about the DLI table permissions, see **Table 7-4**.

Table 7-4 Parameter description

Parameter	Description
Authorization Object	Select **User** or **Project**.
Username/ Project	• If you select **User**, enter the IAM username when granting table permissions to the user. NOTE The username is an existing IAM user name and has logged in to the DLI management console. • If you select **Project**, select the project to be authorized in the current region. NOTE If you select **Project**, you can only view information about the authorized tables and their databases.

Parameter	Description
Non-inheritable Permissions	Select a permission to grant it to the user, or deselect a permission to revoke it. ● The following permissions are applicable to both user and project authorization: – **Select Table**: This permission allows you to query data of the current table. – **View Table Information**: This permission allows you to view information about the current table. – **View Table Creation Statement**: This permission allows you to view the statement for creating the current table. – **Drop Table**: This permission allows you to delete the current table. – **Rename Table**: Rename the current table. – **Insert**: This permission allows you to insert data into the current table. – **Overwrite**: This permission allows you to insert data to overwrite the data in the current table. – **Add Column**: Add columns to the current table. – **Grant Permission**: The current user can grant table permissions to other users. – **Revoke Permission**: The current user can revoke the table's permissions that other users have but cannot revoke the table owner's permissions. – **View Other Users' Permissions**: This permission allows you to query other users' permission on the current table. The partition table also has the following permissions: – **Delete Partition**: This permission allows you to delete existing partitions from a partition table. – **View All Partitions**: This permission allows you to view all partitions in a partition table. ● The following permissions can only be granted to users: – **View Table**: This permission allows you to display the current table.

– For details about the OBS table permissions, see **Table 7-5**.

Table 7-5 Parameter description

Parameter	Description
Authorization Object	Select **User** or **Project**.
Username/ Project	● If you select **User**, enter the IAM username when granting table permissions to the user. **NOTE** The username is an existing IAM user name and has logged in to the DLI management console. ● If you select **Project**, select the project to be authorized in the current region. **NOTE** If you select **Project**, you can only view information about the authorized tables and their databases.

Parameter	Description
Non-inheritable Permissions	Select a permission to grant it to the user, or deselect a permission to revoke it. ● The following permissions are applicable to both user and project authorization: – **View Table Creation Statement**: This permission allows you to view the statement for creating the current table. – **View Table Information**: This permission allows you to view information about the current table. – **Select Table**: This permission allows you to query data of the current table. – **Drop Table**: This permission allows you to delete the current table. – **Rename Table**: Rename the current table. – **Insert**: This permission allows you to insert data into the current table. – **Overwrite**: This permission allows you to insert data to overwrite the data in the current table. – **Add Column**: This permission allows you to add columns to the current table. – **Grant Permission**: This permission allows you to grant table permissions to other users or projects. – **Revoke Permission**: This permission allows you to revoke the table's permissions that other users or projects have but cannot revoke the table owner's permissions. – **View Other Users' Permissions**: This permission allows you to query other users' permission on the current table. The partition table also has the following permissions: – **Add Partition**: This permission allows you to add a partition to a partition table. – **Delete Partition**: This permission allows you to delete existing partitions from a partition table. – **Configure Path for Partition**: This permission allows you to set the path of a partition in a partition table to a specified OBS path. – **Rename Table Partition**: This permission allows you to rename partitions in a partition table. – **Restore Table Partition**: This permission allows you to export partition information from the file system and save the information to metadata. – **View All Partitions**: This permission allows you to view all partitions in a partition table.

Parameter	Description
	● The following permissions can only be granted to users: – **View Table**: This permission allows you to view the current table.

– For details about the view permissions, see **Table 7-6**.

NOTE

A view can be created only by using SQL statements. You cannot create a view on the **Create Table** page.

Table 7-6 Parameter description

Parameter	Description
Authorization Object	Select **User** or **Project**.
Username/ Project	● If you select **User**, enter the IAM username when adding a user to the database. **NOTE** The username is an existing IAM user name and has logged in to the DLI management console. ● If you select **Project**, select the project to be authorized in the current region. **NOTE** If you select **Project**, you can only view information about the authorized tables and their databases.

Parameter	Description
Non-inheritable Permissions	Select a permission to grant it to the user, or deselect a permission to revoke it. ● The following permissions are applicable to both user and project authorization: – **View Table Information**: This permission allows you to view information about the current table. – **View Table Creation Statement**: This permission allows you to view the statement for creating the current table. – **Drop Table**: This permission allows you to delete the current table. – **Select Table**: This permission allows you to query data of the current table. – **Rename Table**: Rename the current table. – **Grant Permission**: The current user or project can grant table permissions to other users or projects. – **Revoke Permission**: The current user or project can revoke the table's permissions that other users or projects have but cannot revoke the table owner's permissions. – **View Other Users' Permissions**: This permission allows you to query other users' permission on the current table. ● Only applicable to – **View Table**: This permission allows you to view the current table.

3. Click **OK**.

Modifying Permissions for an Existing User or Project

For a user or project that has some permissions on the database, you can revoke the existing permissions or grant new ones.

NOTE

If all options under **Set Permission** are gray, you are not allowed to change permissions on this table. You can apply to the administrator, table owner, or other authorized users for granting and revoking table permissions.

1. In the **User Permission Info** list, find the user whose permission needs to be set.
   - If the user is a sub-user and is not the owner of the table, you can set permissions.
   - If the user is an administrator or table owner, you can only view permissions.

In the **Project Permission Info** list, locate the project for which you want to set permissions and click **Set Permission**.

2. In the **Operation** column of the sub-user or project, click **Set Permission**. The **Set Permission** dialog box is displayed.
   - For details about DLI table user or project permissions, see **Table 7-4**.
   - For details about OBS table user or project permissions, see **Table 7-5**.
   - For details about View table user or project permissions, see **Table 7-6**.
3. Click **OK**.

Revoking All Permissions of a User or Project

Revoke all permissions of a user or a project.

- In the user list under **User Permission Info**, locate the row where the target sub-user resides and click **Revoke Permission** in the **Operation** column. In the displayed dialog box, click **OK**. In this case, the user has no permissions on the table.

  NOTE

  In the following cases, **Revoke Permission** is gray, indicating that the permission of the user cannot be revoked.

  - The user is an administrator.
  - The sub-user is the owner of the table.
  - The sub-user has only inheritable permissions.
- In the **Project Permission Info** area, select the project whose permissions need to be revoked and click **Revoke Permission** in the **Operation** column. After you click **OK**, the project does not have any permissions on the table.

  NOTE

  If a project has only inheritable permissions, **Revoke Permission** is gray, indicating that the permissions of the project cannot be revoked.

7.1.4 Creating a Database or a Table

Definition of Database and Table in DLI

A database, built on the computer storage device, is a data warehouse where data is organized, stored, and managed based on its structure.

The table is an important part of the database. It consists of rows and columns. Each column functions as a field. Each value in a field (column) represents a type of data.

The database is a framework and the table contains data content. A database has one or more tables.

The basic database concept and usage in DLI are similar to those in the Oracle database. DLI permissions are granted on a per database basis.

You can create databases and tables on the management console or using SQL statements. This section describes how to create a database and a table on the management console.

NOTE

A view can be created only by using SQL statements. You cannot create a view on the **Create Table** page.

Precautions

- If a folder and a file have the same name in the OBS directory, the file path is preferred as the path of the OBS table to be created.

Creating a Database

Step 1 You can create a database on either the **Data Management** page or the **SQL Editor** page.

- To create a database on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. In the upper right corner of the **Databases and Tables** page, click **Create Database** to create a database.
- To create a database on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation pane on the left, click ⊕ beside **Databases**.

Step 2 In the displayed **Create Database** dialog box, specify **Name** and **Description** by referring to **Table 7-7**.

Table 7-7 Description

Parameter	Description	Example
Database Name	● The database name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). ● The database name is case insensitive and cannot be left blank. ● The length of the database name cannot exceed 128 characters. **NOTE** The **default** database is a built-in database. You cannot create the database named **default**.	DB01
Description	Description of a database.	-

Step 3 Click **OK**.

After a database is created, you can view and select the database for use on the **Databases and Tables** page or **SQL Editor** page.

----End

Creating a Table

Before creating a table, ensure that a database has been created.

Step 1 You can create a table on either the **Databases and Tables** page or the **SQL Editor** page.

NOTE

Datasource connection tables, such as View tables, HBase (MRS) tables, OpenTSDB (MRS) tables, DWS tables, RDS tables, and CSS tables, cannot be created. You can use SQL to create views and datasource connection tables. For details, see sections **Creating a View** and **Creating a Datasource Connection Table** in the *Data Lake Insight SQL Syntax Reference*.

- To create a table on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. On the **Databases and Tables** page, select the database for which you want to create a table. In the **Operation** column, click **More** > **Create Table** to create a table in the current database.
- To create a table on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation pane of the displayed **SQL Editor** page, click **Databases**. You can create a table in either of the following ways:
     - Click a database name. In the **Tables** area, click ⊕ on the right to create a table in the current database.
     - Click ≡ on the right of the database and choose **Create Table** from the shortcut menu to create a table in the current database.

Step 2 In the displayed **Create Table** dialog box, set parameters as required.

- If you set **Data Location** to **DLI**, set related parameters by referring to **Table 7-8**.
- If you set **Data Location** to **OBS**, set related parameters by referring to **Table 7-8** and **Table 7-9**.

Table 7-8 Common parameters

Parameter	Description	Example
Table Name	– The table name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). – The table name is case insensitive and cannot be left unspecified. – The table name can contain the dollar sign ($). An example value is **$test**. – The length of the table name cannot exceed 128 characters.	table01

Parameter	Description	Example
Data Location	Data storage location. Currently, DLI and OBS are supported.	OBS
Description	Description of the table.	-
Column Type	Available values: **Normal** or **Partition**	Normal
Column	Name of a column in a table. The column name must contain at least one letter and can contain underscores (_). It cannot contain only digits. You can select **Normal** or **Partition**. Partition columns are dedicated to partition tables. User data is partitioned to improve query efficiency. **NOTE** The column name is case-insensitive and must be unique.	name
Type	Data type of a column. This parameter corresponds to **Column Name**. – **string**: The data is of the string type. – **int**: Each integer is stored on four bytes. – **date**: The value ranges from 0000-01-01 to 9999-12-31. – **double**: Each number is stored on eight bytes. – **boolean**: Each value is stored on one byte. – **decimal**: The valid bits are positive integers between 1 to 38, including 1 and 38. The decimal digits are integers less than 10. – **smallint/short**: The number is stored on two bytes. – **bigint/long**: The number is stored on eight bytes. – **timestamp**: The data indicates a date and time. The value can be accurate to six decimal points. – **float**: Each number is stored on four bytes. – **tinyint**: Each number is stored on one byte. Only OBS tables support this data type.	string
Column Description	Description of a column.	-

Parameter	Description	Example
Operation	– Add Column – Delete **NOTE** If the table to be created includes a great number of columns, you are advised to use SQL statements to create the table or import column information from the local EXCEL file.	-

Table 7-9 Parameter description when **Data Location** is set to **OBS**

Parameter	Description	Example
Data Format	DLI supports the following data formats: – **Parquet**: DLI can read non-compressed data or data that is compressed using Snappy and gzip. – **CSV**: DLI can read non-compressed data or data that is compressed using gzip. – **ORC**: DLI can read non-compressed data or data that is compressed using Snappy. – **JSON**: DLI can read non-compressed data or data that is compressed using gzip. – **Avro**: DLI can read uncompressed Avro data.	CSV
Storage Path	Enter or select an OBS path. The path can be a folder or a path. **NOTE** If you need to import data stored in OBS to the OBS table, set this parameter to the path of a folder. If the table creation path is a file, data fails to be imported.	obs://obs1/sampledata.csv

Paramete r	Description	Example
Table Header: No/Yes	This parameter is valid only when **Data Format** is set to **CSV**. Whether the data source to be imported contains the table header. Click **Advanced Settings** and select the check box next to **Table Header: No**. If the check box is selected, the table header is displayed. If the check box is deselected, no table header is displayed.	-
User-defined Delimiter	This parameter is valid only when **Data Format** is set to **CSV** and you select **User-defined Delimiter**. The following delimiters are supported: – Comma (,) – Vertical bar (|) – Tab character (\t) – Others: Enter a user-defined delimiter.	Comma (,)
User-defined Quotation Character	This parameter is valid only when **Data Format** is set to **CSV** and you select **User-defined Quotation Character**. The following quotation characters are supported: – Single quotation mark (') – Double quotation marks (") – Others: Enter a user-defined quotation character.	Single quotation mark (')
User-defined Escape Character	This parameter is valid only when **Data Format** is set to **CSV** and you select **User-defined Escape Character**. The following escape characters are supported: – Backslash (\) – Others: Enter a user-defined escape character.	Backslash (\)

Parameter	Description	Example
Date Format	This parameter is valid only when **Data Format** is set to **CSV** or **JSON**. This parameter specifies the format of the date in the table and is valid only **Advanced Settings** is selected. The default value is **yyyy-MM-dd**. For definition of characters involved in the date pattern, see Table 3 in the	2000-01-01
Timestamp Format	This parameter is valid only when **Data Format** is set to **CSV** or **JSON**. This parameter specifies the format of the timestamp in the table and is valid only **Advanced Settings** is selected. The default value is **yyyy-MM-dd HH:mm:ss**. For definition of characters involved in the time pattern, see Table 3 in the	2000-01-01 09:00:00

Step 3 Click **OK**.

After a table is created, you can view and select the table for use on the **Data Management** page or **SQL Editor** page.

Step 4 (Optional) After a DLI table is created, you can decide whether to directly import data to the table.

----End

7.1.5 Deleting a Database or a Table

You can delete unnecessary databases and tables based on actual conditions.

Precautions

- You are not allowed to delete databases or tables that are being used for running jobs.
- The administrator, database owner, and users with the database deletion permission can delete the database. The administrator, database owner, and users with the table deletion permission can delete the table.

NOTE

If a database or table is deleted, it cannot be recovered. Exercise caution when performing this operation.

Deleting a Table

You can delete a table on either the **Data Management** page or the **SQL Editor** page.

- Delete the table on the **Data Management** page.
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. Locate the row where the database whose tables you want to delete, click the database name to switch to the **Table Management** page.
  c. Locate the row where the target table locates and click **More** > **Delete** in the **Operation** column.
  d. In the displayed dialog box, click **Yes**.
- Delete a table on the **SQL Editor** page.
  a. On the top menu bar of the DLI management console, click **SQL Editor**.
  b. In the navigation tree on the left, click **Databases**. Click the name of a database where the table you want belongs. The tables of the selected database are displayed.
  c. Click ≡ on the right of the table and choose **Delete** from the shortcut menu.
  d. In the dialog box that is displayed, click **OK**.

Deleting a Database

1. On the left of the management console, choose **Data Management** > **Databases and Tables**.
2. Locate the row where the target database locates and click **More** > **Drop Database** in the **Operation** column.

   NOTE

   You cannot delete databases that contain tables. To delete a database containing tables, delete the tables first.
3. In the displayed dialog box, click **Yes**.

7.1.6 Modifying the Owners of Databases and Tables

During actual use, developers create databases and tables and submit them to test personnel for testing. After the test is complete, the databases and tables are transferred to O&M personnel for user experience. In this case, you can change the owner of the databases and tables to transfer data to other owners.

Modifying the Database Owner

You can change the owner of a database on either the **Data Management** page or the **SQL Editor** page.

- On the **Data Management** page, change the database owner.
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. On the **Databases and Tables** page, locate the database you want and click **More** > **Modify Database** in the **Operation** column.
  c. In the displayed dialog box, enter a new owner name (an existing username) and click **OK**.

- Change the database owner on the **SQL Editor** page.
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation tree on the left, click **Databases**, click ≡ on the right of the database you want to modify, and choose **Modify Database** from the shortcut menu.
  c. In the displayed dialog box, enter a new owner name (an existing username) and click **OK**.

Modifying the Table Owner

1. On the left of the management console, choose **Data Management** > **Databases and Tables**.
2. Click the name of the database corresponding to the table to be modified. The **Manage Tables** page of the database is displayed.
3. In the **Operation** column of the target table, choose **More** > **Modify Owner**.
4. In the displayed dialog box, enter a new owner name (an existing username) and click **OK**.

7.1.7 Importing Data to the Table

You can import data from OBS to a table created in DLI.

Precautions

- Only one path can be specified during data import. The path cannot contain commas (,).
- To import data in CSV format to a partitioned table, place the column to be partitioned in the last column of the data source.
- You are advised not to concurrently import data in to a table. If you concurrently import data into a table, there is a possibility that conflicts occur, leading to failed data import.
- The imported file can be in CSV, Parquet, ORC, JSON, and Avro format. The encoding format must be UTF-8.

Prerequisites

The data to be imported has been stored on OBS.

Procedure

Step 1 You can import data on either the **Data Management** page or the **SQL Editor** page.

- To import data on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. Click the name of the database corresponding to the table where data is to be imported to switch to the table management page.
  c. Locate the row where the target table resides and choose **More** > **Import** in the **Operation** column. The **Import** dialog box is displayed.

- To import data on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation tree on the left of **SQL Editor**, click **Databases** to see all databases. Click the database where the target table belongs. The table list is displayed.
  c. Click ≡ on the right of the table and choose **Import** from the shortcut menu. The **Import** page is displayed.

Step 2 In the **Import** dialog box, set the parameters based on **Table 7-10**.

Table 7-10 Description

Parameter	Description	Example
Databases	Database where the current table is located.	-
Table Name	Name of the current table.	-
Queues	Queue where the imported data will be used	-
File Format	Format of the data source file to be imported. The CSV, Parquet, ORC, JSON, and Avro formats are supported. Encoding format. Only UTF-8 is supported.	CSV
Path	You can directly enter a path or click 📂 and select an OBS path. If no bucket is available, you can directly switch to the OBS management console and create an OBS bucket. ● **When creating an OBS table, you must specify a folder as the directory. If a file is specified, data import may be failed.** ● **If a folder and a file have the same name in the OBS directory, the file path is preferred as the path of the data to be imported.** **NOTE** The path can be a file or folder.	obs://DLI/sampledata.csv
Table Header: No/Yes	This parameter is valid only when **File Format** is set to **CSV**. Whether the data source to be imported contains the table header. Click **Advanced Settings** and select the check box next to **Table Header: No**. If the check box is selected, the table header is displayed. If the check box is deselected, no table header is displayed.	-

Parameter	Description	Example
User-defined Delimiter	This parameter is valid only when **File Format** is set to **CSV** and you select **User-defined Delimiter**. The following delimiters are supported: ● Comma (,) ● Vertical bar (|) ● Tab character (\t) ● Others: Enter a user-defined delimiter.	Default value: comma (,)
User-defined Quotation Character	This parameter is valid only when **File Format** is set to **CSV** and **User-defined Quotation Character** is selected. The following quotation characters are supported: ● Single quotation mark (') ● Double quotation marks (") ● Others: Enter a user-defined quotation character.	Default value: double quotation marks (")
User-defined Escape Character	This parameter is valid only when **File Format** is set to **CSV** and you select **User-defined Escape Character**. The following escape characters are supported: ● Backslash (\) ● Others: Enter a user-defined escape character.	Default value: backslash (\)
Date Format	This parameter is valid only when **File Format** is set to **CSV** or **JSON**. This parameter specifies the format of the date in the table and is valid only **Advanced Settings** is selected. The default value is **yyyy-MM-dd**. For definition of characters involved in the date pattern, see Table 3 in the	2000-01-01
Timestamp Format	This parameter is valid only when **File Format** is set to **CSV** or **JSON**. This parameter specifies the format of the timestamp in the table and is valid only **Advanced Settings** is selected. The default value is **yyyy-MM-dd HH:mm:ss**. For definition of characters involved in the time pattern, see Table 3 in the	2000-01-01 09:00:00
Error Records Path	This parameter is valid only when **File Format** is set to **CSV** or **JSON**. The parameter specifies the error data is stored in the corresponding OBS path and is valid only **Advanced Settings** is selected.	obs://DLI/

Step 3 Click **OK**.

Step 4 You can view the imported data in either of the following ways:

NOTE

Currently, only the first 10 records are displayed.

- Choose **Data Management** > **Databases and Tables** in the navigation pane of the console. Locate the row that contains the database where the target table belongs and click **More** > **View Properties** in the **Operation** column. In the displayed dialog box, click the **Preview** tab to view the imported data.
- In the **Databases** tab of the **SQL Editor**, click the database name to go to the table list. Click ≡ on the right of a table name and choose **View Properties** from the shortcut menu. In the displayed dialog box, click **Preview** to view the imported data.

Step 5 (Optional) View the status and execution result of the importing job on the **Job Management** > **SQL Jobs** page.

----End

7.1.8 Exporting Data from DLI to OBS

You can export data from a DLI table to OBS. During the export, a folder is created in OBS or the content in the existing folder is overwritten.

Precautions

- The exported file can be in JSON format, and the text format can only be UTF-8.
- Only the data in the DLI table (the table type is **Managed**) can be exported to the OBS bucket, and the export path must be specified to the folder level.
- Data can be exported across accounts. That is, after account B authorizes account A, account A can export data to the OBS path of account B if account A has the permission to read the metadata and permission information about the OBS bucket of account B and read and write the path.

Procedure

Step 1 You can export data on either the **Data Management** page or the **SQL Editor** page.

- To export data on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. Click the name of the database corresponding to the table where data is to be exported to switch to the **Manage Tables** page.
  c. Select the corresponding table (DLI table) and choose **More** > **Export** in the **Operation** column. The **Export Data** page is displayed.
- To export data on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.

b. In the navigation tree on the left, click **Databases** to see all databases. Click the database name corresponding to the table to which data is to be exported. The tables are displayed.

c. Click ≡ on the right of the managed table (DLI table) whose data is to be exported, and choose **Export** from the shortcut menu.

Step 2 In the displayed **Export Data** dialog box, specify parameters by referring to **Table 7-11**.

Table 7-11 Parameter description

Parameter	Description
Databases	Database where the current table is located.
Table Name	Name of the current table.
Data Format	Format of the file storing data to be exported. Formats other than JSON will be supported in later versions.
Queue	Select a queue.
Compression Format	Compression format of the data to be exported. The following compression formats are supported: ● none ● bzip2 ● deflate ● gzip
Storage Path	● Enter or select an OBS path. ● The export path must be a folder that does not exist in the OBS bucket. Specifically, you need to create a folder in the target OBS directory. ● The folder name cannot contain the special characters of **\/:*?"<>|**, and cannot start or end with a dot (.).
Export Mode	Storage mode of the data to be exported. ● **New OBS directory**: If the specified export directory exists, an error is reported and the export operation cannot be performed. ● **Existing OBS directory (Overwritten)**: If you create a file in the specified directory, the existing file will be overwritten.
Table Header: No/Yes	Whether the data to be exported contains the table header.

Step 3 Click **OK**.

Step 4 (Optional) You can view the job status (indicated by **Status**), statements (indicated by **Statement**), and other information about exporting jobs on the **SQL Jobs** page.

1. Select **EXPORT** from the **Job Type** drop-down list box and specify the time range for exporting data. The jobs meeting the requirements are displayed in the job list.
2. Click ˅ to view details about an exporting job.

----End

7.1.9 Viewing Metadata

Metadata Description

- Metadata is used to define data types. It describes information about the data, including the source, size, format, and other data features. In database fields, metadata interprets data content in the data warehouse.
- When you create a table, metadata is defined, consisting of the column name, type, and description.
- The **Metadata** page displays information about the target table, including **Column Name**, **Column Type**, **Data Type**, and **Description**.

Procedure

You can view metadata on either the **Data Management** page or the **SQL Editor** page.

- To view metadata on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. On the displayed **Data Management** page, click the name of the database where the target table whose data you want to export resides to switch to the **Manage Tables** page.
  c. Click **More** in the **Operation** column of the target table and select **View Properties**. In the **Metadata** tab, view the metadata of the table.
- To view metadata on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation pane of the displayed **SQL Editor** page, click **Databases**.
  c. Click the corresponding database name to view the tables in the database.
  d. Click ≡ on the right of the table and choose **View Properties** from the shortcut menu. On the **Metadata** tab page, view the metadata of the table.

7.1.10 Previewing Data

The **Preview** page displays the first 10 records in the table.

Procedure

You can preview data on either the **Data Management** page or the **SQL Editor** page.

- To preview data on the **Data Management** page:
  a. On the left of the management console, choose **Data Management** > **Databases and Tables**.
  b. On the displayed **Data Management** page, click the name of the database where the target table whose data you want to export resides to switch to the **Manage Tables** page.
  c. Click **More** in the **Operation** column of the target table and select **View Properties**.
  d. Click the **Preview** tab to preview the table data.
- To preview data on the **SQL Editor** page:
  a. On the left of the management console, click **SQL Editor**.
  b. In the navigation pane of the displayed **SQL Editor** page, click **Databases**.
  c. Click the corresponding database name to view the tables in the database.
  d. Click ≡ on the right of the corresponding table, choose **View Properties** from the list menu, and click the **Preview** tab to preview the data of the table.

7.2 Package Management

7.2.1 Overview

Package management provides the following functions:

- **Managing Package Permissions**
- **Creating a Package**
- **Deleting a Package**

  NOTE

  You can delete program packages in batches.
- **Modifying the Owner**

Package Management Page

Table 7-12 Package management parameters

Parameter	Description
Group Name	Name of the group to which the package belongs. If the package is not grouped, **--** is displayed.
Package Name	Name of a package.
Owner	Name of the user who uploads the package.

Parameter	Description
Type	Type of a package. The following package types are supported: ● JAR: JAR file ● PyFile: User Python file ● File: User file
Status	Status of the package to be created. ● Uploading: The file is being uploaded. ● Finished: The resource package has been uploaded. ● Failed: The resource package upload failed.
Created	Time when a package is created.
Updated	Time when the package is updated.
Operation	Manage Permissions: Manage user permissions for a package. Delete: Delete the package. **More**: ● **Modify Owner**: Modify the owner of the package.

7.2.2 Managing Permissions on Packages and Package Groups

Scenarios

- You can isolate package groups or packages allocated to different users by setting permissions to ensure data query performance.
- The administrator and the owner of a package group or package have all permissions. You do not need to set permissions and the permissions cannot be modified by other users.
- When you set permissions on a package group or a package to a new user, the user group to which the user belongs must have the Tenant Guest permission. For details about the Tenant Guest permission and how to apply for the permission, see .

On the **Package Management** page, click **Manage Permissions** in the **Operation** column of the target package. On the displayed **User Permission Info** page, you can grant permissions for the package group or package, set and revoke user permissions.

NOTE

- If you select a group when creating a package, you can manage permissions of the corresponding program package group.
- If you select **No grouping** when creating a package, you can manage permissions of the corresponding package.

Granting Permissions on Package Groups/Packages

Click **Grant Permission** in the upper right corner of the page.

- Granting permissions on package groups

Table 7-13 Permission parameters

Parameter	Description
Username	Name of the authorized IAM user. **NOTE** The username is the name of an existing IAM user.
Select the permissions to be granted to the user	● **Use Group**: This permission allows you to use the package of this group. ● **Update Group**: This permission allows you to update the packages in the group, including creating a package in the group. ● **Query Group**: This permission allows you to query the details of a package in a group. ● **Delete Group**: This permission allows you to delete the package of the group. ● **Grant Permission**: This permission allows you to grant group permissions to other users. ● **Revoke Permission**: This permission allows you to revoke the group permissions that other users have but cannot revoke the group owner's permissions. ● **View Other User's Permissions**: This permission allows you to view the group permissions of other users.

- Granting permissions on packages

Table 7-14 Permission parameters

Parameter	Description
Username	Name of the authorized IAM user. **NOTE** The username is the name of an existing IAM user.

Parameter	Description
Select the permissions to be granted to the user	● **Use Package**: This permission allows you to use the package. ● **Update Package**: This permission allows you to update the package. ● **Query Package**: This permission allows you to query the package. ● **Delete Package**: This permission allows you to delete the package. ● **Grant Permission**: This permission allows you to grant package permissions to other users. ● **Revoke Permission**: This permission allows you to revoke the package permissions that other users have but cannot revoke the package owner's permissions. ● **View Other User's Permissions**: This permission allows you to view the package permissions of other users.

Setting Permissions on Package Groups and Packages

Click **Set Permission** in the **Operation** column of the sub-user to modify the permission of the user. **Table 7-13** and **Table 7-14** list the detailed permission descriptions.

If the **Set Permission** button is gray, you do not have the permission to modify the package group or package. You can apply to the administrator, group owner, or other users who have the permissions on granting and revoking permissions of package groups or packages.

Revoking Permissions on Package Groups and Packages

Click **Revoke Permission** in the **Operation** column of a sub-user to revoke the user's permissions. After the operation, the sub-user does not have any permission on the package group or package.

Permissions Description

- Package group permissions

  Querying permissions. A group owner can view the created package group and all packages in the group, and can also view package groups on which he or she has all permissions.

  A package group is a unit. If you select a group when creating a package, you can grant only the permissions of the package group to other users.

- Package permissions

  Querying permissions. A package owner can view the created packages, and can also view packages on which he or she has all permissions.

7.2.3 Creating a Package

DLI allows you to submit program packages in batches to the general-use queue for running.

NOTE

If you need to update a package, you can use the same package or file to upload it to the same location (in the same group) on DLI to overwrite the original package or file.

Prerequisites

All software packages must be uploaded to OBS for storage in advance.

Procedure

1. On the left of the management console, choose **Data Management** > **Package Management**.
2. On the **Package Management** page, click **Create** in the upper right corner to create a package.
3. In the displayed **Create Package** dialog box, set related parameters by referring to **Table 7-15**.

Table 7-15 Parameter description

Parameter	Description
Package Type	The following package types are supported: ● JAR: JAR file ● PyFile: User Python file ● File: User file
Package File Path	Select the OBS path of the corresponding packages. NOTE ● The packages must be uploaded to OBS for storage in advance. ● Only files can be selected.
Group Policy	You can select **Use existing group**, **Use new group**, or **No grouping**.
Group Name	● **Use existing group**: Select an existing group. ● **Use new group**: Enter a custom group name. ● **No grouping**: No need to select or enter a group name. NOTE ● If you select a group, the permission management refers to the permissions of the corresponding package group. ● If no group is selected, the permission management refers to the permissions of the corresponding package. For details about how to manage permissions on package groups and packages, see **Managing Permissions on Packages and Package Groups**.

4. Click **OK**.

   After a package is created, you can view and select the package for use on the **Package Management** page.

7.2.4 Deleting a Package

You can delete a package based on actual conditions.

Procedure

1. On the left of the management console, choose **Data Management** > **Package Management**.
2. Click **Delete** in the **Operation** column of the package to be deleted.
3. In the dialog box that is displayed, click **Yes**.

7.2.5 Modifying the Owner

To change the owner of a package, click **More** > **Modify Owner** in the **Operation** column of a package on the **Package Management** page.

- If the package has been grouped, you can modify the owner of the **Group** or **Resource** of it.
- If the package has not been grouped, change its owner directly.

Table 7-16 Description

Parameter	Description
Group Name	● If you select a group when creating a package, the name of the group is displayed. ● If no group is selected when creating a package, this parameter is not displayed.
Name	Name of a package.
Select Type	● If you select a group when creating a package, you can change the owner of the group or package. ● If no group is selected when creating a package, this parameter is not displayed.
Username	Name of the package owner. **NOTE** The username is the name of an existing IAM user.

7.2.6 Built-in Dependencies

DLI built-in dependencies are provided by the platform by default. In case of conflicts, you do not need to upload them when packaging JAR packages of Spark or Flink Jar jobs.

Spark 2.3.2 Dependencies

- accessors-smart-1.2.jar
- activation-1.1.1.jar
- aircompressor-0.8.jar
- alluxio-2.3.1-luxor-SNAPSHOT-client.jar
- antlr-2.7.7.jar
- antlr4-runtime-4.8-1.jar
- antlr-runtime-3.4.jar
- aopalliance-1.0.jar
- aopalliance-repackaged-2.4.0-b34.jar
- apache-log4j-extras-1.2.17.jar
- arpack_combined_all-0.1.jar
- arrow-format-0.8.0.jar
- arrow-memory-0.8.0.jar
- arrow-vector-0.8.0.jar
- asm-5.0.4.jar
- audience-annotations-0.5.0.jar
- automaton-1.11-8.jar
- avro-1.7.7.jar
- avro-ipc-1.7.7.jar
- avro-ipc-1.7.7-tests.jar
- avro-mapred-1.7.7-hadoop2.jar
- aws-java-sdk-bundle-1.11.271.jar
- base64-2.3.8.jar
- bcpkix-jdk15on-1.66.jar
- bcprov-jdk15on-1.66.jar
- bonecp-0.8.0.RELEASE.jar
- breeze_2.11-0.13.2.jar
- breeze-macros_2.11-0.13.2.jar
- calcite-avatica-1.2.0-incubating.jar
- calcite-core-1.2.0-incubating.jar
- calcite-linq4j-1.2.0-incubating.jar
- checker-qual-2.11.1.jar
- chill_2.11-0.8.4.jar
- chill-java-0.8.4.jar
- commons-beanutils-1.9.4.jar
- commons-cli-1.2.jar
- commons-codec-2.0-20130428.202122-59.jar
- commons-collections-3.2.2.jar
- commons-collections4-4.2.jar

- commons-compiler-3.0.8.jar
- commons-compress-1.4.1.jar
- commons-configuration2-2.1.1.jar
- commons-crypto-1.0.0-20191105.jar
- commons-daemon-1.0.13.jar
- commons-dbcp-1.4.jar
- commons-dbcp2-2.7.0.jar
- commons-httpclient-3.1.jar
- commons-io-2.5.jar
- commons-lang-2.6.jar
- commons-lang3-3.5.jar
- commons-logging-1.2.jar
- commons-math3-3.4.1.jar
- commons-net-2.2.jar
- commons-pool-1.5.4.jar
- commons-pool2-2.8.0.jar
- commons-text-1.3.jar
- compress-lzf-1.0.3.jar
- core-1.1.2.jar
- curator-client-4.2.0.jar
- curator-framework-4.2.0.jar
- curator-recipes-2.7.1.jar
- datanucleus-api-jdo-3.2.6.jar
- datanucleus-core-3.2.10.jar
- datanucleus-rdbms-3.2.9.jar
- derby-10.12.1.1.jar
- dnsjava-2.1.7.jar
- ehcache-3.3.1.jar
- eigenbase-properties-1.1.5.jar
- error_prone_annotations-2.3.4.jar
- failureaccess-1.0.1.jar
- fastutil-8.2.3.jar
- ffmpeg-4.3.1-1.5.4.jar
- ffmpeg-4.3.1-1.5.4-linux-x86_64.jar
- flatbuffers-1.2.0-3f79e055.jar
- generex-1.0.2.jar
- geronimo-jcache_1.0_spec-1.0-alpha-1.jar
- gson-2.2.4.jar
- guava-29.0-jre.jar
- guice-4.0.jar

- guice-servlet-4.0.jar
- hadoop-annotations-3.1.1-ei-302002.jar
- hadoop-auth-3.1.1-ei-302002.jar
- hadoop-aws-3.1.1-ei-302002.jar
- hadoop-client-3.1.1-ei-302002.jar
- hadoop-common-3.1.1-ei-302002.jar
- hadoop-hdfs-3.1.1-ei-302002.jar
- hadoop-hdfs-client-3.1.1-ei-302002.jar
- hadoop-mapreduce-client-common-3.1.1-ei-302002.jar
- hadoop-mapreduce-client-core-3.1.1-ei-302002.jar
- hadoop-mapreduce-client-jobclient-3.1.1-ei-302002.jar
- hadoop-minikdc-3.1.1-hw-ei-302002.jar
- hadoop-yarn-api-3.1.1-ei-302002.jar
- hadoop-yarn-client-3.1.1-ei-302002.jar
- hadoop-yarn-common-3.1.1-ei-302002.jar
- hadoop-yarn-registry-3.1.1-ei-302002.jar
- hadoop-yarn-server-common-3.1.1-ei-302002.jar
- hadoop-yarn-server-web-proxy-3.1.1-ei-302002.jar
- hamcrest-core-1.3.jar
- HikariCP-java7-2.4.12.jar
- hive-common-1.2.1-2.1.0.dli-20201111.064115-91.jar
- hive-exec-1.2.1-2.1.0.dli-20201111.064444-91.jar
- hive-metastore-1.2.1-2.1.0.dli-20201111.064230-91.jar
- hk2-api-2.4.0-b34.jar
- hk2-locator-2.4.0-b34.jar
- hk2-utils-2.4.0-b34.jar
- hppc-0.7.2.jar
- htrace-core4-4.2.0-incubating-hw-1.0.0.jar
- httpclient-4.5.4.jar
- httpcore-4.4.7.jar
- ivy-2.4.0.jar
- j2objc-annotations-1.3.jar
- jackson-annotations-2.10.0.jar
- jackson-core-2.10.0.jar
- jackson-core-asl-1.9.13-atlassian-4.jar
- jackson-databind-2.10.0.jar
- jackson-dataformat-yaml-2.10.0.jar
- jackson-datatype-jsr310-2.10.3.jar
- jackson-jaxrs-base-2.10.3.jar
- jackson-jaxrs-json-provider-2.10.3.jar

- jackson-mapper-asl-1.9.13-atlassian-4.jar
- jackson-module-jaxb-annotations-2.10.3.jar
- jackson-module-paranamer-2.10.0.jar
- jackson-module-scala_2.11-2.10.0.jar
- jakarta.activation-api-1.2.1.jar
- jakarta.xml.bind-api-2.3.2.jar
- janino-3.0.8.jar
- javacpp-1.5.4.jar
- javacpp-1.5.4-linux-x86_64.jar
- javacv-1.5.4.jar
- JavaEWAH-1.1.7.jar
- javassist-3.18.1-GA.jar
- javax.annotation-api-1.2.jar
- javax.inject-1.jar
- javax.inject-2.4.0-b34.jar
- javax.servlet-api-3.1.0.jar
- javax.ws.rs-api-2.0.1.jar
- java-xmlbuilder-1.1.jar
- javolution-5.3.1.jar
- jaxb-api-2.2.11.jar
- jcip-annotations-1.0-1.jar
- jcl-over-slf4j-1.7.26.jar
- jdo-api-3.0.1.jar
- jersey-client-2.23.1.jar
- jersey-common-2.23.1.jar
- jersey-container-servlet-2.23.1.jar
- jersey-container-servlet-core-2.23.1.jar
- jersey-guava-2.23.1.jar
- jersey-media-jaxb-2.23.1.jar
- jersey-server-2.23.1.jar
- jets3t-0.9.4.jar
- jetty-http-9.4.31.v20200723.jar
- jetty-io-9.4.31.v20200723.jar
- jetty-security-9.4.31.v20200723.jar
- jetty-server-9.4.31.v20200723.jar
- jetty-servlet-9.4.31.v20200723.jar
- jetty-util-9.4.31.v20200723.jar
- jetty-util-ajax-9.4.31.v20200723.jar
- jetty-webapp-9.4.31.v20200723.jar
- jetty-xml-9.4.31.v20200723.jar

- joda-time-2.9.3.jar
- jodd-core-4.2.0.jar
- json-20200518.jar
- json4s-ast_2.11-3.2.11.jar
- json4s-core_2.11-3.2.11.jar
- json4s-jackson_2.11-3.2.11.jar
- json-sanitizer-1.2.1.jar
- json-smart-2.3.jar
- jsp-api-2.1.jar
- jsr305-3.0.2.jar
- jta-1.1.jar
- jtransforms-2.4.0.jar
- jul-to-slf4j-1.7.26.jar
- junit-4.11.jar
- kerb-admin-1.0.1.jar
- kerb-client-1.0.1.jar
- kerb-common-1.0.1.jar
- kerb-core-1.0.1.jar
- kerb-crypto-1.0.1.jar
- kerb-identity-1.0.1.jar
- kerb-server-1.0.1.jar
- kerb-simplekdc-1.0.1.jar
- kerb-util-1.0.1.jar
- kerby-asn1-1.0.1.jar
- kerby-config-1.0.1.jar
- kerby-pkix-1.0.1.jar
- kerby-util-1.0.1.jar
- kerby-xdr-1.0.1.jar
- kryo-shaded-3.0.3.jar
- kubernetes-client-4.9.2-20200804.jar
- kubernetes-model-4.9.2-20200804.jar
- kubernetes-model-common-4.9.2-20200804.jar
- leveldbjni-all-1.8-20191105.jar
- libfb303-0.9.3.jar
- libthrift-0.12.0.jar
- listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar
- log4j-1.2.17-cloudera1.jar
- log4j-rolling-appender-20131024-2017.jar
- logging-interceptor-3.14.4.jar
- luxor-encrypt-2.1.0-20201106.065437-53.jar

- luxor-fs3-2.1.0-20201106.065612-53.jar
- luxor-obs-fs3-2.1.0-20201106.065616-53.jar
- luxor-rpc_2.11-2.1.0-20201106.065541-53.jar
- luxor-rpc-protobuf2-2.1.0-20201106.065551-53.jar
- lz4-java-1.7.1.jar
- machinist_2.11-0.6.1.jar
- macro-compat_2.11-1.1.1.jar
- metrics-core-3.1.5.jar
- metrics-graphite-3.1.5.jar
- metrics-jmx-4.1.12.1.jar
- metrics-json-3.1.5.jar
- metrics-jvm-3.1.5.jar
- minlog-1.3.0.jar
- mssql-jdbc-6.2.1.jre7.jar
- netty-3.10.6.Final.jar
- netty-all-4.1.51.Final.jar
- nimbus-jose-jwt-8.19.jar
- objenesis-2.1.jar
- okhttp-3.14.4.jar
- okio-1.17.2.jar
- opencsv-2.3.jar
- opencsv-4.6.jar
- opencv-4.3.0-2.jar
- orc-core-1.4.4-nohive.jar
- orc-mapreduce-1.4.4-nohive.jar
- oro-2.0.8.jar
- osgi-resource-locator-1.0.1.jar
- paranamer-2.8.jar
- parquet-column-1.8.3.jar
- parquet-common-1.8.3.jar
- parquet-encoding-1.8.3.jar
- parquet-format-2.3.1.jar
- parquet-hadoop-1.8.3.jar
- parquet-hadoop-bundle-1.6.0.jar
- parquet-jackson-1.8.3.jar
- parquet-format-2.3.1.jar
- parquet-hadoop-1.8.3.jar
- parquet-hadoop-bundle-1.6.0.jar
- parquet-jackson-1.8.3.jar
- postgresql-42.2.14.jar

- protobuf-java-2.5.0.jar
- py4j-0.10.7.jar
- pyrolite-4.13.jar
- re2j-1.1.jar
- RoaringBitmap-0.5.11.jar
- scala-compiler-2.11.12.jar
- scala-library-2.11.12.jar
- scalap-2.11.0.jar
- scala-parser-combinators_2.11-1.1.0.jar
- scala-reflect-2.11.12.jar
- scala-xml_2.11-1.0.5.jar
- secComponentApi-1.0.5c.jar
- shapeless_2.11-2.3.2.jar
- slf4j-api-1.7.30.jar
- slf4j-log4j12-1.7.30.jar
- snakeyaml-1.24.jar
- snappy-java-1.1.7.5.jar
- spark-catalyst_2.11-2.3.2.0101-2.1.0.dli-20201111.073826-143.jar
- spark-core_2.11-2.3.2.0101-2.1.0.dli-20201111.073836-134.jar
- spark-graphx_2.11-2.3.2.0101-2.1.0.dli-20201111.073847-129.jar
- spark-hive_2.11-2.3.2.0101-2.1.0.dli-20201111.073854-132.jar
- spark-kubernetes_2.11-2.3.2.0101-2.1.0.dli-20201111.073916-85.jar
- spark-kvstore_2.11-2.3.2.0101-2.1.0.dli-20201111.073933-127.jar
- spark-launcher_2.11-2.3.2.0101-2.1.0.dli-20201111.073940-127.jar
- spark-mllib_2.11-2.3.2.0101-2.1.0.dli-20201111.073946-127.jar
- spark-mllib-local_2.11-2.3.2.0101-2.1.0.dli-20201111.073953-127.jar
- spark-network-common_2.11-2.3.2.0101-2.1.0.dli-20201111.073959-127.jar
- spark-network-shuffle_2.11-2.3.2.0101-2.1.0.dli-20201111.074007-127.jar
- spark-om_2.11-2.3.2.0101-2.1.0.dli-20201111.074019-125.jar
- spark-repl_2.11-2.3.2.0101-2.1.0.dli-20201111.074028-125.jar
- spark-sketch_2.11-2.3.2.01012.1.0.dli-20201111.074035-125.jar
- spark-sql_2.11-2.3.2.0101-2.1.0.dli-20201111.074041-126.jar
- spark-streaming_2.11-2.3.2.0101-2.1.0.dli-20201111.074100-123.jar
- spark-tags_2.11-2.3.2.0101-2.1.0.dli-20201111.074136-123.jar
- spark-tags_2.11-2.3.2.0101-2.1.0.dli-20201111.074141-124-tests.jar
- spark-unsafe_2.11-2.3.2.0101-2.1.0.dli-20201111.074144-123.jar
- spark-uquery_2.11-2.3.2.0101-2.1.0.dli-20201111.074906-210.jar
- spark-yarn_2.11-2.3.2.0101-2.1.0.dli-20201111.074151-123.jar
- spire_2.11-0.13.0.jar
- spire-macros_2.11-0.13.0.jar

- ST4-4.3.1.jar
- stax2-api-3.1.4.jar
- stax-api-1.0-2.jar
- stream-2.7.0.jar
- stringtemplate-3.2.1.jar
- token-provider-1.0.1.jar
- univocity-parsers-2.5.9.jar
- validation-api-1.1.0.Final.jar
- woodstox-core-5.0.3.jar
- xbean-asm5-shaded-4.4.jar
- xercesImpl-2.12.0.jar
- xml-apis-1.4.01.jar
- xz-1.0.jar
- zjsonpatch-0.3.0.jar
- zookeeper-3.5.6-ei-302002.jar
- zookeeper-jute-3.5.6-ei-302002.jar
- zstd-jni-1.4.4-11.jar

Flink 1.7.2 Dependencies

- bcpkix-jdk15on-1.60.jar
- bcprov-jdk15on-1.60.jar
- commons-codec-1.9.jar
- commons-configuration-1.7.jar
- deeplearning4j-core-0.9.1.jar
- deeplearning4j-nlp-0.9.1.jar
- deeplearning4j-nn-0.9.1.jar
- ejml-cdense-0.33.jar
- ejml-core-0.33.jar
- ejml-ddense-0.33.jar
- ejml-dsparse-0.33.jar
- ejml-experimental-0.33.jar
- ejml-fdense-0.33.jar
- ejml-simple-0.33.jar
- ejml-zdense-0.33.jar
- elsa-3.0.0-M7.jar
- esdk-obs-java-3.1.3.jar
- flink-cep_2.11-1.7.0.jar
- flink-cep-scala_2.11-1.7.0.jar
- flink-dist_2.11-1.7.0.jar
- flink-gelly_2.11-1.7.0.jar

- flink-gelly-scala_2.11-1.7.0.jar
- flink-ml_2.11-1.7.0.jar
- flink-python_2.11-1.7.0.jar
- flink-queryable-state-runtime_2.11-1.7.0.jar
- flink-shaded-curator-1.7.0.jar
- flink-shaded-hadoop2-uber-1.7.0.jar
- flink-table_2.11-1.7.0.jar
- guava-26.0-jre.jar
- hadoop-cloud-3.1.1-41-20201014.085840-4.jar
- httpasyncclient-4.1.2.jar
- httpclient-4.5.12.jar
- httpcore-4.4.4.jar
- httpcore-nio-4.4.4.jar
- java-xmlbuilder-1.1.jar
- jna-4.1.0.jar
- libtensorflow-1.12.0.jar
- log4j-over-slf4j-1.7.21.jar
- logback-classic-1.2.3.jar
- logback-core-1.2.3.jar
- nd4j-api-0.9.1.jar
- nd4j-native-0.9.1.jar
- nd4j-native-api-0.9.1.jar
- nd4j-native-platform-0.9.1.jar
- okhttp-3.14.8.jar
- okio-1.14.0.jar
- slf4j-api-1.7.21.jar
- tensorflow-1.12.0.jar

Flink 1.10 Dependencies

Only queues created after December 2020 can use the Flink 1.10 dependencies.

- bcpkix-jdk15on-1.60.jar
- bcprov-jdk15on-1.60.jar
- commons-codec-1.9.jar
- commons-configuration-1.7.jar
- deeplearning4j-core-0.9.1.jar
- deeplearning4j-nlp-0.9.1.jar
- deeplearning4j-nn-0.9.1.jar
- ejml-cdense-0.33.jar
- ejml-core-0.33.jar
- ejml-ddense-0.33.jar

- ejml-dsparse-0.33.jar
- ejml-experimental-0.33.jar
- ejml-fdense-0.33.jar
- ejml-simple-0.33.jar
- ejml-zdense-0.33.jar
- elsa-3.0.0-M7.jar
- esdk-obs-java-3.20.6.1.jar
- flink-cep_2.11-1.10.0.jar
- flink-cep-scala_2.11-1.10.0.jar
- flink-dist_2.11-1.10.0.jar
- flink-python_2.11-1.10.0.jar
- flink-queryable-state-runtime_2.11-1.10.0.jar
- flink-sql-client_2.11-1.10.0.jar
- flink-state-processor-api_2.11-1.10.0.jar
- flink-table_2.11-1.10.0.jar
- flink-table-blink_2.11-1.10.0.jar
- guava-26.0-jre.jar
- hadoop-3.1.1-41.jar
- httpasyncclient-4.1.2.jar
- httpclient-4.5.3.jar
- httpcore-4.4.4.jar
- httpcore-nio-4.4.4.jar
- java-xmlbuilder-1.1.jar
- jna-4.1.0.jar
- libtensorflow-1.12.0.jar
- log4j-over-slf4j-1.7.26.jar
- logback-classic-1.2.3.jar
- logback-core-1.2.3.jar
- nd4j-api-0.9.1.jar
- nd4j-native-0.9.1.jar
- nd4j-native-api-0.9.1.jar
- nd4j-native-platform-0.9.1.jar
- okhttp-3.14.8.jar
- okio-1.14.0.jar
- secComponentApi-1.0.5.jar
- slf4j-api-1.7.26.jar
- tensorflow-1.12.0.jar

8 Job Templates

8.1 SQL Template Management

To facilitate SQL operation execution, DLI allows you to customize query templates or save the SQL statements in use as templates. After templates are saved, you do not need to compile SQL statements. Instead, you can directly perform the SQL operations using the templates.

SQL templates include sample templates and custom templates. The default sample template contains 22 standard TPC-H query statements, which can meet most TPC-H test requirements. For details, see **TPC-H Sample Data in the SQL Template**.

SQL template management provides the following functions:

- **Sample Templates**
- **Custom Templates**
- **Creating a Template**
- **Executing the Template**
- **Searching for a Template**
- **Modifying a Template**
- **Deleting a Template**

Table Settings

In the upper right corner of the **SQL Template** page, click **Set Property** to determine whether to display templates by group.

If you select **Display by Group**, the following display modes are available:

- Expand the first group
- Expand all
- Collapse All

Sample Templates

The current sample template contains 22 standard TPC-H query statements. You can view the template name, description, and statements. For details about TPC-H examples, see **TPC-H Sample Data in the SQL Template**.

Table 8-1 Template management parameters

Parameter	Description
Name	Indicates the template name. ● A template name can contain only digits, letters, and underscores (_), but cannot start with an underscore (_) or contain only digits. It cannot be left empty. ● The template name can contain a maximum of 50 characters.
Description	Description of the template you create.
Statement	SQL statement created as the template.
Operation	**Execute**: After you click this button, the system switches to the **SQL Editor** page, where you can modify or directly perform the statement as required. For details, see **Executing the Template**.

The existing sample templates apply to the following scenarios:

- Price summary report query
- Minimum cost supplier analysis
- Shipping priority analysis
- Analysis of order priority check
- Analysis of the number of local suppliers
- Analysis of forecasted income changes
- Freight volume analysis
- National market share analysis
- Profit estimation analysis by product type
- Analysis of returned parts
- Analysis of key inventory indicators
- Freight mode and command priority analysis
- Consumer allocation analysis
- Promotion effect analysis
- Analysis of the supplier with the largest contribution

- Analysis of the relationship between parts and suppliers
- Revenue analysis of small-lot orders
- Customer analysis for large orders
- Discounted revenue analysis
- Analysis of potential component improvements
- Analysis of suppliers who fail to deliver goods on time
- Global sales opportunity analysis

Custom Templates

The custom template list displays all templates you have created. You can view the template name, description, statements, and more.

Table 8-2 Template management parameters

Parameter	Description
Name	Indicates the template name. ● A template name can contain only digits, letters, and underscores (_), but cannot start with an underscore (_) or contain only digits. It cannot be left empty. ● The template name can contain a maximum of 50 characters.
Description	Description of the template you create.
Statement	SQL statement created as the template.
Operation	● **Execute**: After you click this button, the system switches to the **SQL Editor** page, where you can modify or directly perform the statement as required. For details, see **Executing the Template**. ● **Modify**: Click **Modify**. In the displayed **Modify Template** dialog box, modify the template information as required. For details, see **Modifying a Template**.

Creating a Template

You can create a template on either the **Job Templates** or the **SQL Editor** page.

- To create a template on the **Job Templates** page:
  a. On the left of the management console, choose **Job Templates** > **SQL Templates**.
  b. On the **SQL Templates** page, click **Create Template** to create a template.

     Enter the template name, SQL statement, and description information. For details, see **Table 8-3**.

Table 8-3 Parameter description

Parameter	Description
Name	Indicates the template name. ● A template name can contain only digits, letters, and underscores (_), but cannot start with an underscore (_) or contain only digits. It cannot be left empty. ● The template name can contain a maximum of 50 characters.
Statement	SQL statement to be saved as a template.
Description	Description of the template you create.
Group	● Use existing ● Use new ● Do not use
Group Name	If you select **Use existing** or **Use new**, you need to enter the group name.

c. Click **OK**.

- To create a template on the **SQL Editor** page:

  a. On the left of the management console, click **SQL Editor**.

  b. In the SQL job editing area of the displayed **SQL Editor** page, click **More** in the upper right corner and choose **Save as Template**.

     Enter the template name, SQL statement, and description information. For details, see **Table 8-3**.

  c. Click **OK**.

Executing the Template

Perform the template operation as follows:

1. On the left of the management console, choose **Job Templates** > **SQL Templates**.
2. On the **SQL Templates** page, select a template and click **Execute** in the **Operation** column. The **SQL Editor** page is displayed, and the corresponding SQL statement is automatically entered in the SQL job editing window.
3. In the upper right corner of the SQL job editing window, Click **Execute** to run the SQL statement. After the execution is complete, you can view the execution result below the current SQL job editing window.

Searching for a Template

On the **SQL Templates** page, you can enter the template name keyword in the search box on the upper right corner to search for the desired template.

Modifying a Template

Only custom templates can be modified. To modify a template, perform the following steps:

Step 1 On the **SQL Templates** page, locate the target template and click **Modify** in the **Operation** column.

Step 2 In the displayed **Modify Template** dialog box, modify the template name, statement, and description as required.

Step 3 Click **OK**.

----End

Deleting a Template

On the **SQL Templates** page, select one or more templates to be deleted and click **Delete** to delete the selected templates.

8.2 Flink Template Management

Flink templates include sample templates and custom templates. You can modify an existing sample template to meet the actual job logic requirements and save time for editing SQL statements. You can also customize a job template based on your habits and methods so that you can directly invoke or modify the template in later jobs.

Flink template management provides the following functions:

- **Flink SQL Sample Template**
- **Custom Templates**
- **Creating a Template**
- **Creating a Job Based on a Template**
- **Modifying a Template**
- **Deleting a Template**

Flink SQL Sample Template

The template list displays existing sample templates for Flink SQL jobs. **Table 1** describes the parameters in the template list.

Table 8-4 Parameters in the Flink SQL sample template list

Parameter	Description
Name	Name of a template, which has 1 to 64 characters and only contains letters, digits, hyphens (-), and underlines (_).
Description	Description of a template. It contains 0 to 512 characters.

Parameter	Description
Operation	**Create Job**: Create a job directly by using the template. After a job is created, the system switches to the **Edit** page under **Job Management**.

The existing sample templates apply to the following scenarios:

- Real-time ETL of Nginx access logs to the database
- Fake plate vehicle detection
- Geo-fencing alarm
- Vehicle yawing alarm
- Vehicle speeding alarm
- Stream-based random forest anomaly detection
- CloudTable-DLI Flink-CloudTable
- DIS-DLI Flink-CSS (Elasticsearch)
- DIS-DLI Flink-CloudTable (OpenTSDB)
- DIS-DLI Flink-CloudTable
- DIS-DLI Flink-DCS
- DIS-DLI Flink-DDS (MongoDB)
- DIS-DLI Flink-DIS
- DIS-DLI Flink-OBS-DWS
- DIS-DLI Flink-SMN
- Kafka-DLI Flink-Kafka
- OBS-DLI Flink-RDS
- Stream-Join-Table (DCS)
- Stream-Join-Table (RDS)

Custom Templates

The custom template list displays all Jar job templates. **Table 1** describes parameters in the custom template list.

Table 8-5 Parameters in the custom template list

Parameter	Description
Name	Name of a template, which has 1 to 64 characters and only contains letters, digits, hyphens (-), and underlines (_).
Description	Description of a template. It contains 0 to 512 characters.
Created	Time when a template is created.
Updated	Latest time when a template is modified.

Parameter	Description
Operation	● **Edit**: Modify a template that has been created. ● **Create Job**: Create a job directly by using the template. After a job is created, the system switches to the **Edit** page under **Job Management**. ● **More**: – **Delete**: Delete a created template.

Creating a Template

You can create a template using any of the following methods:

- Creating a template on the **Template Management** page

  a. In the left navigation pane of the DLI management console, choose **Job Templates** > **Flink Templates**.

  b. Click **Create Template** in the upper right corner of the page. The **Create Template** dialog box is displayed.

  c. Specify **Name** and **Description**.

  Table 8-6 Template parameters

Parameter	Description
Name	Name of a template, which has 1 to 64 characters and only contains letters, digits, hyphens (-), and underlines (_). **NOTE** The template name must be unique.
Description	Description of a template. It contains 0 to 512 characters.

  d. Click **OK** to enter the **Edit** page.

  Job template parameters

Parameter	Description
Name	You can modify the template name.
Description	You can modify the template description.
Saving Mode	● **Save Here**: Save the modification to the current template. ● **Save as New**: Save the modification as a new template.

Parameter	Description
SQL statement editing area	In the area, you can enter detailed SQL statements to implement business logic. For details about how to compile SQL statements, see the Data Lake Insight SQL Syntax Reference.
Save	Save the modifications.
Create Job	Use the current template to create a job.
Format	Format SQL statements. After SQL statements are formatted, you need to compile SQL statements again.
Theme Settings	Change the font size, word wrap, and page style (black or white background).

e. In the SQL statement editing area, enter SQL statements to implement service logic. For details about how to compile SQL statements, see the Data Lake Insight SQL Syntax Reference.

f. After the SQL statement is edited, click **Save** in the upper right corner to complete the template creation.

g. (Optional) If you do not need to modify the template, click **Create Job** in the upper right corner to create a job based on the current template. For details about how to create a job, see **Creating a Flink SQL Job**, and **Creating a Flink Jar Job**.

- Creating a template based on an existing job template
  a. In the left navigation pane of the DLI management console, choose **Job Templates** > **Flink Templates**. Click the **Custom Templates** tab.
  b. In the row where the desired template is located in the custom template list, click **Edit** under **Operation** to enter the **Edit** page.
  c. After the modification is complete, set **Saving Mode** to **Save as New**.
  d. Click **Save** in the upper right corner to save the template as a new one.
- Creating a template using a created job
  a. In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.
  b. Click **Create Job** in the upper right corner. The **Create Job** page is displayed.
  c. Specify parameters as required.
  d. Click **OK** to enter the **Edit** page.
  e. After the SQL statement is compiled, click **Set as Template**.
  f. In the **Set as Template** dialog box that is displayed, specify **Name** and **Description** and click **OK**.
- Creating a template based on the existing job
  a. In the left navigation pane of the DLI management console, choose **Job Management** > **Flink Jobs**. The **Flink Jobs** page is displayed.
  b. In the job list, locate the row where the job that you want to set as a template resides, and click **Edit** in the **Operation** column.

c. After the SQL statement is compiled, click **Set as Template**.

d. In the **Set as Template** dialog box that is displayed, specify **Name** and **Description** and click **OK**.

Creating a Job Based on a Template

You can create jobs based on sample templates or custom templates.

1. In the left navigation pane of the DLI management console, choose **Job Templates** > **Flink Templates**.
2. In the sample template list, click **Create Job** in the **Operation** column of the target template. For details about how to create a job, see **Creating a Flink SQL Job** and **Creating a Flink Jar Job**.

Modifying a Template

After creating a custom template, you can modify it as required. The sample template cannot be modified, but you can view the template details.

1. In the left navigation pane of the DLI management console, choose **Job Templates** > **Flink Templates**. Click the **Custom Templates** tab.
2. In the row where the template you want to modify is located in the custom template list, click **Edit** in the **Operation** column to enter the **Edit** page.
3. In the SQL statement editing area, modify the SQL statements as required.
4. Set **Saving Mode** to **Save Here**.
5. Click **Save** in the upper right corner to save the modification.

Deleting a Template

You can delete a custom template as required. The sample templates cannot be deleted. Deleted templates cannot be restored. Exercise caution when performing this operation.

1. In the left navigation pane of the DLI management console, choose **Job Templates** > **Flink Templates**. Click the **Custom Templates** tab.
2. In the custom template list, select the templates you want to delete and click **Delete** in the upper left of the custom template list.

   Alternatively, you can delete a template by performing the following operations: In the custom template list, locate the row where the template you want to delete resides, and click **More** > **Delete** in the **Operation** column.
3. In the displayed dialog box, click **Yes**.

8.3 Appendix

8.3.1 TPC-H Sample Data in the SQL Template

TPC-H Sample Data

TPC-H is a test set developed by the Transaction Processing Performance Council (TPC) to simulate decision-making support applications. It is widely used in academia and industry to evaluate the performance of decision-making support technology. This business test has higher requirements on vendors, because it can comprehensively evaluate the overall business computing capability. With universal business significance, is widely used in analysis of bank credit, credit card, telecom operation, tax, as well as tobacco industry decision-making analysis.

The TPC-H benchmark test is developed from TPC-D (a standard specified by TPC in 1994 and used as the test benchmark for decision-making support systems). TPC-H implements a 3NF data warehouse that contains eight basic relationships, with a data volume range from 1 GB to 3 TB. The TPC-H benchmark test includes 22 queries (Q1 to Q22). The main evaluation indicator is the response time of each query (from submission to result return). The unit of the TPC-H benchmark test is the query number per hour (QphH@size). **H** indicates the average number of complex queries per hour. **size** indicates the size of database, which reflects the query processing capability of the system. TPC-H can evaluate key performance parameters that other tests cannot evaluate, because it is modeled based on the actual production and operation environment. In a word, the TPC-H standard by TPC meets the test requirements of data warehouse and motivate vendors and research institutes to stretch the limit of this technology.

In this example, DLI directly queries the TPC-H dataset on OBS. DLI has generated a standard TPC-H-2.18 dataset of 100 MB which is uploaded to the tpch folder on OBS. The read-only permission is granted to you to facilitate query operations.

TPC-H Test and Metrics

TPC-H test is divided into three sub-tests: data loading test, Power test, and Throughput test. Data loading indicates the process of setting up a test database, and the loading test is to test the data loading ability of DBMS. The first test is data loading test that tests data loading time, which is time-consuming. The second test is Power test, also called raw query. After data loading test is complete, the database is in the initial state without any other operation, especially the data in the buffer is not tested. Power test requires that the 22 queries be executed once in sequence and a pair of RF1 and RF2 operations be executed at the same time. The third test is Throughput test, the core and most complex test, more similar to the actual application environment. With multiple query statement groups and a pair of RF1 and RF2 update flows, Throughput test pose greater pressure on the SUT system than Power test does.

The basic data in the test is related to the execution time (the time of each data loading step, each query execution, and each update execution), based on which you can calculate the data loading time, Power@Size, Throughput@Size, qphH@Size and $/QphH@Size.

Power@Size is the result of the Power test, which is defined as the reciprocal of the geometric average value of the query time and change time. The formula is as follows:

$$\text{TPC-H Power@Size} = \frac{3600 * SF}{\sqrt[24]{\prod_{i=1}^{i=22} QI(i,0) * \prod_{j=1}^{j=2} RI(j,0)}}$$

Size indicates the data size. SF is the scaling factor of data scale. QI (i, 0) indicates the time of the ith query, in seconds. R (I j, 0) is the update time of RFj, in seconds.

Throughput@Size is the Throughput test result, which is defined as the reciprocal of the average value of all query execution time. The formula is as follows:

$$\text{QphH@Size} = \sqrt{Power\ @\ Size * Throughput\ @\ Size}$$

Service Scenario

You can use the built-in TPC-H test suite of DLI to perform interactive query without uploading data.

Advantages of DLI Built-in TPC-H

- You can log in to DLI and get permission to run SQL statements without creating tables or import data.
- The 22 preset TPC-H SQL query templates with rich functions meet the requirements of most business scenarios. You do not need to download TPC-H query statements, which saves your time and energy.
- Data Lake gives you brand-new experience of serverless DLI product within the minimum time.

Precautions

When a sub-account uses the TPC-H test suite, the main account needs to grant the sub-account the OBS access permission and the permission to view the main account table. If the master account has not logged in to DLI, the sub-account needs to have the permissions to create databases and tables in addition to the preceding permissions.

Operation Description

For details, see **SQL Template Management**.

9 Datasource Connections

9.1 Datasource Connection and Cross-Source Analysis

DLI supports the datasource capability of the native Spark and extends it. With DLI datasource connection, you can access other data storage services through SQL statements, Spark jobs, and Flink jobs and import, query, analyze, and process data in the services.

Datasource Connections

Before using DLI to perform cross source analysis, you need to set up a datasource connection to enable the network between data sources.

The enhanced datasource connection uses VPC peering at the bottom layer to directly connect the VPC network between the DLI queue and the destination datasource. Data is exchanged in point-to-point mode.

NOTE

- Datasource connections cannot be created for the default queue.
- **VPC Administrator** permissions are required to use the VPC, subnet, route, VPC peering connection, and port for DLI datasource connections. You can set this parameter in **Service Authorization**.

Cross-Source Analysis

The enhanced datasource connection supports all cross-source services implemented by DLI and implements access to self-built data sources by using UDFs, Spark jobs, and Flink jobs.

Currently, DLI supports datasource connection to the following data sources: CloudTable HBase, CloudTable OpenTSDB, CSS, DCS Redis, DDS Mongo, DIS, DMS, DWS, MRS HBase, MRS Kafka, MRS OpenTSDB, OBS, RDS MySQL, RDS PostGre, and SMN.

NOTE

- To access a datasource connection table, you need to use the queue for which a datasource connection has been created.
- The preview function is not supported for datasource tables.

Datasource Connection Analysis Process

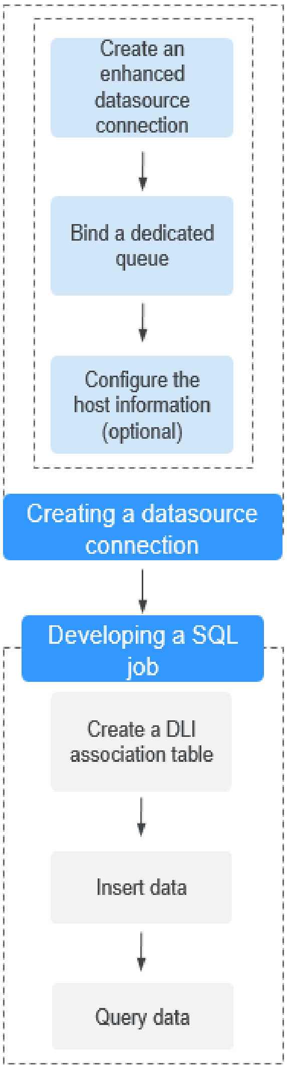
To use DLI for cross-source analysis, you need to create datasource connections and then develop different jobs to access data sources. Perform the following steps:

1. Create a datasource connection. You can create a connection in either of the following ways:
   - Create a datasource connection on the management console.
   - Create a datasource connection through an API.
2. Develop a DLI job to connect to a datasource. You can connect to a datasource in one of the following ways:
   - Develop a SQL job to connect to a datasource.
   - Develop a Spark job to connect to a datasource.
   - Develop a Flink job to connect to a datasource.

The following describes the basic processes of developing SQL jobs, Spark jobs, and Flink jobs for datasource connection.

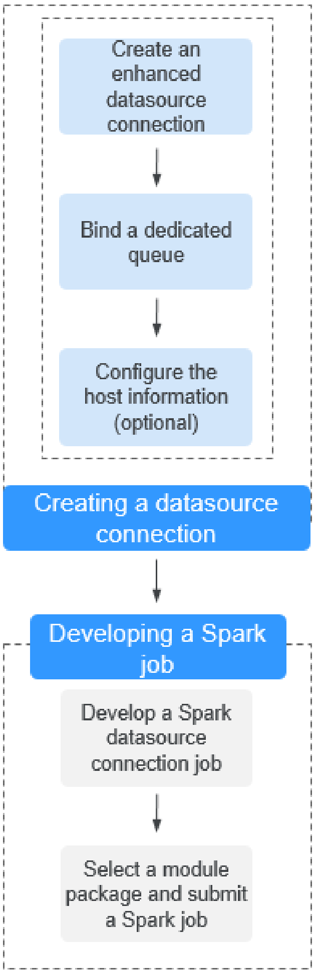
- SQL Job

Figure 9-1 Datasource connection analysis process


Create an enhanced datasource connection
Bind a dedicated queue
Configure the host information (optional)
Creating a datasource connection
Developing a SQL job
Create a DLI association table
Insert data
Query data

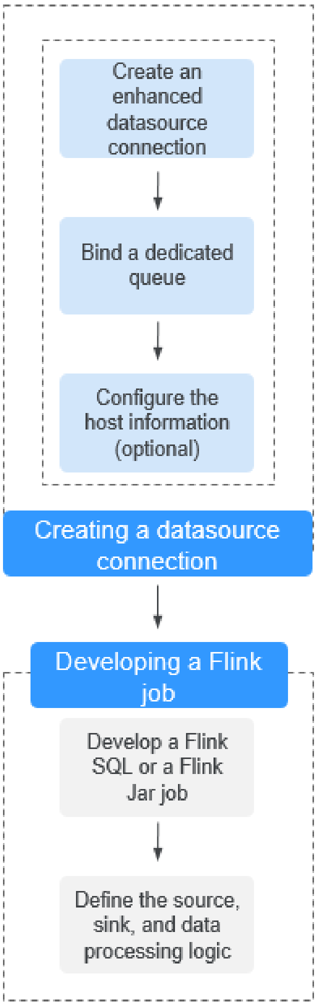
- Spark Job

Figure 9-2 Datasource connection analysis process


Create an enhanced datasource connection
Bind a dedicated queue
Configure the host information (optional)
Creating a datasource connection
Developing a Spark job
Develop a Spark datasource connection job
Select a module package and submit a Spark job

- Flink Job

Figure 9-3 Datasource connection analysis process


Create an enhanced datasource connection
Bind a dedicated queue
Configure the host information (optional)
Creating a datasource connection
Developing a Flink job
Develop a Flink SQL or a Flink Jar job
Define the source, sink, and data processing logic

9.2 Enhanced Datasource Connections

9.2.1 Overview

The enhanced datasource connection uses VPC peering at the bottom layer to directly connect the VPC network between the DLI cluster and the destination datasource. Data is exchanged in point-to-point mode. The enhanced datasource connection function supports all cross-source services implemented by DLI, including CloudTable HBase, CloudTableOpenTSDB, MRS OpenTSDB, DWS, RDS, CSS, DCS, and DDS. In addition, UDFs, Spark jobs, and Flink jobs can be used to access self-built data sources.

NOTE

- The CIDR block of the DLI queue bound with a datasource connection cannot overlap with that of the data source.
- Datasource connections cannot be created for the default queue.
- To access a datasource connection table, you need to use the queue for which a datasource connection has been created.
- The preview function is not supported for datasource tables.

The enhanced datasource scenario provides the following functions:

- **Creating, Querying, and Deleting an Enhanced Datasource Connection**
- **Binding and Unbinding a Queue**
- **Modifying Host Information**
- **Custom Route Information**
- **Enhanced Datasource Connection Permission Management**

Enhanced Datasource Connection Page

This page displays all enhanced datasource connections. If there are a large number of connections, they are displayed on multiple pages.

Table 9-1 Datasource connection list parameters

Parameter	Description
Connection Name	Name of the created datasource connection.
Connection Status	Status of a datasource connection. Currently, the console displays only connections in the **Active** state.
VPC	VPC used by the destination data source.
Subnet	Subnet used by the destination data source.
Created	Time when a connection is created. The connections in the connection list can be displayed according to the creation time in ascending or descending order.

Parameter	Description
Operation	● **Delete Connection**: Delete a created enhanced datasource connection. ● **Manage Route**: Used to display the custom route information of the enhanced datasource connection. ● **More**: – **Modify Host**: Customize the IP address corresponding to the host or domain name. – **Bind Queue**: Bind a queue to an enhanced datasource connection. – **Unbind Queue**: Unbind an enhanced datasource connection from a queue. – **Add Route**: Add a custom route for the enhanced datasource connection. – **Delete Route**: Delete a custom route for an enhanced datasource connection. – **Manage Permissions**: Authorize or reclaim permissions for other projects.

9.2.2 Creating, Querying, and Deleting an Enhanced Datasource Connection

Creating an Enhanced Datasource Connection

The following describes how to create a datasource HBase connection for MRS.

NOTE

Only enhanced datasource connection to MRS HBase is supported.

1. Apply for a cluster in MRS.

   If a cluster is available, you do not need to apply for one.

2. In the navigation pane of the DLI management console, choose **Datasource Connections**.
3. Click the **Enhanced** tab and click **Create** in the upper left corner.

   Enter the **Connection Name**, select the **Bind Queue** (optional), **VPC**, and **Subnet**, and enter the **Host Information** (optional). For details about the parameters, see **Table 9-2**.

Table 9-2 Parameter description

Parameter	Description
Connection Name	Name of the created datasource connection. ● The name can contain only letters, digits, and underscores (_). The parameter must be specified. ● A maximum of 64 characters are allowed.
Bound Queue	(Optional) Bind a queue that requires datasource connections. **NOTE** Before using an enhanced datasource connection, you must bind a queue and ensure that the VPC peering connection is in the **Active** state.
VPC	VPC used by the destination data source.
Subnet	Subnet used by the destination data source.
Host Information	(Optional) When connecting to the HBase cluster of MRS, enter the host name and IP address of the ZooKeeper instance. Enter one record in each line. The format is as follows: **IP address Host name/Domain name**. To obtain the host name and IP address of the MRS cluster, perform the following steps (with MRS3.x as an example): 1. Log in to the MRS management console. 2. In the navigation pane, choose **Clusters** > **Active Clusters**. Click the target cluster name to access the cluster details page. 3. Click **Component Management**. 4. Click **Zookeeper**. 5. Click the **Instance** tab to view the corresponding service IP address. You can select any service IP address. **NOTE** If the MRS cluster has multiple IP addresses, enter any service IP address when creating a datasource connection.

4. Click **OK**.
5. To connect to Kafka, DWS, and RDS instances, add security group rules for the DLI network segment to the security group where the instances belong. The following uses RDS as an example:
   a. Choose **Resources** > **Queue Management** in the navigation pane. Select the target queue, and click ⌄ to expand the row containing the target queue to view its CIDR block.
   b. On the **Instance Management** page of the RDS console, click the instance name. In the **Connection Information** area, view the port number of the RDS database instance.
   c. In the **Connection Information** area locate the **Security Group** and click the group name to switch to the security group management page.

Select the **Inbound Rules** tab and click **Add Rule**. Set the priority to 1, protocol to TCP, port to the database port number, and source to the CIDR block of the DLI queue. Click **OK**.

6. Test the connectivity between the DLI queue and the connection instance. The following example describes how to test the connectivity between DLI and an RDS DB instance.
   a. On the **Instance Management** page, click the target DB instance. On the displayed page, locate the **Connection Information** pane and view the floating IP address. In the **Connection Information** pane, locate the **Database Port** to view the port number of the RDS DB instance.
   b. Go back to the DLI console. On the **Resources** > **Queue Management** page, locate the target queue. In the **Operation** column, click **More** > **Test Address Connectivity**.
   c. Enter the connection address of the RDS DB instance and port number in the format of **IP address:port** to test the network connectivity.

Querying an Enhanced Datasource Connection

On the **Enhanced** tab page, you can enter the keyword of a connection name in the search box to search for the matching connection.

Viewing Details

On the **Enhanced** tab page, select a connection and click ˅ to view its details. The connection ID and host information are displayed.

Deleting an Enhanced Datasource Connection

On the **Enhanced** tab page, click **Delete Connection** in the **Operation** column to delete an unnecessary connection.

NOTE

A connection with **Connection Status** of **Creating** cannot be deleted.

9.2.3 Binding and Unbinding a Queue

Constraints

- The CIDR block of the DLI queue that is bound with a datasource connection cannot overlap with that of the data source.
- The default queue cannot be bound with a connection.

Binding a Queue

Before using an enhanced datasource connection, you must bind a queue and ensure that the VPC peering connection is in the **Active** state.

- Method 1

  In the **Enhanced** tab, select a connection and click **Bind Queue** in the **Operation** column. In the dialog box that is displayed, select the queues to be bound and click **OK**.

- Method 2

  Click the name of the selected connection. The **Details** page is displayed. Click **Create** in the upper left corner. In the dialog box that is displayed, select the queues to be bound and click **OK**.

Viewing Details about a Bound Queue

On the **Enhanced** tab page, select a connection and click the connection name to view the information about the bound queue.

Table 9-3 Parameters in the details list of datasource connection queues

Parameter	Description
VPC Peering ID	ID of the VPC peering connection created in the cluster to which the queue belongs. **NOTE** A VPC peering connection is created for each queue bound to an enhanced datasource connection. The VPC peering connection is used for cross-VPC communication. Ensure that the security group used by the data source allows access from the DLI queue CIDR block, and do not delete the VPC peering connection during the datasource connection.
Name	Name of a bound queue.
Connection Status	Datasource connection status. The following three statuses are available: ● Creating ● Active ● Failed **NOTE** If the connection status is **Failed**, click ⌄ on the left to view the detailed error information.
Updated	Time when a connection is updated. The connections in the connection list can be displayed according to the update time in ascending or descending order.
Operation	**Unbind Queue**: This operation is used to unbind a datasource connection from a queue.

Unbinding a Queue

If you do not need to use an enhanced datasource connection, you can unbind the queue from it to release resources.

- Method 1

  On the **Enhanced** tab page, select a connection and click **More** > **Unbind Queue** in the **Operation** column. In the dialog box that is displayed, select the queues to be unbound and click **OK**.

- Method 2

  Click the name of the target connection. The **Details** page is displayed. Select the queue to be unbound and click **Unbind Queue** in the **Operation** column. In the displayed dialog box, confirm the queue to be unbound and click **Yes**.

9.2.4 Modifying Host Information

Modifying Hosts Information

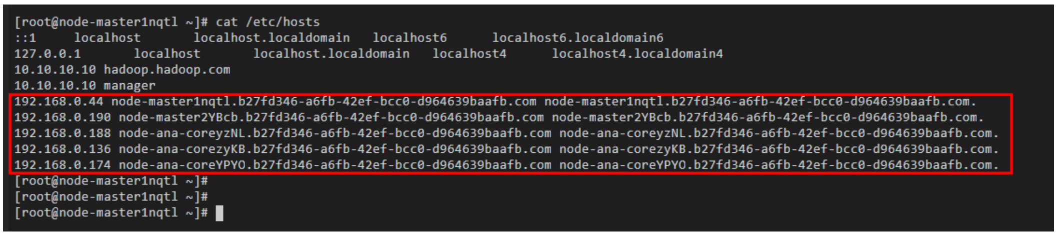
Method 1: Copy hosts information in /etc/hosts of an MRS node.

Step 1 Log in to any MRS node as the **root** user.

Step 2 Run the following command to obtain MRS hosts information. Copy and save the information.

cat /etc/hosts

Figure 9-4 Obtaining hosts information



```
[root@node-master1nqtl ~]# cat /etc/hosts
::1     localhost       localhost.localdomain   localhost6      localhost6.localdomain6
127.0.0.1       localhost       localhost.localdomain   localhost4      localhost4.localdomain4
10.10.10.10 hadoop.hadoop.com
10.10.10.10 manager
192.168.0.44 node-master1nqtl.b27fd346-a6fb-42ef-bcc0-d964639baafb.com node-master1nqtl.b27fd346-a6fb-42ef-bcc0-d964639baafb.com.
192.168.0.190 node-master2YBcb.b27fd346-a6fb-42ef-bcc0-d964639baafb.com node-master2YBcb.b27fd346-a6fb-42ef-bcc0-d964639baafb.com.
192.168.0.188 node-ana-coreyzNL.b27fd346-a6fb-42ef-bcc0-d964639baafb.com node-ana-coreyzNL.b27fd346-a6fb-42ef-bcc0-d964639baafb.com.
192.168.0.136 node-ana-corezyKB.b27fd346-a6fb-42ef-bcc0-d964639baafb.com node-ana-corezyKB.b27fd346-a6fb-42ef-bcc0-d964639baafb.com.
192.168.0.174 node-ana-coreYPYO.b27fd346-a6fb-42ef-bcc0-d964639baafb.com node-ana-coreYPYO.b27fd346-a6fb-42ef-bcc0-d964639baafb.com.
[root@node-master1nqtl ~]#
[root@node-master1nqtl ~]#
[root@node-master1nqtl ~]# 
```

Step 3 Log in to the DLI console, choose **Datasource Connections** > **Enhanced**. On the **Enhanced** tab page, select a connection and click **Modify Host**. In the displayed dialog box, paste the host information copied in **Step 2**. Click **OK**.

----End

Method 2: Log in to FusionInsight Manager to obtain MRS hosts information.

Step 1 Log in to FusionInsight Manager.

Step 2 On FusionInsight Manager, click **Hosts**. On the **Hosts** page, obtain the host names and service IP addresses of the MRS hosts.

Step 3 Log in to the DLI console, choose **Datasource Connections** > **Enhanced**. On the **Enhanced** tab page, select a connection and click **Modify Host**. In the displayed dialog box, enter the host information. Click **OK**.

The host information is in the format of *Service IP address Host name*. Specify the IP addresses and host names obtained in **Step 2**, and separate multiple records by line breaks.

For example:

192.168.0.22 node-masterxxx1.com

192.168.0.23 node-masterxxx2.com

----End

NOTE

- After this configuration, all hosts information is overwritten.
- A host name or domain name can contain a maximum of 128 characters, including digits, letters, underscores (_), hyphens (-), and periods (.). It must start with a letter.
- The host information is *IP address Host name/Domain name* format. Separate multiple records by line breaks.

9.2.5 Custom Route Information

Customizing Route Information

After an enhanced datasource connection is created and bound to a queue, the system automatically configures route information. You can also add a custom route for the queue to which the enhanced connection is bound.

- Viewing route information

  On the **Enhanced** tab page, select a connection and click **Manage Route** in the **Operation** column to view the route information of the datasource connection.

- Adding a route

  On the **Enhanced** tab page, select a connection and choose **More** > **Add Route** in the **Operation** column, or click **Add Route** on the **Details** page of the connection to add a custom route. In the displayed dialog box, enter the route name and route CIDR block. For details about the parameters, see **Table 9-4**.

  Table 9-4 Parameters for adding a custom route

Parameter	Description
Route Name	Name of a custom route, which is unique in the same enhanced datasource scenario. The name contains 1 to 64 characters, including digits, letters, underscores (_), and hyphens (-).
IP Address	Custom route CIDR block. The CIDR block of different routes can overlap but cannot be the same.

- Deleting a route

  On the **Enhanced** page, select a connection and choose **More** > **Delete Route** in the **Operation** column, or click **Delete Route** on the **Details** page of the connection to delete a custom route.

9.2.6 Enhanced Datasource Connection Permission Management

Permission Management

You can grant or revoke permissions for enhanced datasource connections of other projects.

- Authorization
  a. Log in to the DLI management console, choose **Datasource Connections** and select the **Enhanced** tab, select a datasource connection, and choose **More** > **Manage Permission** in the **Operation** column. In the displayed **Permissions** dialog box, select **Grant Permission**, enter the project ID, and click **OK**.
  b. After a project is authorized, you can log in to the system as a user of the authorized project or switch to the corresponding project. In the **Enhanced** tab, you can view the authorized datasource connection and bind the created queue to the datasource connection. Cross-project datasource connections and routes can be created.

  NOTE

  - If the authorized projects belong to different users in the same region, you can use the user account of the authorized projects to log in.
  - If the authorized projects belong to the same user in the same region, you can use the current account to switch to the corresponding project.

  For example, if project B needs to access the data source of project A, perform the following operations:

  – For Project A:
    i. Log in to DLI using the account of project A.
    ii. Create an enhanced datasource connection **ds** in DLI based on the VPC information of the corresponding data source.
    iii. Grant project B the permission to access the enhanced datasource connection **ds**.

  – For Project B:
    i. Log in to DLI using the account of project B.
    ii. Bind the enhanced datasource connection **ds** to a queue.
    iii. (Optional) Set host information and create a route.

  After creating a VPC peering connection and route between the enhanced datasource connection of project A and the queue of project B, you can create a job in the queue of project B to access the data source of project A.
- Revoke: In the **Manage Permissions** dialog box, select **Revoke Permission** and select the ID of the project whose permissions need to be revoked.

9.3 Managing Datasource Connection Permissions

Scenarios

- You can isolate datasource connections allocated to different users by setting permissions to ensure data query performance.
- The administrator and datasource connection owner have all permissions, which cannot be set or modified by other users.

On the **Datasource Authentication** tab page, click **Manage Permissions** in the **Operation** column of the row that contains the authentication to be modified. On the **User Permission Info** page that is displayed, you can grant, set, and revoke permissions of the datasource connection.

Granting Permissions on Datasource Connections

Log in to the DLI management console, choose **Datasource Connections**. Click the **Datasource Authentication** tab, locate the target authentication certificate, click **Permission Management** in the **Operation** column. Click **Grant Permission** in the upper right corner of the page.

Table 9-5 Permission granting parameters

Parameter	Description
Username	Name of the authorized IAM user. **NOTE** The username is the name of an existing IAM user.
Select the permissions to be granted to the user	● Access: This permission allows you to access the datasource connection. ● Update: This permission allows you to update the datasource connection. ● Delete: This permission allows you to delete the datasource connection. ● Grant Permission: This permission allows you to grant the datasource connection permission to other users. ● Grant Permission: This permission allows you to revoke the datasource connection permission to other users. However, you cannot revoke the permissions of the datasource connection owner. ● View Other User's Permissions: This permission allows you to view the datasource connection permissions of other users.

Setting Permissions

Click **Set Permission** in the **Operation** column of the sub-user to modify the permission of the user. **Table 9-5** lists the detailed permission descriptions.

If all options under **Set Permission** are gray, you are not allowed to change permissions on this datasource connection. You can apply for the granting and revoking permissions from administrators, group owners, and other users who have the permission to grant permissions.

Revoking Permissions

Click **Revoke Permission** in the **Operation** column of a sub-user to revoke the user's permissions. After this operation, the sub-user does not have any permission of the datasource connection.

9.4 Creating and Managing Datasource Authentication

Scenarios

Datasource authentication is used to manage authentication information for accessing CSS and MRS security environments and encrypt passwords for accessing DWS, RDS, DDS, and DCS data sources.

- When creating a CSS cluster in the security mode, you need to specify the username and password. A certificate is available for downloading the security CSS cluster. To access the security CSS cluster, the username, password, and certificate need to be stored in DLI. The username, password, and certificate are the datasource authentication information for accessing the CSS cluster.
- When creating an MRS cluster in the security mode, you need to enable the kerberos authentication. Download the authentication credential, which contains the **krb5.conf** and **user.keytab** files. When connecting DLI to an MRS cluster, you can store the two files in DLI. The two files are the datasource authentication information for accessing the MRS cluster.
- To create a Kafka cluster, you need to enable SSL access. You can configure this function on the **Service Configuration** page of MRS and Kafka, or set up an open-source Kafka cluster and modify the configuration file to enable this function.
- Through datasource password authentication, the passwords for accessing DWS, RDS, DDS, and DCS data sources can be encrypted for storage.

Creating a Datasource Authentication

1. Create a data source to be accessed.

   NOTE

   If a cluster is available, you do not need to apply for one.

   - Create a CSS cluster in the security mode: Select 6.5.4 or a later version for **Cluster Version** and enable the **Security Mode**.
   - Create an MRS cluster in the security mode: On the MRS management console, apply for a cluster, select **Custom Config**, set **Cluster Version** to MRS 2.1.0 or later, and enable **Kerberos Authentication**.
   - Creating a Kafka cluster: You can configure the Kafka cluster on the Service Configuration page of MRS and Kafka. For details about how to configure SSL on MRS, see ***Security Configuration*** in the *MapReduce Service User Guide*.
   - Creating a DWS cluster: Apply for a data warehouse cluster from DWS. For details, see the *Data Warehouse Service Management Guide*.
   - Creating an RDS data source: Apply for a database instance in the RDS service. For details, see the *Relational Database Service Getting Started*.
   - Creating a DCS data source: Apply for a DCS instance in DCS. For details, see the *Distributed Cache Service User Guide*.
   - Creating a DDS data source: Apply for a DDS DB instance. For details, see *Document Database Service Getting Started*.

2. Download the authentication credential. Skip this step when you access DWS, RDS, DCS, or DDS data sources.
   - CSS security cluster: On the **Cluster Management** page, click the cluster name. On the **Basic Information** page that is displayed, find the **Security Mode** and download the certificate of the CSS security cluster.
   - MRS security cluster: On the cluster list page, click the cluster name. On the displayed cluster information page, download the authentication credential. After downloading the authentication credential, decompress it to obtain the **krb5.conf** and **user.keytab** files.
3. Upload the authentication credential. Skip this step when you access DWS, RDS, DCS, or DDS data sources.

   Upload the obtained authentication credential file to the user-defined OBS bucket.
4. On the DLI management console, click **Datasource Connections**.
5. On the **Datasource Authentication** tab page, click **Create** to create an authentication information.
   - CSS

Table 9-6 Parameters

Parameter	Description
Type	Select CSS.
Authentica tion Certificate	Name of the datasource authentication information to be created. ● The name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). ● The length of the database name cannot exceed 128 characters. ● It is recommended that the name contain the CSS security cluster name to distinguish security authentication information of different clusters.
Username	Username for logging in to the security cluster.
Password	The password of the security cluster
Certificate Path	The OBS path for uploading the security certificate

   - MRS

Table 9-7 Description

Parameter	Description
Type	Select **Kerberos**.

Parameter	Description
Authentication Certificate	Name of the datasource authentication information to be created. ● The name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). ● The length of the database name cannot exceed 128 characters. ● It is recommended that the name contain the MRS security cluster name to distinguish security authentication information of different clusters.
Username	Username for logging in to the security cluster.
krb5_conf Path	OBS path to which the **krb5.conf** file is uploaded **NOTE** The **renew_lifetime** configuration item under **[libdefaults]** must be removed from **krb5.conf**. Otherwise, the **Message stream modified (41)** problem may occur.
keytab Path	OBS path to which the **user.keytab** file is uploaded

– Kafka

Table 9-8 Parameters

Parameter	Description
Type	Select **Kafka_SSL**.
Authentication Certificate	Name of the datasource authentication information to be created. ● The name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). ● The length of the database name cannot exceed 128 characters.
Truststore Path	OBS path to which the SSL truststore file is uploaded.
Truststore Password	Truststore password. The default value is **dms@kafka**.
Keystore Path	OBS path to which the SSL keystore file (key and certificate) is uploaded.
Keystore Password	Keystore (key and certificate) password.
Key Password	Password of the private key in the keystore file.

– Password (datasource password authentication)

Create datasource authentication for accessing DWS, RDS, DCS, and DDS data sources.

NOTE

Currently, database password authentication supports Spark SQL jobs only.

Table 9-9 Parameters

Parameter	Description
Type	Select **Password**.
Authentication Certificate	Name of the datasource authentication information to be created. ● The name can contain only digits, letters, and underscores (_), but cannot contain only digits or start with an underscore (_). ● The length of the database name cannot exceed 128 characters.
Username	Username for accessing the datasource
Password	Password for accessing the datasource

Searching for a Datasource Authentication

On the **Datasource Authentication** tab, you can enter the authentication information name in the search box to search for the matching authentication information. To ensure user information security, the password field is not returned.

Updating Authentication Information

On the **Datasource Authentication** tab, click **Update** in the **Operation** column of the authentication information to be modified. Currently, only the username and password can be updated. If you need to update the certificate, delete the authentication information and create a new one.

NOTE

The username and password are optional. If they are not set, the field is not modified.

Deleting a Datasource Authentication

On the **Datasource Authentication** tab, click **Delete** in the **Operation** column of the authentication information to be deleted.

10 Global Configuration

10.1 Global Variables

Global variables can be used to simplify complex parameters. For example, long and difficult variables can be replaced to improve the readability of SQL statements.

Creating Variables

1. In the navigation pane of the DLI console, choose **Global Configuration** > **Global Variables**.
2. On the **Global Variables** page, click **Create** in the upper right corner to create a global variable.

Table 10-1 Parameters description

Parameter	Description
Variable	Name of the created global variable.
Sensitive	If the value is sensitive information, such as passwords, set this parameter to **Yes**. Otherwise, set it to **No**.
Value	Global variable value.

3. After creating a global variable, use **{{xxxx}}** in the SQL statement to replace the parameter value set as the global variable. **xxxx** indicates the variable name. For example, if you set global variable **abc** to represent the table name, replace the actual table name with **{{abc}}** in the table creation statement.

```
create table {{table_name}} (String1 String, int4 int, varchar1 varchar(10))
  partitioned by (int1 int,int2 int,int3 int)
```

NOTE

- Only the user who creates a global variable can use the variable.
- Do not use global variables in **OPTIONS** of the table creation statements.

Modifying Variables

On the **Global Variables** page, click **Modify** in the **Operation** column of a variable to modify the variable value.

NOTE

Only the user who creates a global variable can modify the variable.

Deleting Variables

On the **Global Variables** page, click **Delete** in the **Operation** column of a variable to delete the variable value.

NOTE

- Only the user who creates a global variable can delete the variable.
- After a variable is deleted, the variable cannot be used in SQL statements.

10.2 Service Authorization

Prerequisites

Only the tenant account or a subaccount of user group **admin** can authorize access.

Procedure

After entering the DLI management console, you are advised to set agency permissions to ensure that DLI can be used properly.

If you need to adjust the agency permissions, modify them on the **Service Authorization** page. For details about the required agency permissions, see **Table 10-2**.

1. Select required agency permissions and click **Update Authorization**. Only the tenant account or a subaccount of user group **admin** can authorize access. If the message "Agency permissions updated" is displayed, the update is successful.
2. Once service authorization has succeeded, an agency named **dli_admin_agency** on IAM will be created. Go to the agency list to view the details. Do not delete **dli_admin_agency**.

Table 10-2 DLI agency permissions

Permission	Details	Remarks
Tenant Administrator (global service)	**Tenant Administrator** permissions are required to access data from OBS to execute Flink jobs on DLI, for example, obtaining OBS/DWS data sources, log dump (including bucket authorization), checkpointing enabling, and job import and export.	Due to cloud service cache differences, permission setting operations require about 60 minutes to take effect.
DIS Administrator	**DIS Administrator** permissions are required to use DIS data as the data source of DLI Flink jobs.	Due to cloud service cache differences, permission setting operations require about 30 minutes to take effect.
VPC Administrator	**VPC Administrator** permissions are required to use the VPC, subnet, route, VPC peering connection, and port for DLI datasource connections.	Due to cloud service cache differences, permission setting operations require about 3 minutes to take effect.
SMN Administrator	To receive notifications when a DLI job fails, **SMN Administrator** permissions are required.	Due to cloud service cache differences, permission setting operations require about 3 minutes to take effect.
Tenant Administrator (project-level)	**Tenant Administrator (Project-level)** permission is needed if you use services that can run only with this permission.	Due to cloud service cache differences, permission setting operations require about 3 minutes to take effect.

11 UDFs

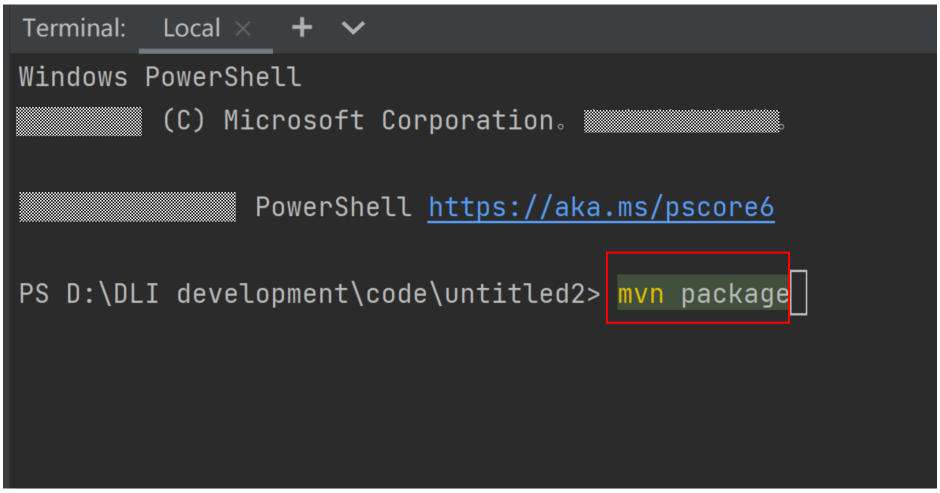
DLI allows you to query data by using the user-defined functions (Hive UDF).

Procedure

1. Compile a UDF.
   a. Add the following information to the POM file:
   ```
   <dependency>
       <groupId>org.apache.hive</groupId>
       <artifactId>hive-exec</artifactId>
       <version>1.2.1</version>
   </dependency>
   ```
   b. Add **import org.apache.hadoop.hive.ql.exec.UDF** to the code.
   c. Create or modify the content of **SumUdfDemo.java** in the sample code based on service requirements.
2. Generate a JAR package. Run the **mvn package** command to compile the program and generate the JAR package.

   For example, after the program is compiled using IDEA, click **Terminal** at the bottom of the tool pane and enter **mvn package** in the command line window to pack the program.

Figure 11-1 Compiling and packaging the program



```
Windows PowerShell
[redacted] (C) Microsoft Corporation。 [redacted]

[redacted] PowerShell https://aka.ms/pscore6

PS D:\DLI development\code\untitled2> mvn package
```

Obtain the generated JAR package, for example, **TestUDF.ja**, from the path where the compilation result is stored.

3. Upload **TestUDF.jar** to OBS. For details about how to upload data to OBS, see "Step 2: Upload Data to OBS" in **Creating and Submitting a Spark SQL Job**.
4. Upload the **TestUDF.jar** package to DLI package management.
   a. Log in to the DLI management console and choose **Data Management** > **Package Management**.
   b. On the **Package Management** page, click **Create** in the upper right corner.
   c. In the **Create Package** dialog, set the following parameters:
      i. **Type**: Select **JAR**.
      ii. **OBS Path**: Specify the OBS path in **3** for storing the package.
      iii. Set **Group** and **Group Name** as required for package identification and management.
   d. Click **OK**.
5. Create a function.

   Run the following command on the management console to create a function:

   ```
   CREATE FUNCTION fun1 AS 'com.demo.SumUdfDemo' using jar 'obs://udf/TestUDF.jar';
   ```
6. Restart the original SQL queue for the added function to take effect.
   a. Log in to the DLI management console and choose **Resources** > **Queue Management** from the navigation pane. In the **Operation** column of the SQL queue job, click **Restart**.
   b. In the **Restart** dialog box, click **OK**.
7. Use the created function.

   Run the following statement to query using the function created in **5**.

   ```
   select fun1(ip) from ip_tables;
   ```
8. Delete the created function.

   If this function is no longer used, run the following statement to delete the function:

   ```
   Drop FUNCTION fun1;
   ```

Sample Code

The sample code in **SumUdfDemo.java** is as follows:

```
package com.demo;
  import org.apache.hadoop.hive.ql.exec.UDF;
  public class SumUdfDemo extends UDF {
    public Int evaluate(Int a, Int b) {
     return a + b;
  }
 }
```

12 Permissions Management

12.1 Overview

DLI has a comprehensive permission control mechanism and supports fine-grained authentication through Identity and Access Management (IAM). You can create policies in IAM to manage DLI permissions. You can use both the DLI's permission control mechanism and the IAM service for permission management.

Application Scenarios of IAM Authentication

When using DLI on the cloud, enterprise users need to manage DLI resources (queues) used by employees in different departments, including creating, deleting, using, and isolating resources. In addition, data of different departments needs to be managed, including data isolation and sharing.

DLI uses IAM for refined enterprise-level multi-tenant management. IAM provides identity authentication, permissions management, and access control, helping you securely access to your cloud resources.

With IAM, you can use your cloud account to create IAM users for your employees, and assign permissions to the users to control their access to specific resource types. For example, some software developers in your enterprise need to use DLI resources but must not delete them or perform any high-risk operations. To achieve this result, you can create IAM users for the software developers and grant them only the permissions required for using DLI resources.

NOTE

For a new user, you need to log in for the system to record the metadata before using DLI.

IAM is free of charge. You pay only for the resources you use.

If your cloud account does not need individual IAM users for permissions management, skip this chapter.

DLI System Permissions

Table 12-1 lists all the system-defined roles and policies supported by DLI.

Type: There are roles and policies.

- Roles: A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. This mechanism provides only a limited number of service-level roles for authorization. When using roles to grant permissions, you need to also assign other roles on which the permissions depend to take effect. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant DLI users only the permissions for managing a certain type of ECSs.

Table 12-1 DLI system permissions

Role/Policy Name	Description	Category
DLI FullAccess	Full permissions for DLI.	System-defined policy
DLI ReadOnlyAccess	Read-only permissions for DLI.	System-defined policy
Tenant Administrator	Tenant administrator ● Administer permissions for managing and accessing all cloud services. After a database or a queue is created, the user can use the ACL to assign rights to other users. ● Scope: project-level service	System-defined role
DLI Service Admin	DLI administrator ● Administer permissions for managing and accessing the queues and data of DLI. After a database or a queue is created, the user can use the ACL to assign rights to other users. ● Scope: project-level service	System-defined role

DLI Permission Types

Table 12-2 lists the DLI service permissions. For details about the resources that can be controlled by DLI, see **Table 12-7**.

Table 12-2 DLI permission types

Permission Type	Subtype	Console Operations	SQL Syntax	API Definition
Queue Permissions	Queue management permissions	For details, see **Queue Permission Management**.	None	For details, see "Granting Users with the Queue Usage Permission" in the *Data Lake Insight API Reference*.
	Queue usage permission			
Data Permissions	Database permissions	For details, see **Database Permission Management** and **Table Permission Management**.	For details, see **SQL Syntax of Batch Jobs** > **Data Permissions Management** > **Data Permissions List** in the *Data Lake Insight SQL Syntax Reference*.	For details, see **Permission-related APIs** > **Granting Users with the Data Usage Permission** in the *Data Lake Insight API Reference*.
	Table permissions			
	Column permissions			
Job Permissions	Flink job permissions	For details, see **Managing Flink Job Permissions**.	None	For details, see **Permission-related APIs** > **Granting Users with the Data Usage Permission** in the *Data Lake Insight API Reference*.
Package Permissions	Package group permissions	For details, see **Managing Permissions on Packages and Package Groups**.	None	For details, see **Permission-related APIs** > **Granting Users with the Data Usage Permission** in the *Data Lake Insight API Reference*.
	Package permissions			

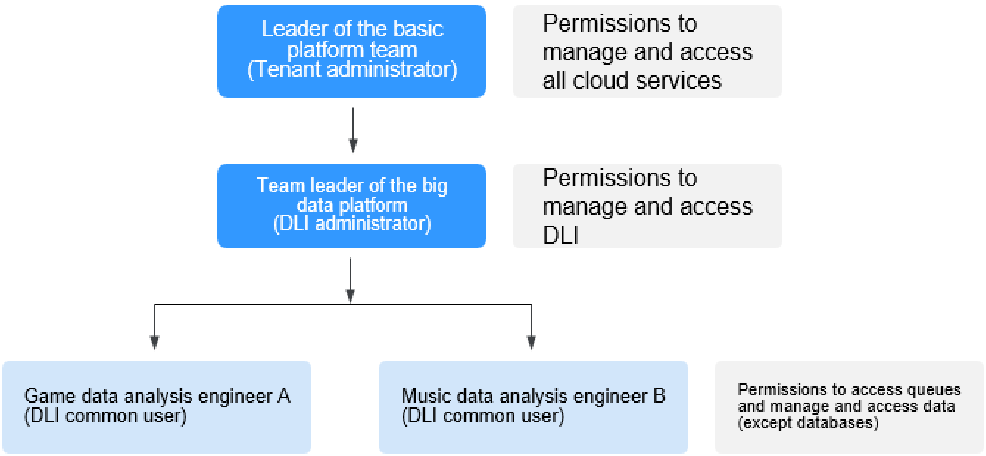
Permission Type	Subtype	Console Operations	SQL Syntax	API Definition
Datasource Connection Permissions	Datasource connection permissions	For details, see **Managing Datasource Connection Permissions**.	None	For details, see **Permission-related APIs** > **Granting Users with the Data Usage Permission** in the *Data Lake Insight API Reference*.

Examples

An Internet company mainly provides game and music services. DLI is used to analyze user behaviors and assist decision making.

As shown in **Figure 12-1**, the **Leader of the Basic Platform Team** has applied for a **Tenant Administrator** account to manage and use cloud services. Since the **Big Data Platform Team** needs DLI for data analysis, the **Leader of the Basic Platform Team** adds a subaccount with the permission of **DLI Service Admin** to manage and use DLI. The **Leader of the Basic Platform Team** creates a **Queue A** and assigns it to **Data Engineer A** to analyze the gaming data. He also assigns a **Queue B** to **Data Engineer B** to analyze the music data. Besides granting the queue usage permission, the **Leader of the Basic Platform Team** grants data (except the database) management and usage permissions to the two engineers.

Figure 12-1 Granting permissions


Leader of the basic platform team (Tenant administrator)
Permissions to manage and access all cloud services
Team leader of the big data platform (DLI administrator)
Permissions to manage and access DLI
Game data analysis engineer A (DLI common user)
Music data analysis engineer B (DLI common user)
Permissions to access queues and manage and access data (except databases)

The **Data Engineer A** creates a table named **gameTable** for storing game prop data and a table named **userTable** for storing game user data. The music service is a new service. To explore potential music users among existing game users, the **Data Engineer A** assigns the query permission on the **userTable** to the **Data**

Engineer B. In addition, **Data Engineer B** creates a table named **musicTable** for storing music copyrights information.

Table 12-3 describes the queue and data permissions of **Data Engineer A** and **Data Engineer B**.

Table 12-3 Permission description

User	Data Engineer A (game data analysis)	Data Engineer B (music data analysis)
Queues	Queue A (queue usage permission)	Queue B (queue usage permission)
Data (Table)	gameTable (table management and usage permission)	musicTable (table management and usage permission)
	userTable (table management and usage permission)	userTable (table query permission)

NOTE

The queue usage permission includes job submitting and terminating permissions.

12.2 Creating an IAM User and Granting Permissions

You can use Identity and Access Management (IAM) to implement fine-grained permissions control on DLI resources. For details, see **Overview**.

If your cloud account does not need individual IAM users, then you may skip over this chapter.

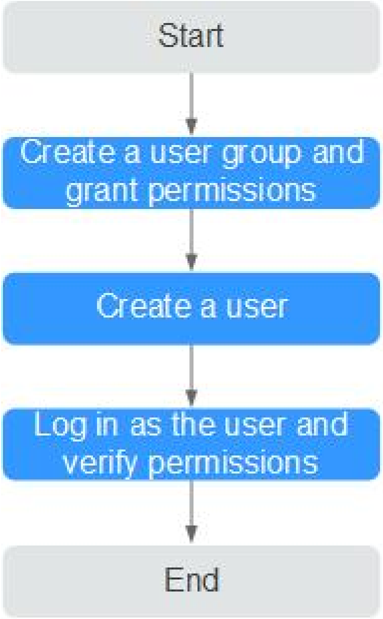
This section describes how to create an IAM user and grant DLI permissions to the user. **Figure 12-2** shows the procedure.

Prerequisites

Before assigning permissions to user groups, you should learn about system policies and select the policies based on service requirements. For details about system permissions supported by DLI, see **DLI System Permissions**.

Process Flow

Figure 12-2 Process for granting DLI permissions


Start
Create a user group and grant permissions
Create a user
Log in as the user and verify permissions
End

1. Create a user group and grant the permission to it.

   Create a user group on the IAM console, and assign the **DLI ReadOnlyAccess** permission to the group.

2. Create a user and add the user to the user group.

   Create a user on the IAM console and add the user to the group created in **1**.

3. Log in and verify the permission.

   Log in to the management console using the newly created user, and verify that the user's permissions.

   – Choose **Service List** > **Data Lake Insight**. The DLI management console is displayed. If you can view the queue list on the **Queue Management** page but cannot buy DLI queues by clicking **Buy Queue** in the upper right corner (assume that the current permission contains only **DLI ReadOnlyAccess**), the **DLI ReadOnlyAccess** permission has taken effect.

   – Choose any other service in **Service List**. If a message appears indicating that you have insufficient permissions to access the service, the **DLI ReadOnlyAccess** permission has already taken effect.

12.3 Creating a Custom Policy

Custom policies can be created as a supplement to the system policies of DLI. You can add actions to custom policies. For the actions supported for custom policies, see "Permissions Policies and Supported Actions" in the *Elastic Volume Service API Reference*.

You can create custom policies in either of the following two ways:

- Visual editor: Select cloud services, actions, resources, and request conditions without the need to know policy syntax.
- JSON: Create a policy in the JSON format from scratch or based on an existing policy.

. This section describes common DLI custom policies.

Policy Field Description

The following example assumes that the authorized user has the permission to create tables in all databases in all regions:

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dli:database:create_table"
            ],
            "Resource": [
                "dli:*:*:database:*"
            ]
        }
    ]
}
```

- Version

  1.1 indicates a fine-grained permission policy that defines permissions required to perform operations on specific cloud resources under certain conditions.

- Effect

  The value can be **Allow** and **Deny**. If both **Allow** and **Deny** are found in statements, the **Deny** overrides the **Allow**.

- Action

  Specific operation on a resource. A maximum of 100 actions are allowed.

  NOTE

  - The format is *Service name*:*Resource type*:*Action*, for example, **dli:queue:submit_job**.
  - *Service name*: product name, such as **dli**, **evs**, and **vpc**. Only lowercase letters are allowed. Resource types and operations are not case-sensitive. You can use an asterisk (*) to represent all operations.
  - *Resource type*: For details, see **Table 12-7**.
  - *Action*: action registered in IAM.

- Condition

  Conditions determine when a policy takes effect. A condition consists of a condition key and operator.

  A condition key is a key in the **Condition** element of a statement. There are global and service-level condition keys.

  - Global condition keys (prefixed with **g:**) apply to all actions. For details, see condition key description in Policy Syntax.
  - Service-level condition keys (prefixed with the abbreviation of a service, for example, **dli:**. Currently, DLI does not provide service-level condition keys.) apply only to the actions of the corresponding service.

An operator is used together with a condition key to form a complete condition statement. For details, see **Table 12-4**.

IAM provides a set of DLI predefined condition keys. The following table lists the predefined condition keys of DLI.

Table 12-4 DLI request conditions

Condition Key	Operator	Description
g:CurrentTime	Date and time	Time when an authentication request is received **NOTE** The time is expressed in the format defined by **ISO 8601**, for example, **2012-11-11T23:59:59Z**.
g:MFAPresent	Boolean	Whether multi-factor authentication is used during user login
g:UserId	String	ID of the current login user
g:UserName	String	Current login user
g:ProjectName	String	Project that you have logged in to
g:DomainName	String	Domain that you have logged in to

- Resource

  The format is *Service name*:*Region*:*Domain ID*:*Resource type*:*Resource path*. The wildcard (*) indicates all options. For details about the resource types and path, see **Table 12-7**.

  Example:

  dli:*:*:queue:* indicates all queues.

Creating a Custom Policy

You can set actions and resources of different levels based on scenarios.

1. Define an action.

   The format is *Service name*:*Resource type*:*Action*. The wildcard is *. Example:

Table 12-5 Action

Action	Description
dli:queue:submit_job	Submission operations on a DLI queue
dli:queue:*	All operations on a DLI queue
dli:*:*	All operations on all DLI resource types

For more information about the relationship between operations and system permissions, see **Common Operations Supported by DLI System Policy**.

2. Define a resource.

   The format is *Service name*:*Region*:*Domain ID*:**Resource type:Resource path**. The wildcard (*) indicates all resources. The five fields can be flexibly set. Different levels of permission control can be set for resource paths based on scenario requirements. If you need to set all resources of the service, you do not need to specify this field. For details about the definition of Resource, see **Table 12-6**. For details about the resource types and paths in Resource, see **Table 12-7**.

Table 12-6 Resource

Resource	Description
DLI:*:*:table:databases.dbname.tables.*	DLI, any region, any account ID, all table resources of database **dbname**
DLI:*:*:database:databases.dbname	DLI, any region, any account ID, queue of database **dbname**
DLI:*:*:queue:queues.*	DLI, any region, any account ID, any queue resource
DLI:*:*:jobs:jobs.flink.1	DLI, any region, any account ID, Flink job whose ID is 1

Table 12-7 DLI resources and their paths

Resource Type	Resource Names	Path
queue	DLI queue	queues.queuename
database	DLI database	databases.dbname
table	DLI table	databases.dbname.tables.tbname
column	DLI column	databases.dbname.tables.tbname.columns.colname
jobs	DLI Flink job	jobs.flink.jobid
resource	DLI package	resources.resourcename
group	DLI package group	groups.groupname
datasource auth	DLI cross-source authentication information	datasourceauth.name

3. Combine all the preceding fields into a JSON file to form a complete policy. You can set multiple actions and resources. You can also create a policy on the visualized page provided by IAM. For example:

   The authorized user has the permission to create and delete any database, submit jobs for any queue, and delete any table under any account ID in any region of DLI.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": " Allow",
            "Action": [
                  "dli:database:create_database",
                "dli:database:drop_database",
                  "dli:queue:submit_job",
                  "dli:table:drop_table"
            ],
            "Resource": [
                  "dli:*:*:database:*",
                "dli:*:*:queue:*",
                "dli:*:*:table:*"
            ]
        }
    ]
}
```

Example Custom Policies

- Example 1: Allow policies
  - Allow users to create tables in all databases of all regions:

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dli:database:create_table"
            ],
            "Resource": [
                "dli:*:*:database:*"
            ]
        }
    ]
}
```

  - Allow users to query column **col** in the table **tb** of the database **db**:

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dli:column:select"
            ],
            "Resource": [
                "dli:*:*:column:databases.db.tables.tb.columns.col"
            ]
        }
    ]
}
```

- Example 2: Deny policies

  A deny policy must be used in conjunction with other policies to take effect. If the permissions assigned to a user contain both Allow and Deny actions, the Deny actions take precedence over the Allow actions.

- Deny users to create or delete databases, submit jobs (except the default queue), or delete tables.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "dli:database:create_database",
                "dli:database:drop_database",
                "dli:queue:submit_job",
                "dli:table:drop_table"
            ],
            "Resource": [
                "dli:*:*:database:*",
                "dli:*:*:queue:*",
                "dli:*:*:table:*"
            ]
        }
    ]
}
```

- Deny users to submit jobs in the demo queue.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "dli:queue:submit_job"
            ],
            "Resource": [
                "dli:*:*:queue:queues.demo"
            ]
        }
    ]
}
```

12.4 DLI Resources

A resource is an object that exists within a service. You can select DLI resources by specifying their paths.

Table 12-8 DLI resources and their paths

Resource Type	Resource Names	Path
queue	DLI queue	queues.queuename
database	DLI database	databases.dbname
table	DLI table	databases.dbname.tables.tbname
column	DLI column	databases.dbname.tables.tbname.columns.colname
jobs	DLI Flink job	jobs.flink.jobid
resource	DLI package	resources.resourcename

Resource Type	Resource Names	Path
group	DLI package group	groups.groupname
datasourceauth	DLI cross-source authentication information	datasourceauth.name

12.5 DLI Request Conditions

Request conditions are useful in determining when a custom policy takes effect. A request condition consists of a condition key and operator. Condition keys are either global or service-level and are used in the Condition element of a policy statement. Global condition keys (starting with **g:**) are available for operations of all services, while service-level condition keys (starting with a service name such as **dli**) are available only for operations of a specific service. An operator is used together with a condition key to form a complete condition statement.

IAM provides a set of DLI predefined condition keys. The following table lists the predefined condition keys of *DLI*.

Table 12-9 DLI request conditions

Condition Key	Operator	Description
g:CurrentTime	Date and time	Time when an authentication request is received **NOTE** The time is expressed in the format defined by **ISO 8601**, for example, **2012-11-11T23:59:59Z**.
g:MFAPresent	Boolean	Whether multi-factor authentication is used during user login
g:UserId	String	ID of the current login user
g:UserName	String	Current login user
g:ProjectName	String	Project that you have logged in to
g:DomainName	String	Domain that you have logged in to

12.6 Common Operations Supported by DLI System Policy

Table 12-10 lists the common operations supported by each system policy of DLI. Choose proper system policies according to this table. For details about the SQL statement permission matrix in DLI in terms of permissions on databases, tables, and roles, see **SQL Syntax of Batch Jobs** > **Data Permissions Management** > **Data Permissions List** in the *Data Lake Insight SQL Syntax Reference*.

Table 12-10 Common operations supported by each system policy

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
Queue	DROP_QUEUE	Deleting a queue	√	×	√	√
	SUBMIT_JOB	Submitting the job	√	×	√	√
	CANCEL_JOB	Terminating the job	√	×	√	√
	RESTART	Restarting a queue	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	GRANT_PRIVILEGE	Granting permissions to the queue	√	×	√	√
	REVOKE_PRIVILEGE	Revoking permissions from the queue	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	SHOW_PRIVILEGES	Viewing the queue permissions of other users	√	×	√	√
Database	DROP_DATABASE	Deleting a database	√	×	√	√
	CREATE_TABLE	Creating a table	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	CREATE_VIEW	Creating a view	√	×	√	√
	EXPLAIN	Explaining the SQL statement as an execution plan	√	×	√	√
	CREATE_ROLE	Creating a role	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	DROP_ROLE	Deleting a role	√	×	√	√
	SHOW_ROLES	Displaying a role	√	×	√	√
	GRANT_ROLE	Binding a role	√	×	√	√
	REVOKE_ROLE	Unbinding the role	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	SHOW_USERS	Displaying the binding relationships between all roles and users	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	GRANT_PRIVILEGE	Granting permissions to the database	√	×	√	√
	REVOKE_PRIVILEGE	Revoking permissions to the database	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	SHOW_PRIVILEGES	Viewing database permissions of other users	√	×	√	√
	DISPLAY_ALL_TABLES	Displaying table information in the database	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	DISPLAY_DATABASE	Displaying database information	√	×	√	√
	CREATE_FUNCTION	Creating a function	√	×	√	√
	DROP_FUNCTION	Deleting a function	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	SHOW_FUNCTIONS	Displaying all functions	√	×	√	√
	DESCRIBE_FUNCTION	Displaying function details	√	×	√	√
Table	DROP_TABLE	Deleting a table	√	×	√	√
	SELECT	Querying a table	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	INSERT_INTO_TABLE	Inserting	√	×	√	√
	ALTER_TABLE_ADD_COLUMNS	Adding a column	√	×	√	√
	INSERT_OVERWRITE_TABLE	Rewriting	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	ALTER_TABLE_RENAME	Renaming a table	√	×	√	√
	ALTER_TABLE_ADD_PARTITION	Adding partitions to the partition table	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	ALTER_TABLE_RENAME_PARTITION	Renaming a table partition	√	×	√	√
	ALTER_TABLE_DROP_PARTITION	Deleting partitions from a partition table	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	SHOW_PARTITIONS	Displaying all partitions	√	×	√	√
	ALTER_TABLE_RECOVER_PARTITION	Restoring table partitions	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	ALTER_TABLE_SET_LOCATION	Setting the partition path	√	×	√	√
	GRANT_PRIVILEGE	Granting permissions to the table	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	REVOKE_PRIVILEGE	Revoking permissions from the table	√	×	√	√
	SHOW_PRIVILEGES	Viewing table permissions of other users	√	×	√	√

Resources	Operation	Description	DLI FullAccess	DLI ReadOnlyAccess	Tenant Administrator	DLI Service Admin
	DISPLAY_TABLE	Displaying a table	√	√	√	√
	DESCRIBE_TABLE	Displaying table information	√	×	√	√

A Change History

Released On	What's New
2022-12-06	● Optimized the structure of the user guide.