

Data Lake Compute

Getting Started

Product Documentation



Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Getting Started

Complete Process for New User Activation

DLC Data Import Guide

Quick Start with Data Analytics in Data Lake Compute

Quick Start with Permission Management in Data Lake Compute

Quick Start with Partition Table

Quick Start with UDFs

Enabling Data Optimization

Cross-Source Analysis of EMR Hive Data

Getting Started

Complete Process for New User Activation

Last updated : 2024-07-31 17:22:46

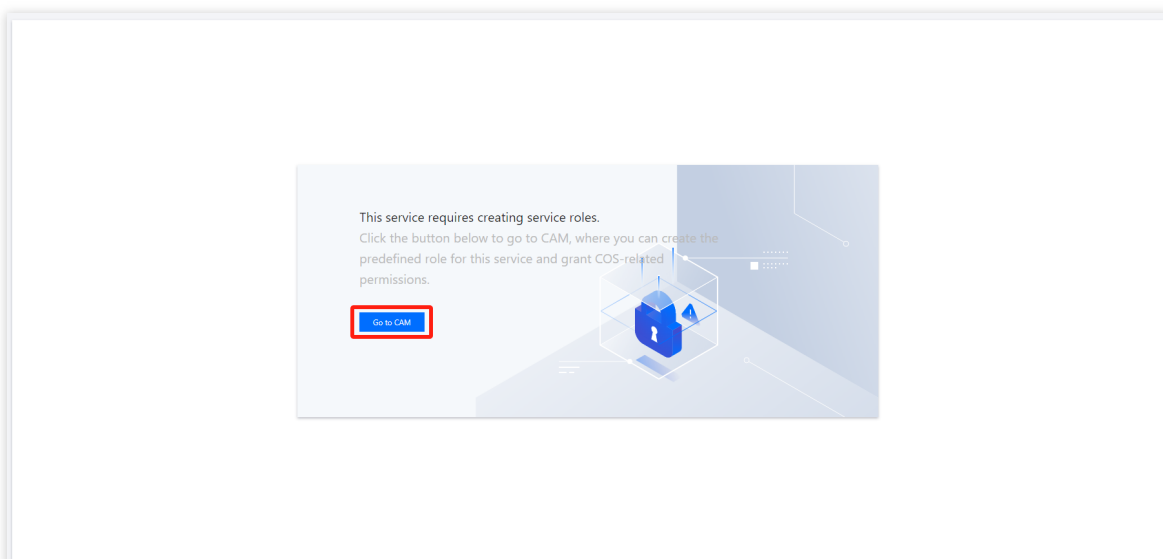
Preparations

Signing up for an account

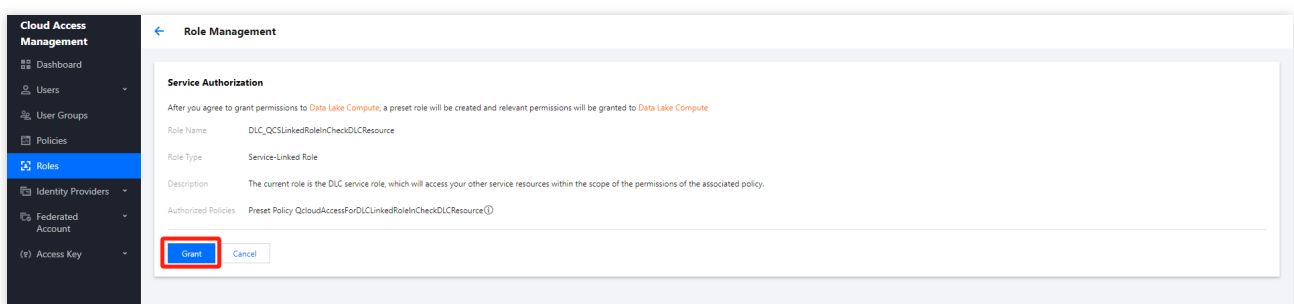
Note:

For administrators only, the following operations may be involved.

1. Enter the [DLC Console](#), click **Go to CAM**, and authorize Data Lake Computing.



2. In Role Management, click **Grant**.



Purchase Engine

Note:**You must have financial permissions in CAM.**

1. On the [DLC console](#), you can go to the engine purchase page from the **Overview** and **Data Engine** pages.

Overview > Initial Configuration > Purchase Data Engine:

Overview Guangzhou

Data Lake Compute overview

Tencent Cloud Data Lake Compute provides agile and efficient data lake analytics and compute services through its serverless architecture. With storage and compute separated, it offers cost-effective options and enables imperceptible resource auto-scaling. It also allows you to use standard SQL statements to perform joint analytics and compute with COS and other cloud data services.

Data management

The data management module of Data Lake Compute allows you to import data from local system or COS and manage (i.e., create/modify/delete) it in a visual interface.

[Create database](#)

Data Explore

Query and visually explore data in a lake using the Tencent Cloud data engines. Support one-click export of results and saving to COS.

[Create query](#)

Data engine

Self-developed data engines that use standard syntax and are compatible with Hive, Spark, and Presto engines. They support auto-scaling and can be settled in various modes to minimize your costs.

[Create engine](#)

Data overview Usage data by 2023-12-05

Total data volume^①	Managed storage usage^①	Private clusters^①	CU usage yesterday^①
1.8T	591.9G	2	0 CUs

Data engine Data of tasks and CUs used by elastic cluster resources by 2023-12-05

[Create compute engine](#)

Private data engines ^①	Engines to expire in 7 days ^①	Isolated engines ^①	Tasks in the last 7 days ^①	Average task time in the last 7 days ^①	CUs used by elastic resources in the last 7 days ^①
2	0	0	0	0	0

Data Engine > Create Resource:

Data Lake Compute Guangzhou Data engine guide

Data engine Network configuration Cluster monitor

[Create resource](#) [All query](#) [Renewal management](#)

Select a resource tag or enter keyword(s) (separate two)

Resource name/ID	Engine type	Kernel version ^①	Running sta...	Billing mode	Auto-renewal	Start and stop policy	Cluster description	Cluster spec	Network configuration	Operation
...	Monitor Spec configuration Parameter Configuration More
...	Monitor Spec configuration Parameter Configuration More

Total items: 2

10 / page 1 / 1 page

2. In the purchase page, select the Engine Type you want to buy:

Note:

SparkSQL: It is suitable for stable and efficient offline SQL tasks.

Spark job: It is suitable for native Spark stream/batch data job processing.

Presto: It is suitable for agile and fast interactive query and analysis.

The screenshot shows the 'Data Lake Compute' configuration page. It includes sections for 'Engine edition' (SuperSQL engine, Standard engine), 'Billing mode' (Pay-as-you-go, Monthly subscription), 'Region' (various global locations with Guangzhou selected), and 'Cluster configuration' (Basic configuration, Compute engine type: SparkSQL, Spark job, Presto). A note at the bottom states: 'This is a memory engine for distributed SQL query. It supports real-time data write to SQL and real-time result return in Data Explore. It is suitable for applications with small loads. It runs faster than a SparkSQL engine.'

Note:

A cluster size of 6CUs is relatively small, advisable only for testing scenarios. For actual production scenarios, it is recommended to select clusters with 64CUs or more.

Engine operation permissions are granted automatically

DLC supports default enablement of engine operation class permissions. Once enabled, all users will by default have the following permissions for that engine:

Utilize: Execute tasks using this engine.

Operation: Initiation of engine suspension or standby.

Monitoring: Administration of engine usage monitoring.

Note:

1. Upon termination, administrators inherently maintain all engine privileges. Ordinary users require an administrator to add permissions on the permission management page.
2. Existing ordinary user permissions will remain intact and can be deleted on the [Permission Management](#) page.
3. Subsequent newly created ordinary users have no usage rights, which should be manually added on the [Permission Management](#) page.

How do I enable or disable the self-delivery authorization engine

By default, the engine enables/disables two operation permission entries:

Access 1: [Engine Purchase page](#) > [Advanced Configuration Items](#)

Timing policy ☐

If this option is enabled, start and suspension time can be set for a data engine, and the cluster status will be changed automatically based on the policy. If not set.

Advanced configuration

IP range of cluster

Auto-granting of engine permissions ☒

If this option is enabled, all users are granted the following permissions on this engine:

- USE: Use this engine to execute tasks
- OPERATE: Pause or suspend the engine
- MONITOR: Monitor and maintain the engine based on its usage

For more engine permissions, see [here](#)

Access 2: Go to the [Data Engine](#) page and click Edit Auto-granting of engine permissions.

Tencent Cloud Overview Products +

SuperSQL engine Hong Kong

Data Lake Compute offers both public and private data engines. A public data engine is managed by Data Lake Compute and billed by scanned data volume, which can be billed on a pay-as-you-go basis or subscribed monthly. For more billing info, see [Billing Overview](#). A pay-as-you-go data engine can be configured with no fees charged on it after suspension. For operations and notes, see [Managing Private Data Engines](#).

[Create resource](#) [Bill query](#) [Renewal management](#) Select

Engine Name/ID	Auto-renewal	Start and stop policy	Cluster description	Auto-granting of engine permissions	Engine Size
	No	Manual start, Manual suspension	Private engine	No	16CU Standard 1-2 cluster(s)
	--	Auto-start, Manual suspension	Private engine	No	16CU Standard 1-5 cluster(s)
	--	Manual start, Manual suspension	Public engine	No	--

Total items: 3

After setting engine permissions, click Confirm.

SuperSQL engine Hong Kong

Data Lake Compute offers both public and private data engines. A public data engine is managed by Data Lake Compute and billed on a pay-as-you-go basis; a private data engine can be billed on a pay-as-you-go basis or subscribed monthly. For more billing info, see [Billing](#). For operations and notes, see [Data Lake Compute](#).

[Create resource](#) [Bill query](#) [Renewal management](#)

Engine Name/ID	Auto-renewal	Start and stop policy	Cluster description	Auto-granting
自动化专用常稳拨测_勿用 DataEngine-lwxhwnud	No	Manual start, Manual suspension	Private engine	No
at_data_engine_presto DataEngine-p3d2xfq1	--	Auto-start, Manual suspension	Private engine	No
public-engine DataEngine-public-1313074...	--	Manual start, Manual suspension	Public engine	No

Total items: 3

Team account activation

If you need multi-account collaborative product usage, you can activate it following these suggestions:

1. Permissions are not universal across different regions; separate permissions need to be set for each region.
2. Quick access to permission activation for Data Lake Compute:

To enable a sub-account's access to Data Lake Compute, please go to the [CAM console](#) for configuration.

To grant a sub-account read-write permissions for data and engines within Data Lake Compute, please go to the [DLC console](#) for setup.

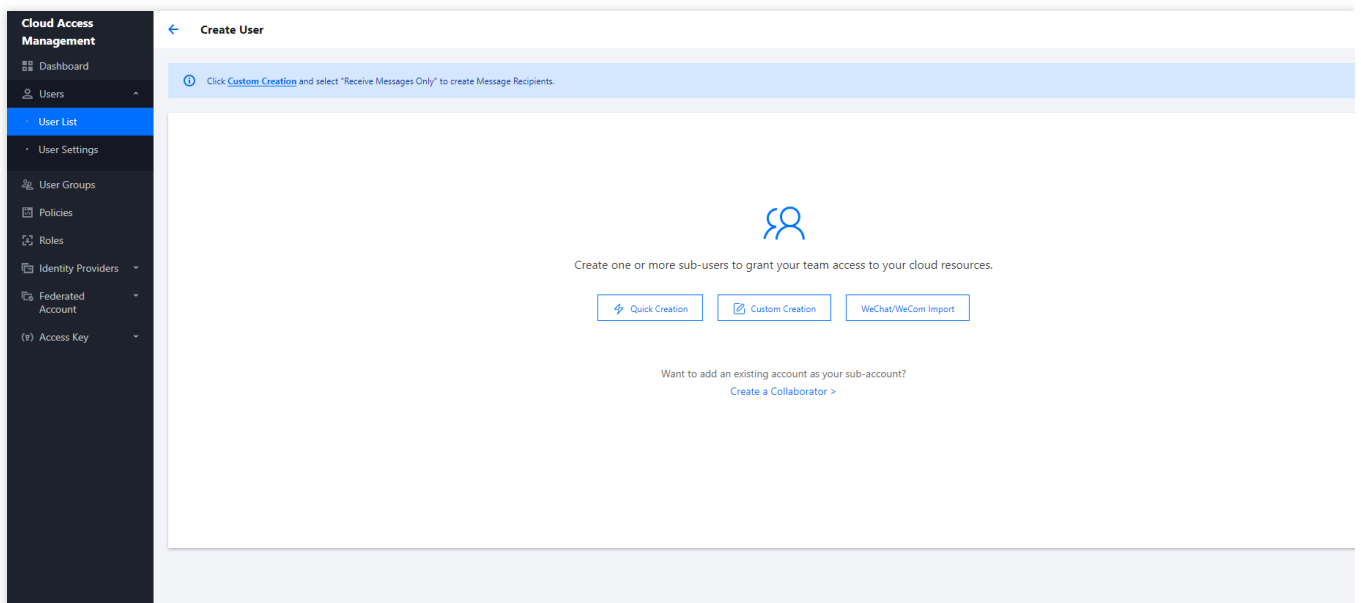
Enable sub-account access to Data Lake Compute permissions

A primary account has all operation permissions for Data Lake Compute by default. The root account grants access permissions of Data Lake Compute to Sub-users through CAM, enabling Sub-users to have corresponding operation

permissions for Data Lake Compute: **QcloudDLCFullAccess**(all operation permissions for Data Lake Compute).

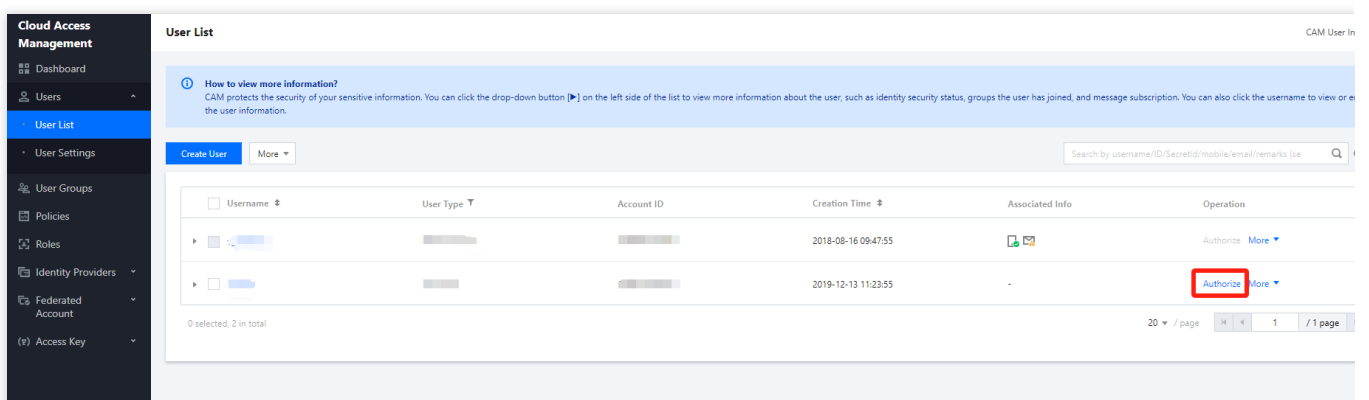
Operation step

1. Log in to the [CAM Console](#) to create Sub-users. For detailed operations, see [Creating and Authorizing Sub-account](#).

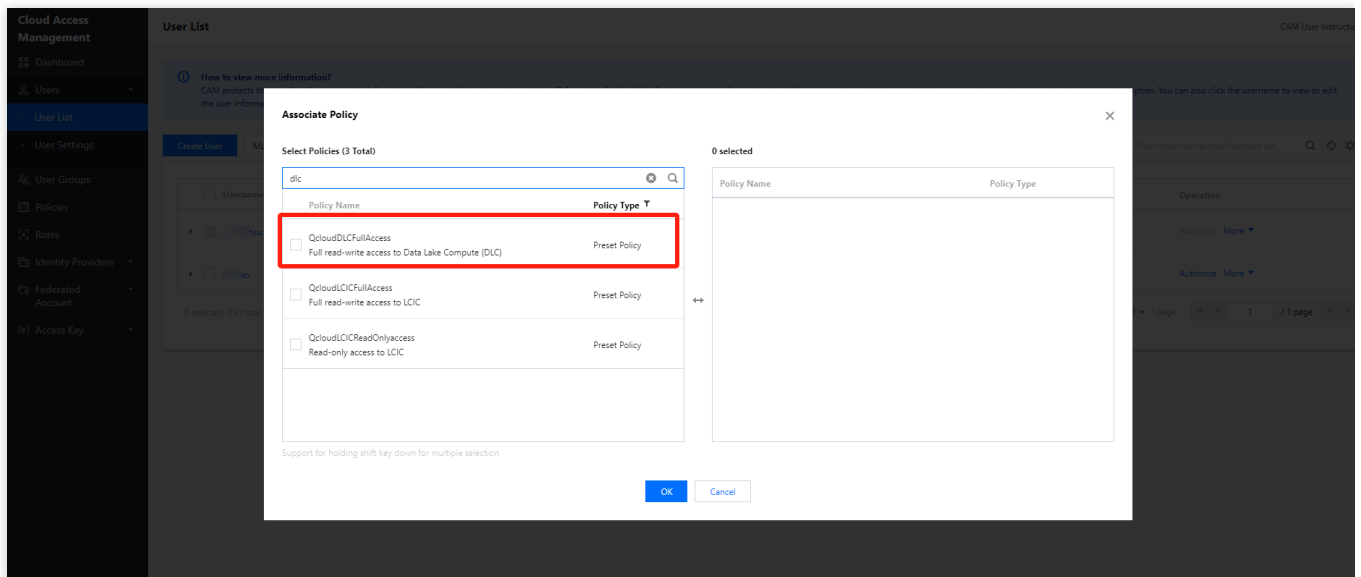


2. Add a preset policy to the sub-account: **QcloudDLCFullAccess**(all operation permissions for Data Lake Compute).

You can search for the user to be authorized in the user list and click **Authorize**.



In the policy list, select **QcloudDLCFullAccess**(all operation permissions for Data Lake Compute).



Enable sub-account access to data and engine permissions in Data Lake Compute

Add a user to Data Lake Compute Permission Management

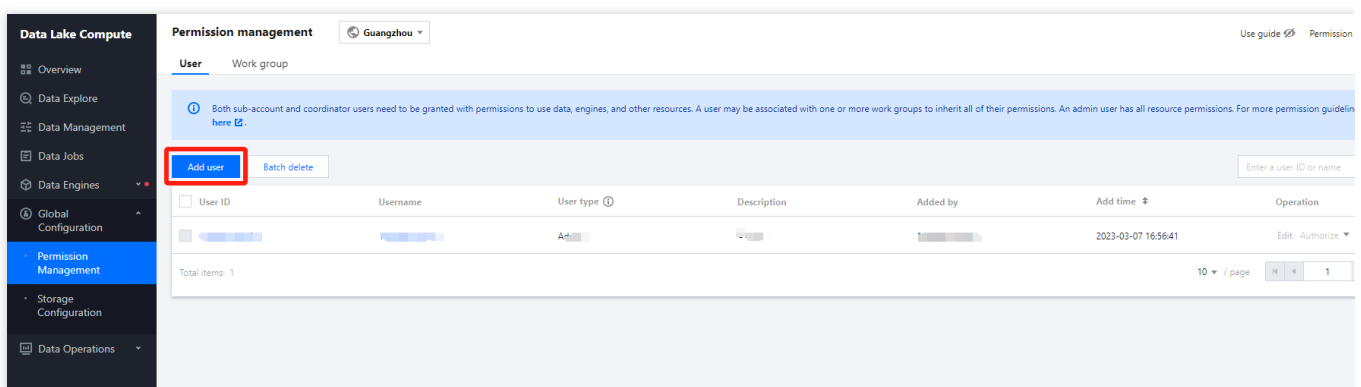
Note:

1. Please confirm the user permission effective region.
2. About User Segmentation:

Administrator: Has permissions for all resources.

Regular User: Needs specific permissions granted or must be assigned to a workgroup to obtain permissions.

1. log in to [DLC Console](#), go to the Permission Management Page, select **Corresponding Service Region**, enter Permission Management Page. click **Add User**.



2. Add the account to DLC for management through the Sub-user's **CAM ID**. Please select the user type as needed.

3. Bind the user to the work group (optional step).

Note:

If you need to manage the permissions of multiple users, you can do so by binding them to a work group. After the work group is created, proceed with the addition. For the specific creation process, please refer to [Quick Start with Permission Management in Data Lake Compute](#).

Add Engine and Data Permissions

After creating a user or work group, click 'Authorization Operation' in the list to add permissions to the work group, including data and engine permissions.

User ID	Username	User type	Description	Added by	Add time	Operation
1			--		2023-12-05 16:49:02	Edit Authorize De
			--		2023-11-22 14:53:11	Edit Data permission Engine permis

Data Permissions

Data Catalog Permissions: Includes the permissions to create databases and data directories under the data catalog.

Add permission ×

Permission type ☒ Catalog ☐ Database & table

The catalog option covers permissions to create databases under DataLakeCatalog and other catalogs, while the database and table option covers permissions of databases, data tables, views, and functions.

Permission ☐ Create database under DataLakeCatalog ☐ Create catalog

Authorizable ☐ Yes

Database Table Permissions: Grants fine-grained permissions at the database and table level, including the inquiry and editing of libraries, tables, views, functions, etc.

Add permission

Permission type
☐ Catalog
☒ Database & table

The catalog option covers permissions to create databases under DataLakeCatalog and other catalogs, while the database and table option covers permissions of databases, data tables, views, and functions.

Catalog
DataLakeCatalog

Setting mode
Standard
Advanced

Database

Select a database/view/function

Enter a database name

☐ All

Selected (0)

Enter a database name

☐ All

Permission
☐ Query analysis ⓘ
☐ Edit data ⓘ
☐ Owned by ⓘ

Select a target permission set. "Query & analytics" and "Data edit" cover the permissions required to analyze or edit selected targets; "Owner" grants the permission to re-authorize permissions in addition to data edit permissions.

Engine permissions

Select engine permission policies according to the usage scenarios of users or work groups.

Note:

Utilize: Execute tasks using this engine.

Modification: Modify the engine's configuration parameters, such as specification configuration.

Operation: Initiation of engine suspension or standby.

Monitoring: Administration of engine usage monitoring.

Delete: Remove the engine.

Authorized: Once selected, all members under this Sub-user or workgroup will have the authorization permissions for the engine.

Add permission

Data engine

Enter

Engine permission

☐ All

☒ Use

☐ Modify

☐ Operation

☒ Monitor

☐ Delete

Authorizable

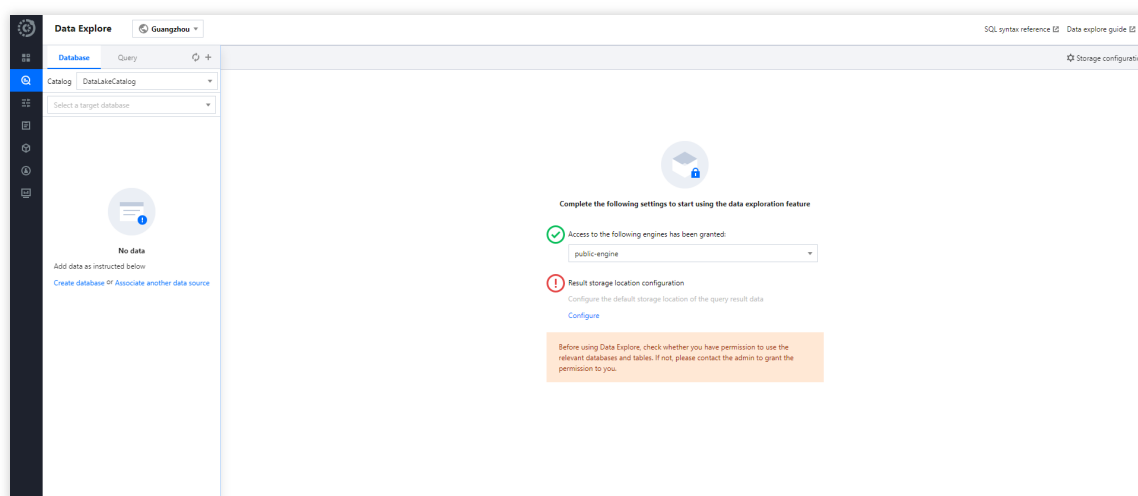
☐ Yes

Configure Result Storage Location

Before using the Data Exploration feature, it is necessary to configure the Query Result Path. Once configured, the query results will be saved to the specified COS Path or DLC's Managed Storage. For detailed steps, please refer to the [Configure Query Result Path Operation Guide](#).

Configure Result Storage Location

Enter the [DLC Console](#), select the Data Exploration feature, choose Result Storage Location Configuration, and click **Click to Configure**.



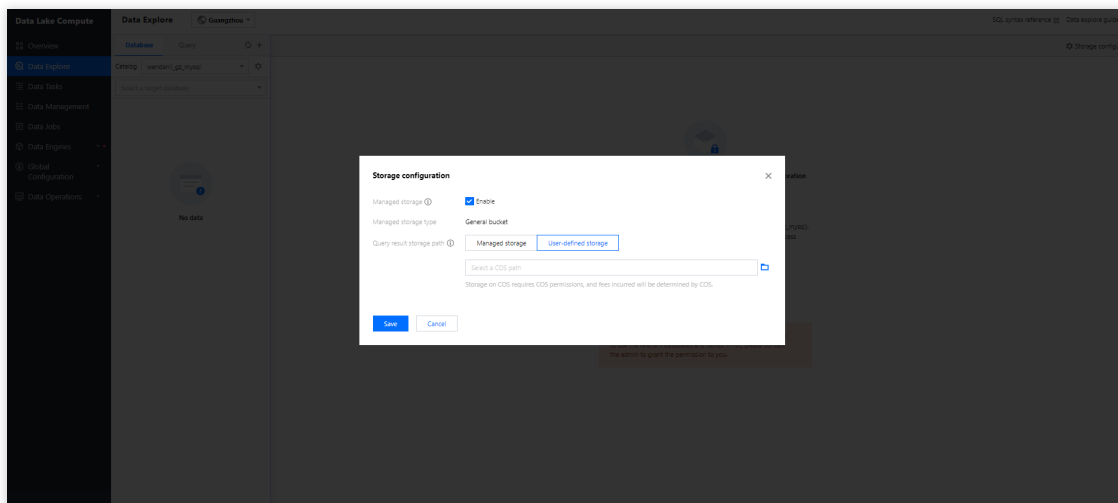
Select the Storage Location and Method

Note:

Metadata Acceleration Bucket: In the current region, it can significantly improve query analysis performance. Inner tables can be enabled directly, while external tables require confirmation that engine permissions allow for it.

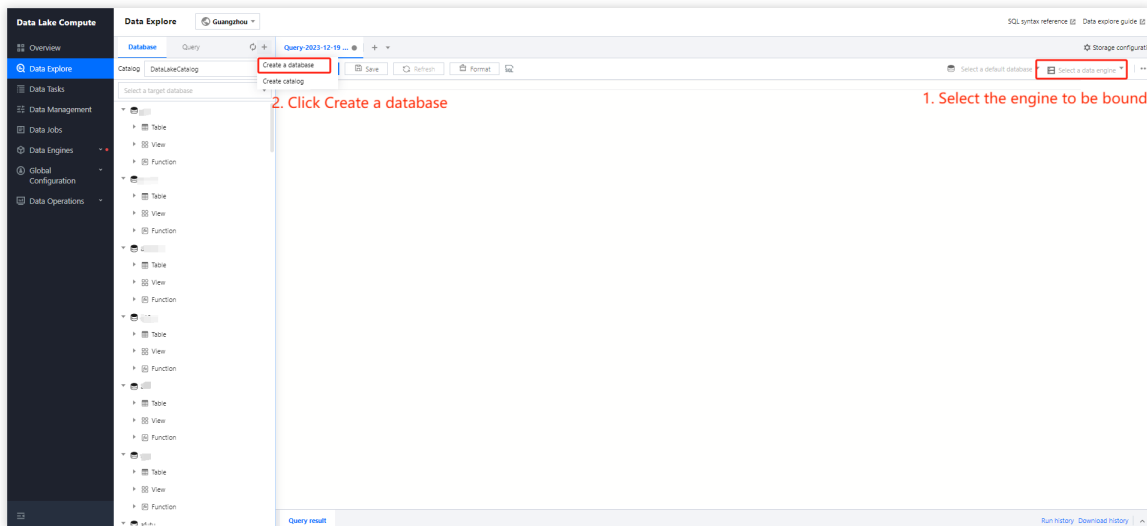
Please Note: Shared Engines cannot be associated with Metadata Acceleration Buckets. When a user selects a User Storage Path, a Dedicated Engine needs to bind to a Metadata Acceleration Bucket first before queries can take effect.

User Storage: User storage refers to your bucket path on COS.

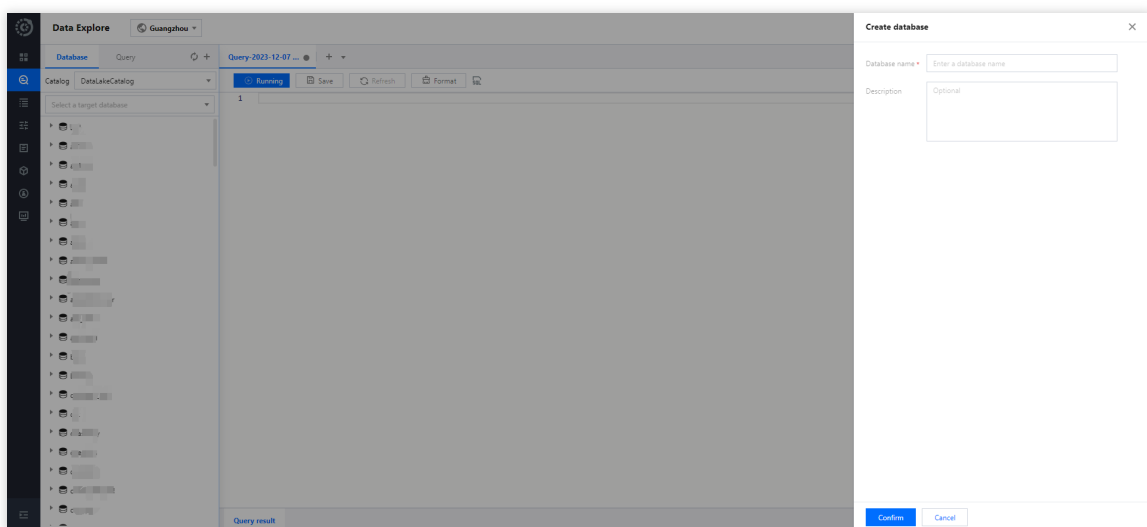


Create Library

Before creating a database, choose the engine type to be used.

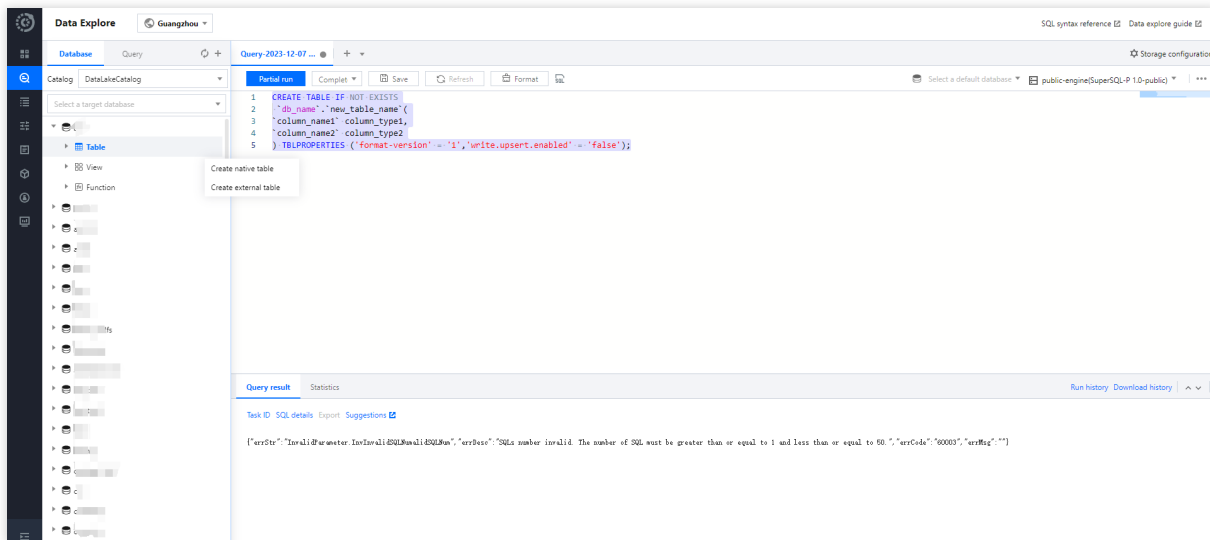


Enter the database name, and click **Confirm**.

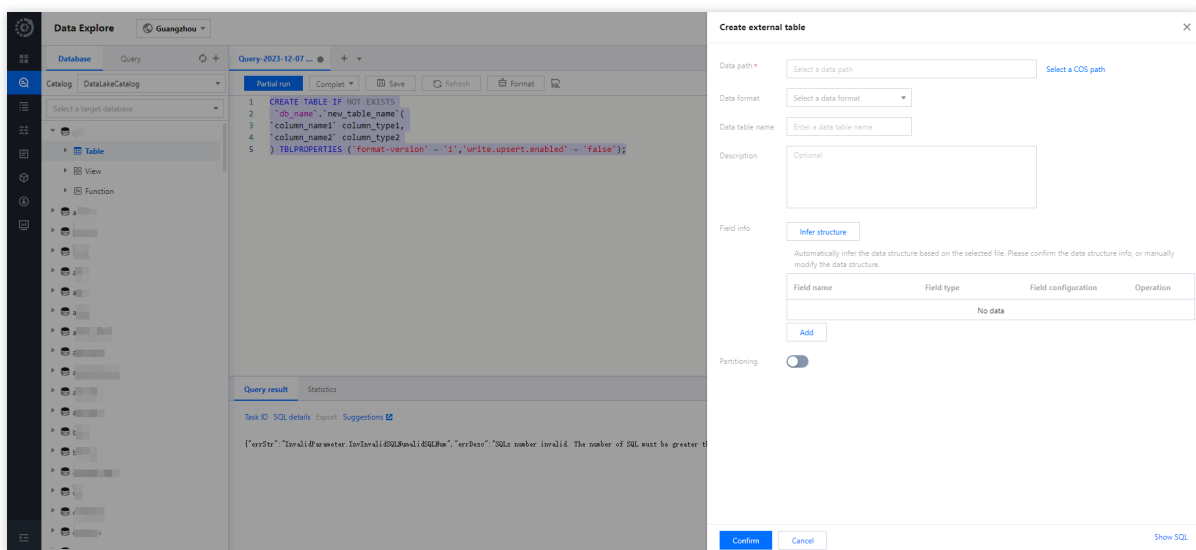


Create a table

Native Table: These are tables stored in your DLC managed storage, by default in Iceberg format. Using native tables eliminates the need to deal with Iceberg's underlying files and offers capabilities such as data optimization to help build your data lake.



External Table: These tables represent data stored in your own COS buckets or other third-party data storage. DLC can directly create external tables for analysis without the need to load additional data.

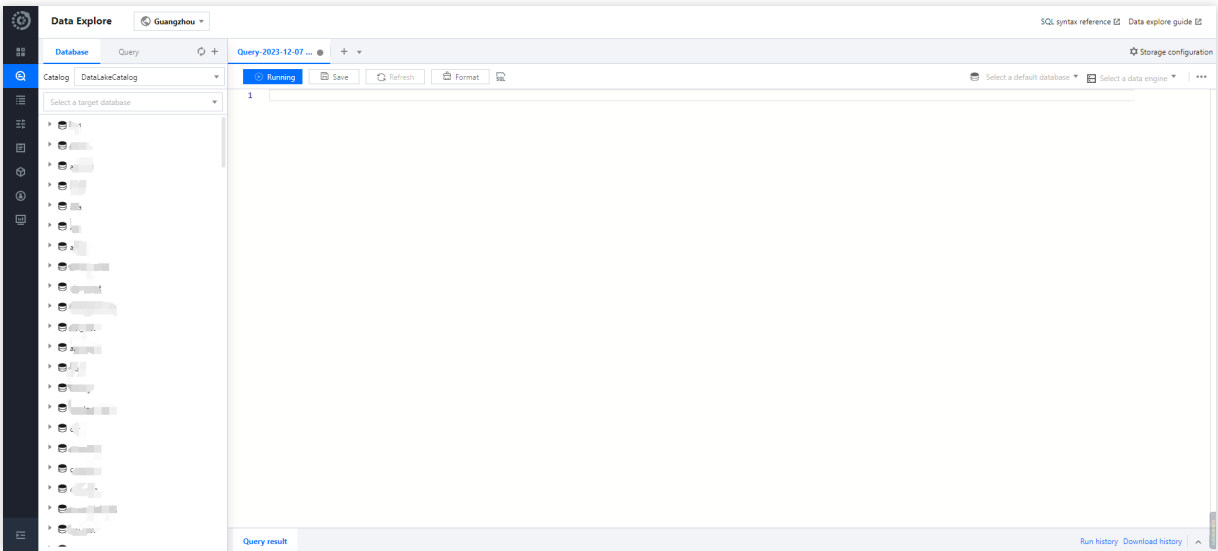


Note:

After creating a Native Table, you need to refresh the browser to use it.

Data Query

Enter the DLC Console - Data Exploration. SQL Queries can be created on the analysis page. Features supported include Run All, Run Partially, Download Results, and materialized views.



DLC Data Import Guide

Last updated : 2024-07-31 17:23:10

External Table Data Import via COS

DLC supports querying and analyzing data directly on COS without migrating the data. Therefore, you only need to import the data into COS to start using DLC for seamless data analysis, achieving complete decoupling of data storage and computation. Currently, it supports uploading in multiple formats such as orc, parquet, avro, json, csv, and text files.

Currently, COS offers a variety of data import methods. You can choose from the following methods based on your situation.

log in to [COS](#) and proceed with file upload directly. For related operating steps, see [Uploading an Object](#).

Import data using various upload tools provided by COS. For a list of supported tools, see [Tool Overview](#).

Import data using SDKs or APIs provided by the COS service. For service-related instructions, see [Upload Interface Documentation](#).

If you need to analyze logs from CLS, you can directly deliver logs to COS by partition and then analyze and query directly through DLC. For related operations, see [Using DLC \(Hive\) to Analyze CLS Logs](#).

If you need to import data from other cloud services (such as database CDB, etc.) into COS, you can use DataInLong to perform the import. When creating a data synchronization link, select the cloud service to export from as the data source and choose COS as the destination to complete the data import.

If you encounter any issues during data import, you can consult us for a solution by [Submitting a Ticket](#).

After importing data into COS, you can perform SQL queries through the DLC console, API, or SDKs, enabling table creation, analysis, and export of results. For detailed operations, see [Quick Start with Data Analytics in Data Lake Compute](#).

Data import into native tables

To provide better data query performance, DLC also supports importing data into native tables for query analysis.

DLC native tables are arranged in the Iceberg table format, optimizing data during the import process. If you have the following use cases, it is recommended to use native tables for data query analysis.

In data warehouse analysis scenarios, aiming to leverage the Iceberg index for better analytical performance.

If there's a need to update data, the DLC service supports performing UPSERT operations through SQL or data jobs.

Data is written or updated in real-time through DataInLong, Flink, SCS, Spark Streaming, with concurrent reads and writes, requiring transactional guarantees for data processing business.

Wishing to utilize Iceberg table features, such as time travel, multi-version snapshots, hidden partitions, partition evolution, and other advanced data lake features.

If you need to import data into a native table, you can choose one of the following methods based on your situation.

Directly import through the [DLC console](#).

Caution

When importing data through the console, there are certain restrictions, mainly for rapid testing and it's not recommended for production use.

If your original data is in services like MySQL or Kafka and you need to write or update MySQL binlog and message middleware data to DLC in near real-time, this can be achieved through DataInLong DataInLong's real-time import capability. Or through SCS, Flink writing. For operational guidance, you can contact us through a [Work Order](#).

If the original data is in data services such as MySQL, Kafka, MongoDB, etc., offline synchronization tasks by DataInLong DataInLong can be used to transfer data to native tables. During the data warehouse modeling process, external tables are used as the source layer of original data. In the process of transferring data to native tables, business-specific data distributions can be reorganized through building sparse indexes, etc., to achieve excellent query analysis performance of native tables. If guidance is needed, you can [Contact Us](#).

Use SQL statements SELECT INSERT to query the data from the external table and then write it into the native table. For example, after creating a native table in DLC with the same table structure as the external table, the transfer can be completed by executing SQL syntax with the SparkSQL engine. Syntax example is as follows:



```
--- External table name: outtertable, Native table name: innertable  
insert into innertable select * from outtertable
```

If you encounter any issues during data import, you can consult us for solutions by [submitting a work order](#).

Multiple data sources federated query analysis

If you do not wish to export data to the native tables of COS or DLC, DLC also offers the capability of data federation query analysis. It supports rapid association and analysis of data from multiple data sources through SQL without relocating data. Currently, it supports a variety of data sources including MySQL, SQLServer, clickhouse, PostgreSQL, EMR on HDFS, and EMR on COS.

When using federated analysis, it is necessary for the data source and data engine to be on the same network, ensuring network connectivity. Management can refer to [Engine Network Configuration](#).

When querying EMR data through DLC federated analysis, the query performance will be on par with or even exceed that of EMR, making it suitable for production environments. It allows for the full utilization of DLC's fully-managed elastic capabilities to reduce costs and increase efficiency without relocating EMR services.

Federated analysis enables quick unification and analysis of data from multiple data sources, providing a convenient method for data insights and rapid analysis. With the support of DLC's fully-managed elastic capabilities, it effectively reduces the cost of use. It also supports the use of INSERT INTO/INSERT OVERWRITE syntax to write federated data into DLC native tables, completing data import.

When analyzing data from other data sources through federated analysis, since the computation process involves synchronizing data to the DLC for analysis, there is some performance loss compared to directly querying the original data sources. If high query performance is required, data can be imported into native tables for analysis. The operation can be seen in Data import into native tables.

Quick Start with Data Analytics in Data Lake Compute

Last updated : 2024-07-17 15:19:00

Data Lake Compute allows you to quickly query and analyze COS data. Currently, CSV, ORC, Parquet, JSON, Avro, and text files are supported.

With Data Lake Compute, you can complete data analysis queries on COS in just a minute. It currently supports multiple formats including CSV, ORC, PARQUET, JSON, ARVO, and text files.

Preliminary Preparations

Before initiating a query, you need to activate the internal permissions of Data Lake Compute and configure the path for query results.

Step 1: Establish the necessary internal permissions for Data Lake Compute.

Note

If the user already has the necessary permissions, or if they are the root account administrator, this step can be disregarded.

If you are logging in as a sub-account for the first time, in addition to the necessary CAM authorization, you also need to request any Data Lake Compute admin or root account admin to grant you the necessary Data Lake Compute permissions from the **Permission Management** menu on the left side of the Data Lake Compute console (for a detailed explanation of permissions, please refer to [DLC Permission Overview](#)).

1. Table Permissions: Grant read and write operation permissions to the corresponding catalog, database, table, and view.
2. Engine Permissions: These can grant usage, monitoring, and modification rights to the computation engine.

Note

The system will automatically provide each user with a shared public-engine based on the Presto kernel, allowing you to quickly try it out without the need to purchase a private cluster first.

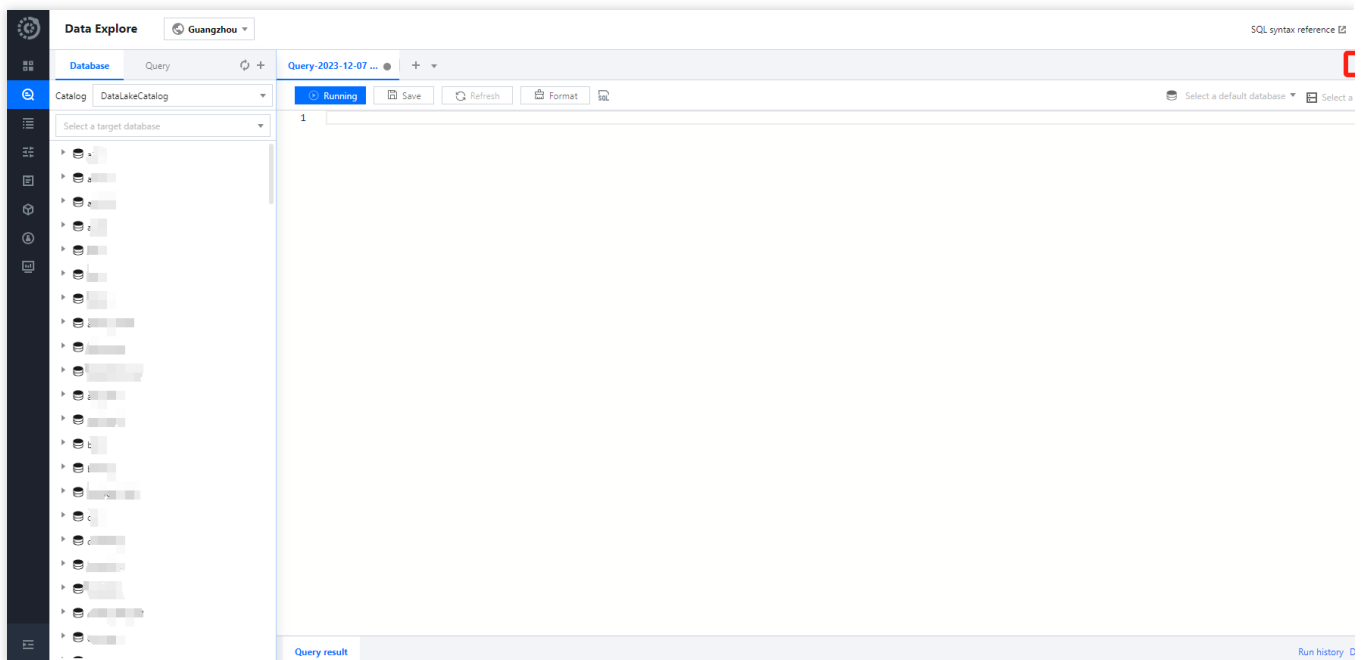
For detailed steps on granting permissions, please refer to [Sub-account Permission Management](#).

Step 2: Configure the path for query results.

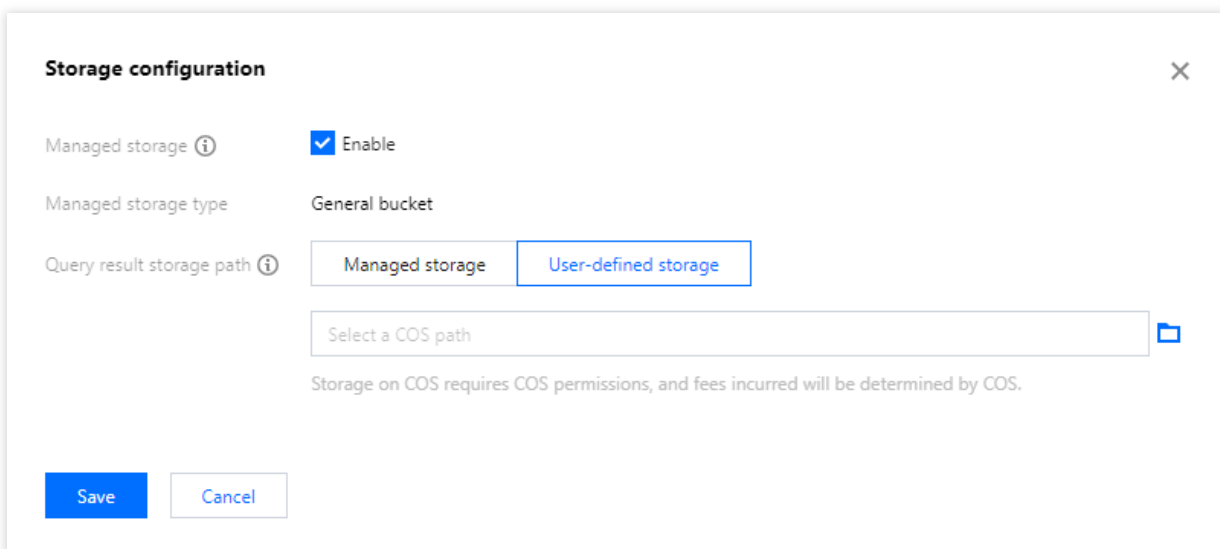
Upon initial use of Data Lake Compute, you must first configure the path for query results. Once configured, the query results will be saved to this COS path.

1. Log in to the [Data Lake Compute DLC console](#) and select the **service region**.
2. Navigate to **Data Exploration** via the left sidebar menu.

3. Under the **Database and Tables** page, click on **Storage Configuration** to set the path for query results.



Specify the COS path for storage. If there are no available COS buckets in your account, you can create one through the [Object Storage Console](#).

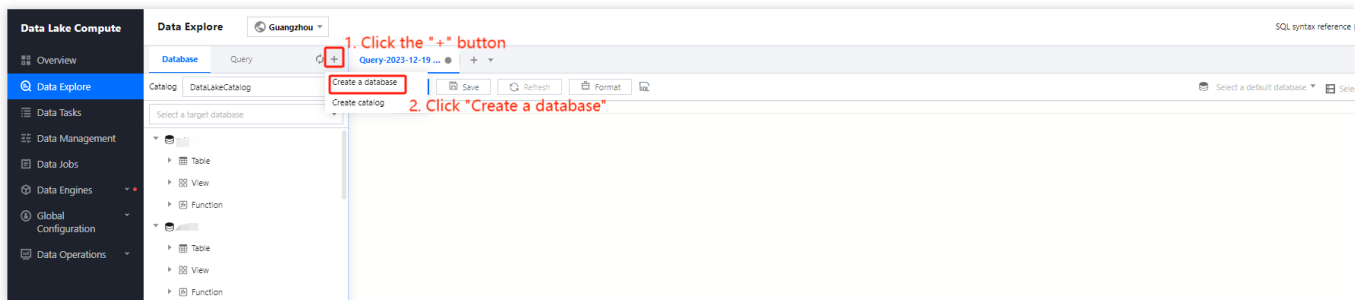


Analysis Steps

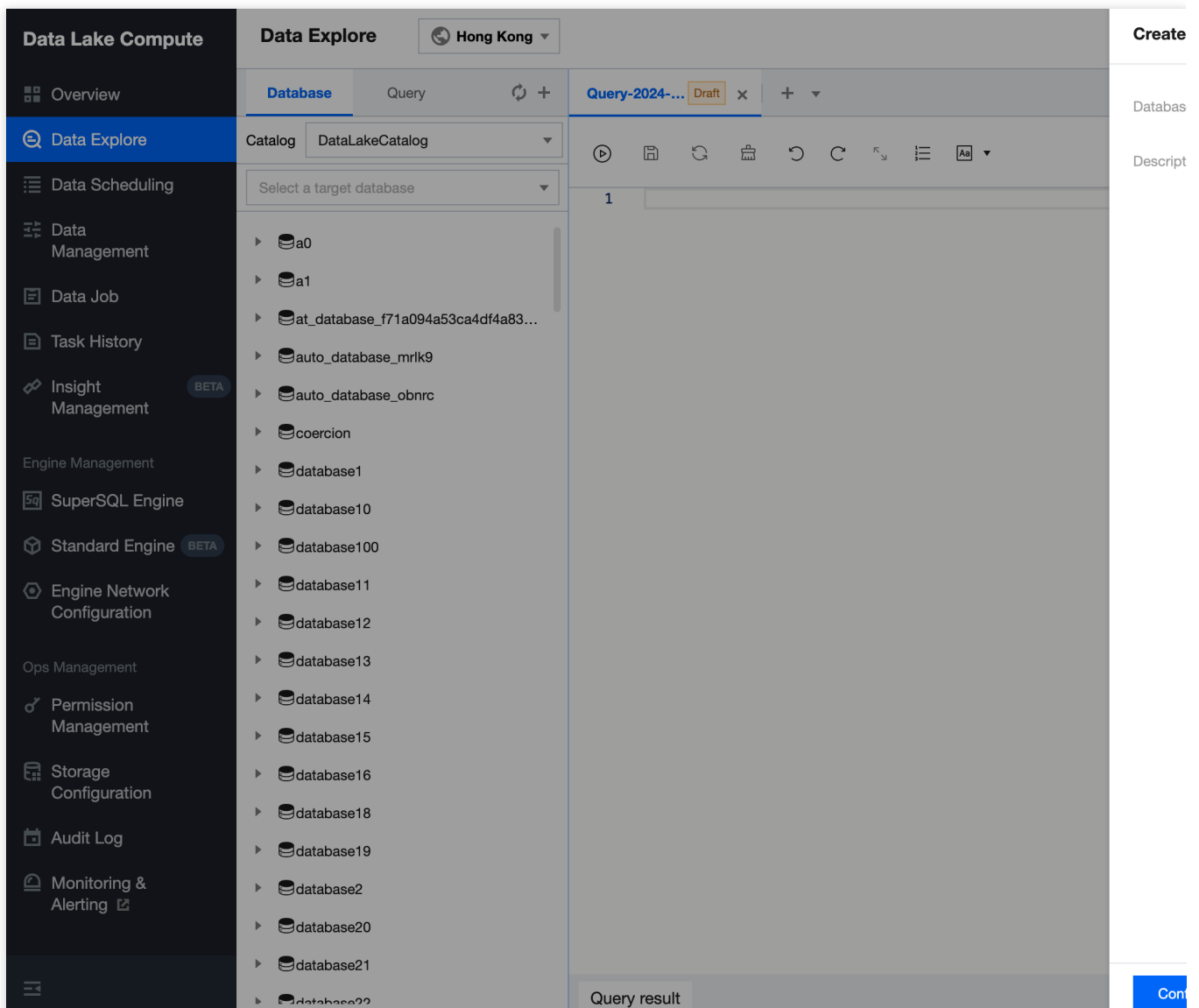
Step 1. Create a database

If you are familiar with SQL statements, write the `CREATE DATABASE` statement in the query and skip the creation wizard.

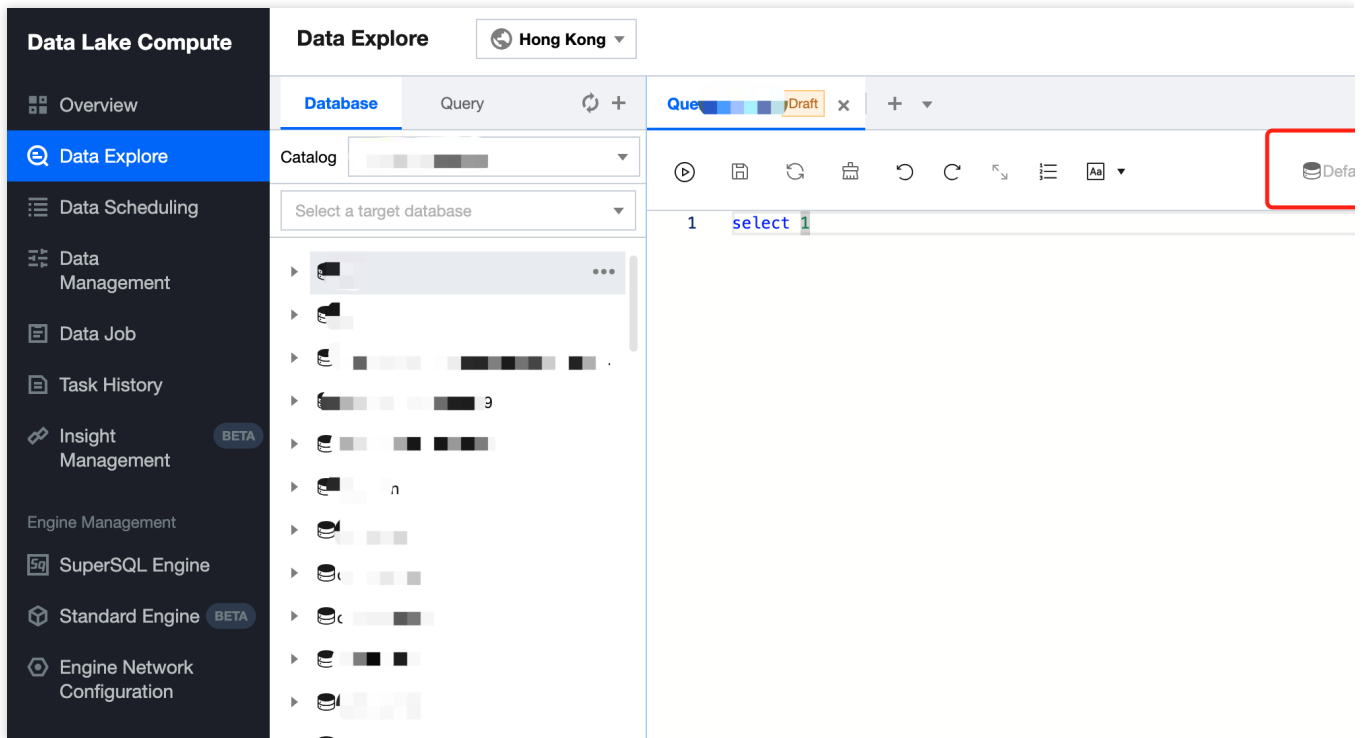
1. Log in to the [Data Lake Compute console](#) and select the service region.
2. Select **Data Explore** on the left sidebar.
3. Select **Database & table**, click "+", and select **Create a database** as shown below:



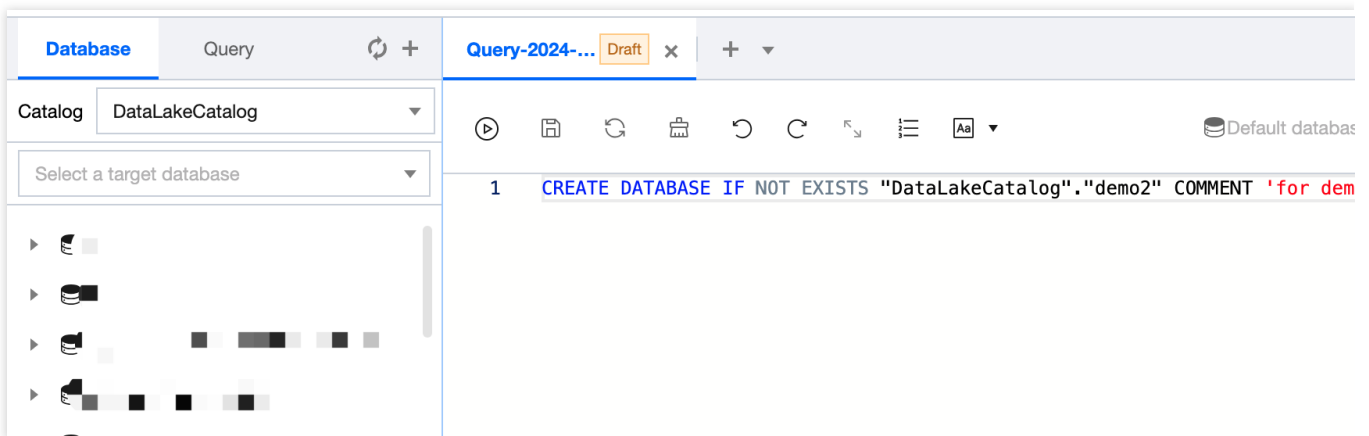
Enter the database name and description.



4. After selecting an execution engine in the top-right corner, run the `CREATE DATABASE` statement.



As shown in the picture below:



For details, see [Table Management](#).

Step 2. Create an external table

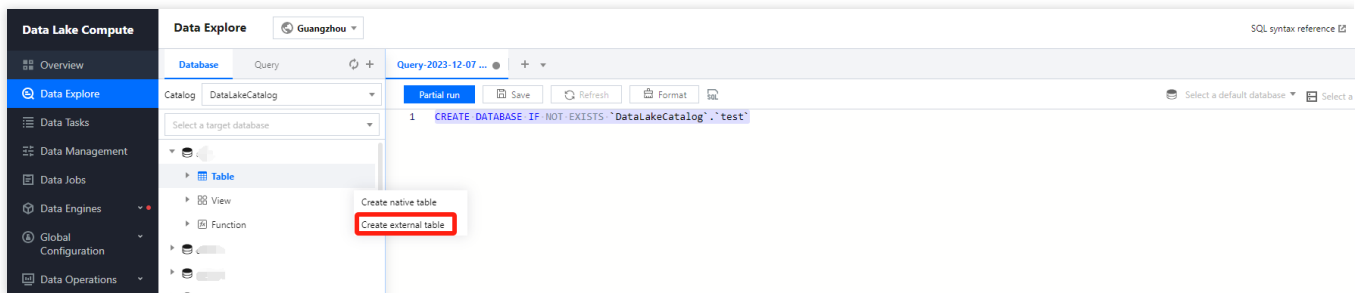
If you are familiar with SQL statements, write the `CREATE TABLE` statement in the query and skip the creation wizard.

1. Log in to the [Data Lake Compute console](#) and select the service region.
2. Select **Data Explore** on the left sidebar.

3. Select **Database & table**, select the created table, and right-click to select **Create external table**.

Note:

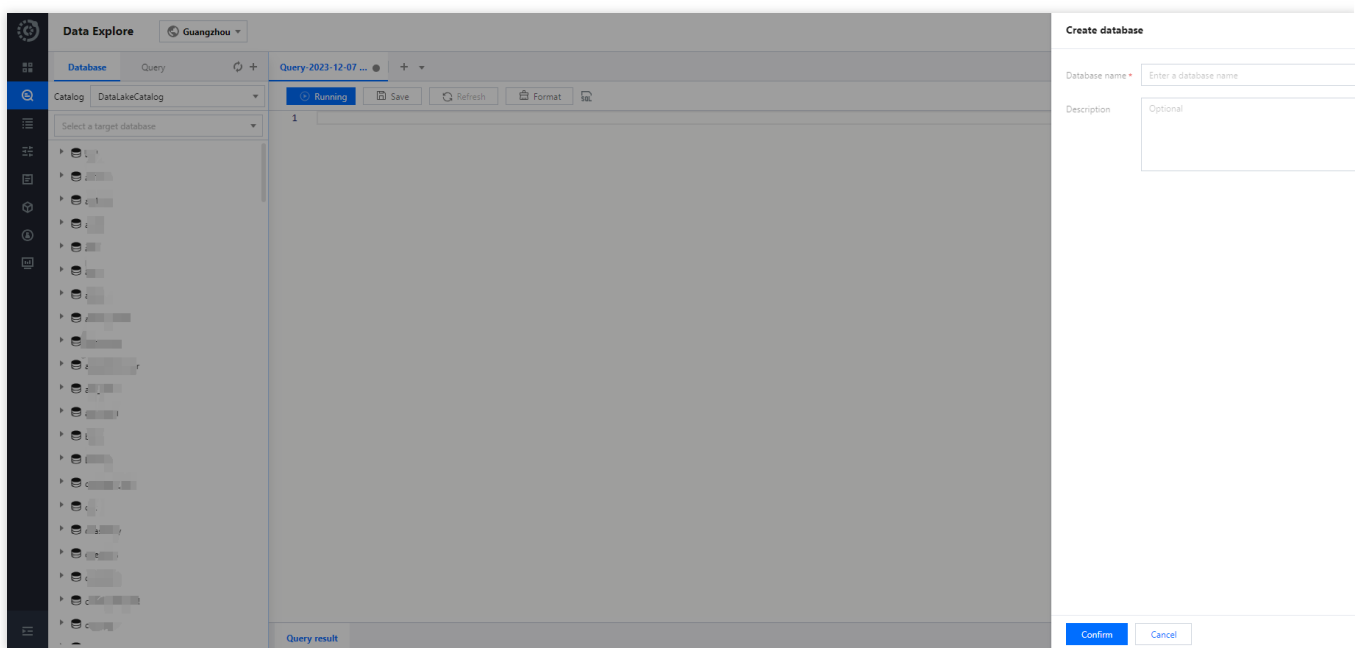
An external table generally refers to a data file stored in a COS bucket under your account. It can be directly created in Data Lake Compute for analysis with no need to load additional data. It is external, so only its metadata will be deleted when you run `DROP TABLE`, while your original data will remain.



4. Generate the table creation statement based on the wizard, and then complete the steps of setting the basic information, selecting the data format, editing the column, and editing the partition.

Step 1. Select the COS path of the data file (which must be a directory in a COS bucket but not a bucket itself). There is also a quick method to upload a file to COS. The operations require relevant COS permissions.

Step 2. Select the data file format. In the **Advanced options**, you can select automatic inference, and then the backend will parse the file format and automatically generate the table column information for fast column inference.



Note:

Structure inference is an auxiliary tool for table creation and may not be 100% accurate. You need to check and modify the field names and types as needed.

Create external table

Data path *

Select a data path

Select a COS path

Data format

Select a data format

Text file (Log, TXT, and others)

CSV

JSON

PARQUET

ORC

AVRO

Data table name

Description

Field info

Infer structure

Automatically infer the data structure based on the selected file. Please confirm the data structure info, or manually modify the data structure.

Field name	Field type	Field configuration	Operation
No data			

Add

Partitioning

☐

Confirm

Cancel

Show SQL

Step 3. Skip this step if there is no partition. Proper partitioning helps improve the analysis performance. For more information on partitioning, see [Querying Partition Table](#).

Partitioning

☒

Partition field	Partition type	Operation
<div>Enter</div>	<div>Select</div>	<div>Insert Delete</div>

Add

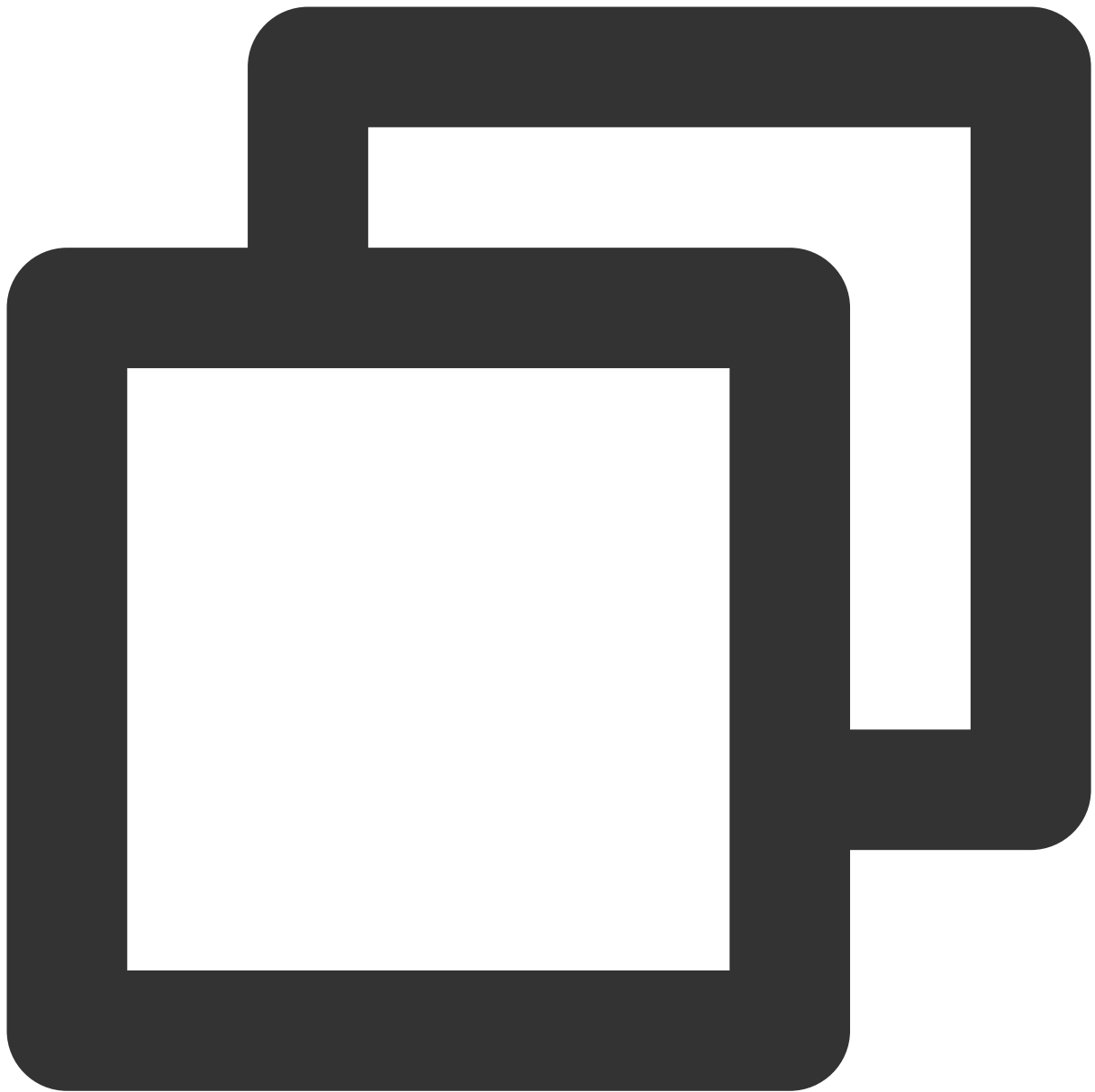
5. Click **Complete** to generate the SQL statement for table creation. Then, select a data engine and run the statement to create a table.



After the data is prepared, write the SQL analysis statement, select an appropriate compute engine, and start data analysis.



Write a SQL statement with all data query results being `SUCCESS` and run the statement after selecting a compute engine.



```
select * from `DataLakeCatalog`.`demo2`.`demo_audit_table` where _c5 = 'SUCCESS'
```

Quick Start with Permission Management in Data Lake Compute

Last updated : 2024-07-17 15:24:34

During the utilization of Data Lake Compute (DLC), if you need to establish varying access permissions for employees within your organization to achieve isolation of authority among them, you can employ the permissions management feature for meticulous management of user and workgroup permissions.

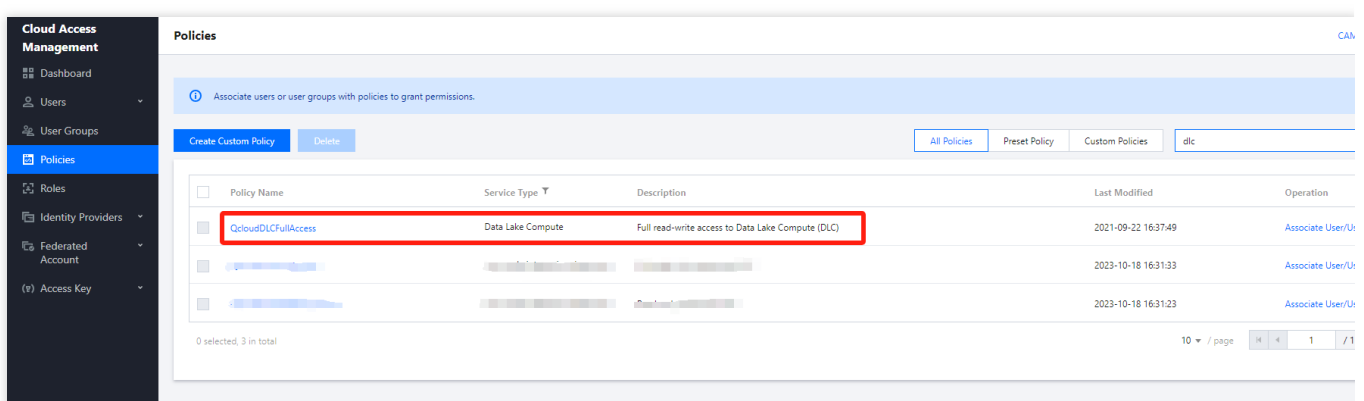
Note :

1. The policy of permissions is highly correlated with the usage of the product. It is recommended that administrators configure the policies for roles such as workgroups and sub-users in advance before officially utilizing the product features.
2. In different regions, administrators are required to reconfigure the member management and permissions management for DLC in that specific region.

CAM Authorization

Data Lake Compute (DLC) possesses a comprehensive data access permission mechanism. If you have sub-account management requirements, please grant the corresponding sub-account with the QcloudDLCFullAccess (Full read-write access to Data Lake Compute (DLC)) policy in the [Access Management Console](#). For specific steps on creating sub-accounts and authorizing policies.

Data Lake Compute (DLC) offers permissions refined to the granularity of row and column levels in data tables, ensuring that you need not worry about overstepping authority with this operation.

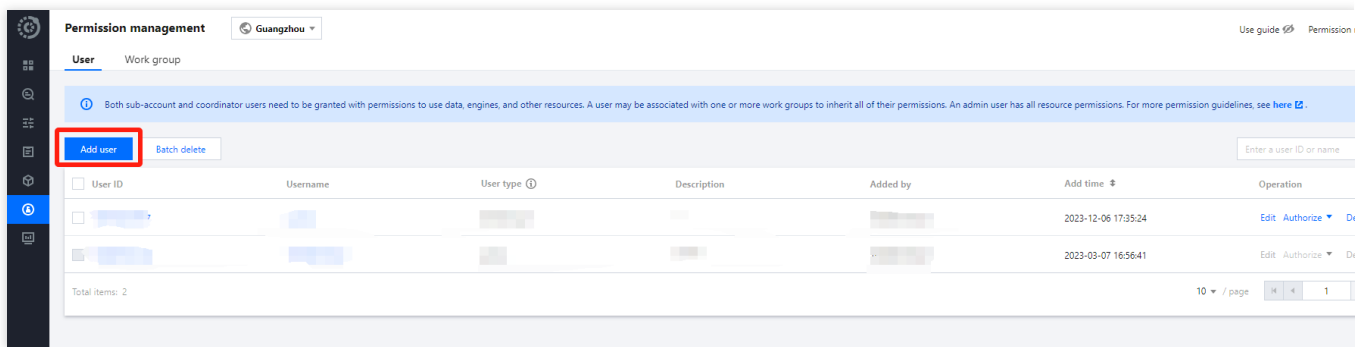


Adding a User

Data Lake Compute utilizes the Tencent Cloud account ID as the default user ID. It distinguishes between two user types: administrators and ordinary users. Administrators inherently possess all resource permissions, while ordinary users must be granted specific permissions or be associated with a work group to acquire permissions.

1. Incorporate a user and associate them with a work group.

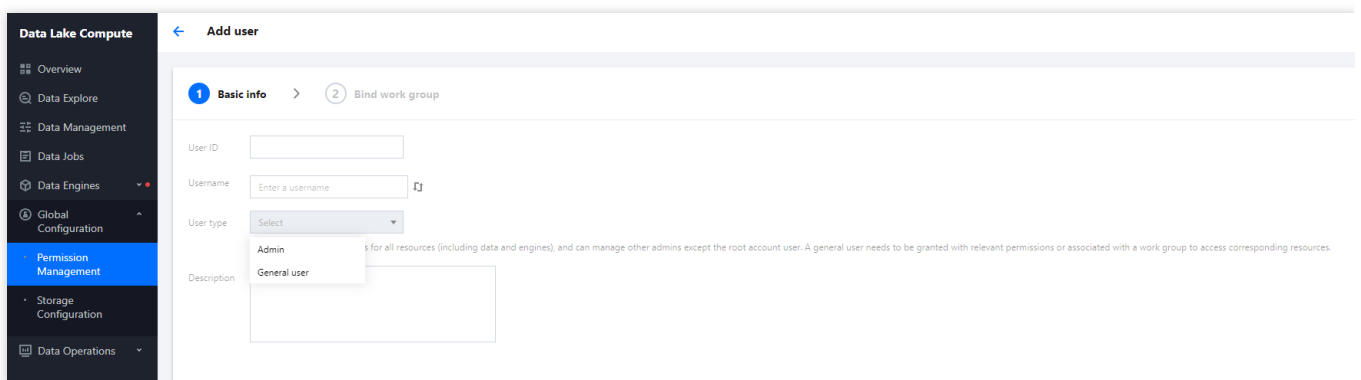
Log into the DLC console, select [Permission Management](#), and click on Users > Add User to incorporate a new user.



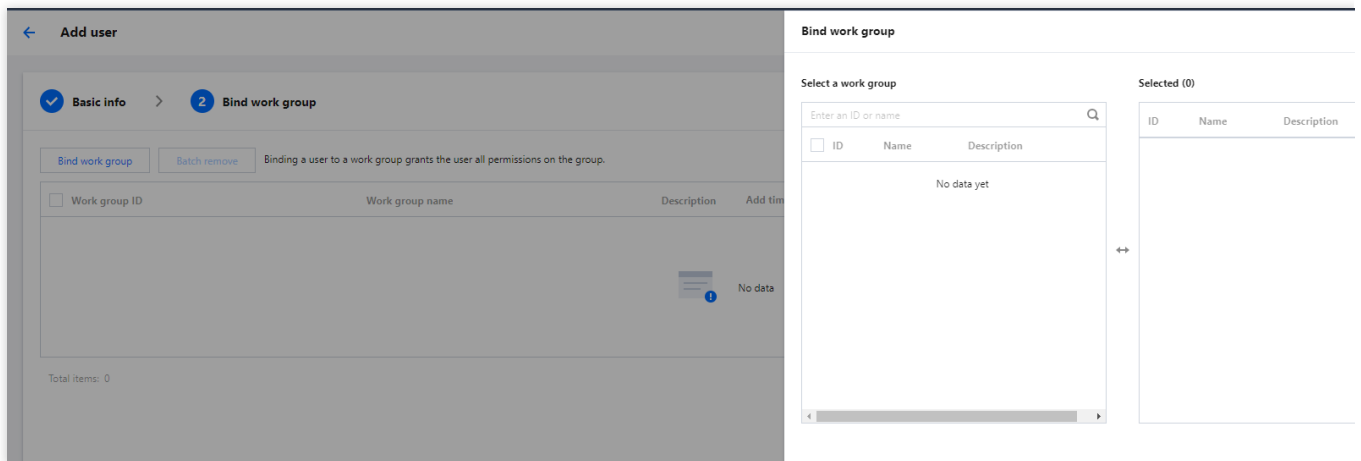
2. Enter the basic information: Provide the user ID, user name, and description, and select the user type.

Note :

When selecting the user type as "Ordinary User", permissions can be obtained through individual authorization or by acquiring all permissions of a specified work group. When selecting "Administrator" as the user type, there is no need to associate with a work group to gain all permissions.

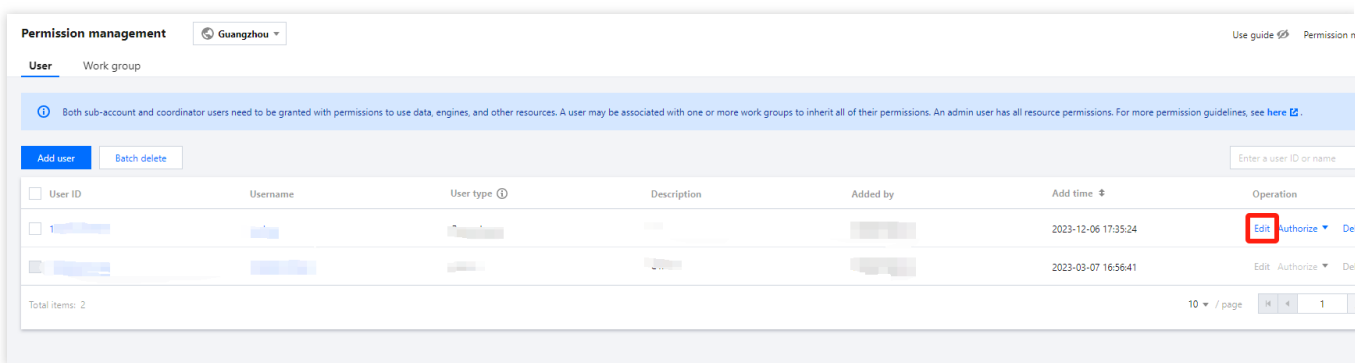


3. Associate with a work group: Select a work group for association (optional).



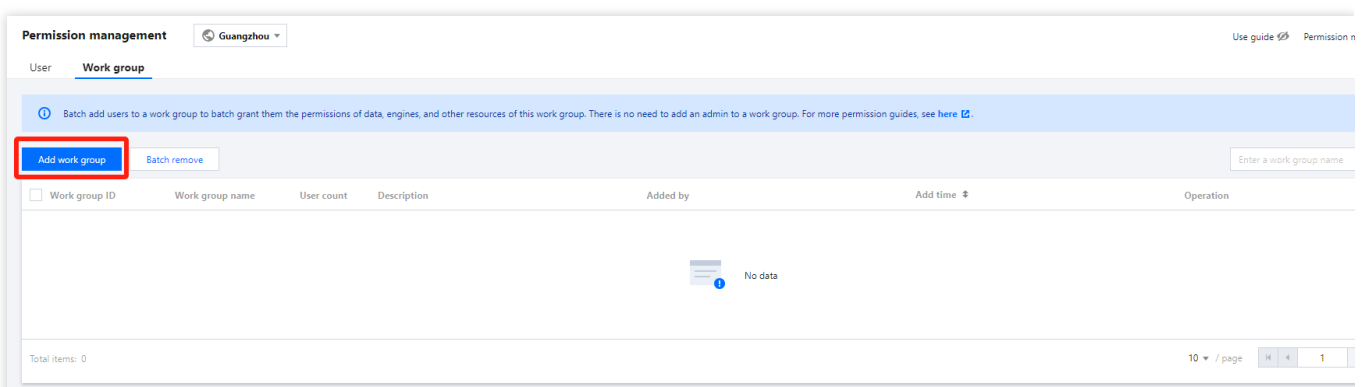
User authorization

In the user list, authorize each user individually. The authorization includes "Data Permissions" and "Engine Permissions", and the permission policy is consistent with the work group's permission policy.



Add Work Group

1. In the Data Lake Compute DLC, select Permission Management from the left sidebar, and click on Work Group > Add Work Group to create a work group for the user. When creating a work group, you can choose to bind it to a user or create an empty work group. For detailed operations, refer to Users and User Groups.



2. Enter the basic information: Provide the work group name and description.

The screenshot shows the 'Add work group' interface with a back arrow and the title 'Add work group'. Below the title are two steps: '1 Basic info' (active) and '2 Bind user'. The 'Basic info' section contains two input fields: 'Work group name' with the placeholder 'Enter a work group name' and 'Description' with the placeholder 'Enter a description'.

3. Associate a user: The associated user will acquire all permissions under the respective work group.

The screenshot shows the 'Add work group' interface with 'Basic info' completed and '2 Bind user' active. It features a 'Bind user' button and a 'Batch remove' button. Below these is a table with columns: Username, User type, Description, Add time, and Added by. The table is currently empty, showing 'No data'. At the bottom, it indicates 'Total items: 0' and a pagination control for '10 / page'.

Granting permissions to a work group

After creating the work group, click on the Authorize operation in the list to add permissions to the work group, including Data Permissions and Engine Permissions.

The screenshot shows the 'Permission management' interface for the 'Guangzhou' region. It has tabs for 'User' and 'Work group', with 'Work group' selected. A blue information banner at the top explains batch granting permissions. Below are buttons for 'Add work group' and 'Batch remove'. A table lists work groups with columns: Work group ID, Work group name, User count, Description, Added by, Add time, and Operation. One work group is listed with ID '30635', name 'test', and user count '0'. The 'Operation' column for this row has buttons for 'Edit', 'Authorize' (highlighted with a red box), and 'Remove'. A dropdown menu is open under 'Authorize', showing 'Data permission' and 'Engine permission'. The bottom shows 'Total items: 1' and a pagination control for '10 / page'.

Data permission

Data permissions include:

Data Catalog Permissions: These include two types of permissions under the data catalog, namely, the ability to Create Database and Create Data Catalog.

Grant data permissions

Basic info

Work group name: test

Description: --

Catalog/Database/Table

[Add permission](#) [Batch repossess](#)

Permission type	Catalog	Database	Table/View/Function	Column
No data				

Total items: 0

Row-level permissions

[Add permission](#) [Batch repossess](#)

Permission type	Catalog	Database	Data table
No data			

Add permission

Permission type: ☒ Catalog ☐ Database & table

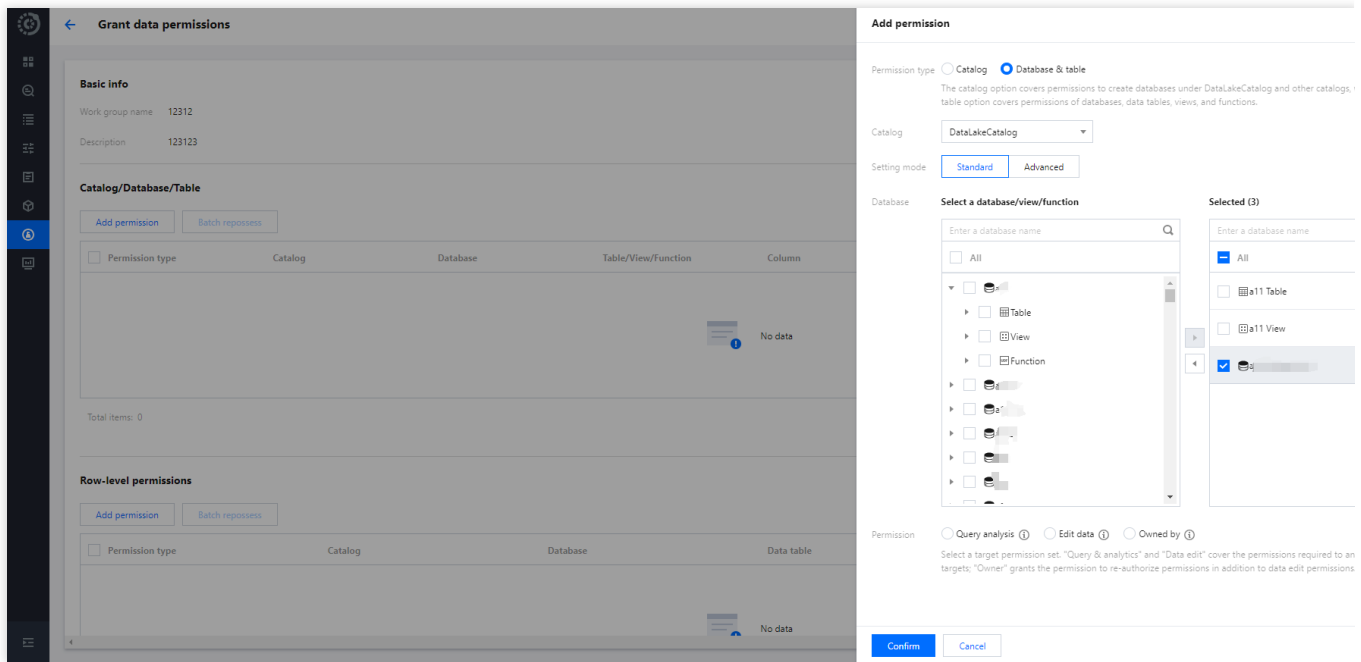
The catalog option covers permissions to create databases under DataLakeCatalog and other catalogs, while the table option covers permissions of databases, data tables, views, and functions.

Permission: ☐ Create database under DataLakeCatalog ☐ Create catalog

Authorizable: ☐ Yes

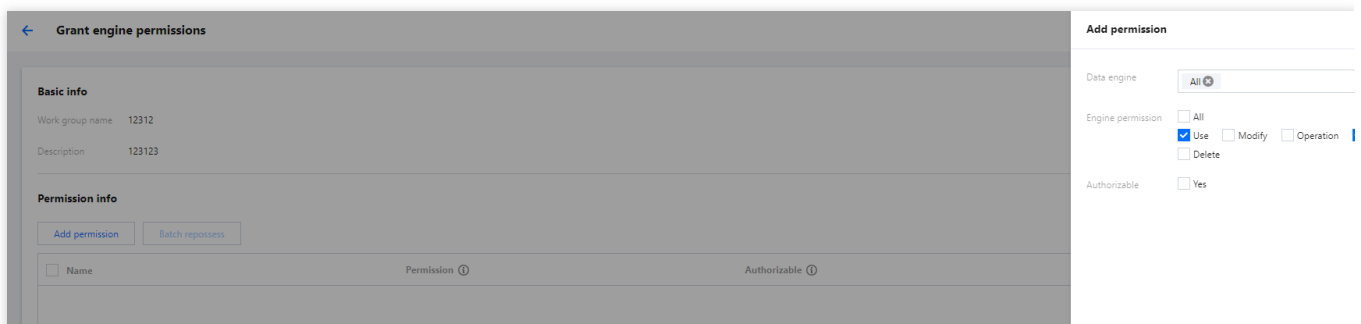
[Confirm](#) [Cancel](#)

Database Table Permissions: Fine-grained permissions at the database table level can be granted, including query and edit permissions for databases, tables, views, and functions.



Engine permission

Select a data engine and grant the permissions to use, modify, or delete it.



Engine operation permissions are granted automatically

DLC supports default enablement of engine operation class permissions. Once enabled, all users will by default have the following permissions for that engine:

Utilize: Execute tasks using this engine.

Operation: Initiation of engine suspension or standby.

Monitoring: Administration of engine usage monitoring.

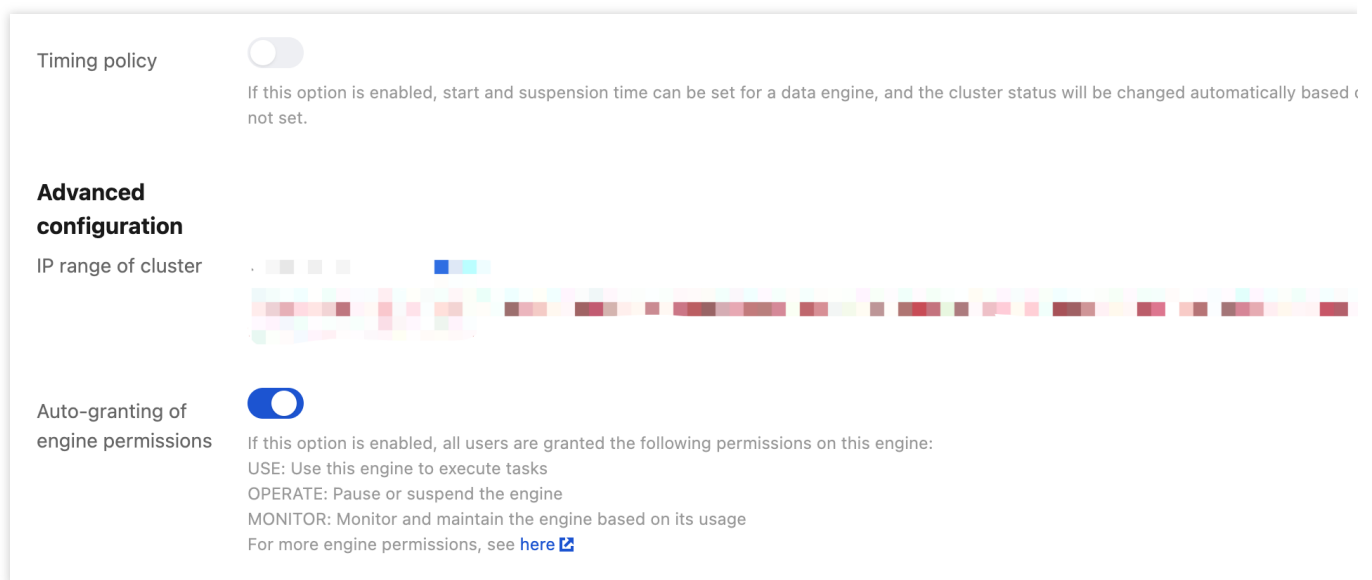
Note:

1. Upon termination, administrators inherently maintain all engine privileges. Ordinary users require an administrator to add permissions on the permission management page.
2. Existing ordinary user permissions will remain intact and can be deleted on the [Permission Management](#) page.
3. Subsequent newly created ordinary users have no usage rights, which should be manually added on the [Permission Management](#) page.

How do I enable or disable the self-delivery authorization engine

By default, the engine enables/disables two operation permission entries:

Access 1: [Engine Purchase page > Advanced Configuration Items](#)



Access 2: Go to the [SuperSQL engine](#) page and click Edit Auto-granting of engine permissions.

After setting engine permissions, click **Confirm**.

Tencent Cloud

OverviewProducts+

Data Lake Compute

Overview

Data Explore

Data Scheduling

Data Management

Data Job

Task History

Insight Management

Engine Management

SuperSQL Engine

Standard Engine

Engine Network Configuration

Ops Management

Permission

SuperSQL engine

Hong Kong

Data Lake Compute offers both public and private data engines. A public data engine is managed by Data Lake Compute and billed on a pay-as-you-go basis; a private data engine requires a dedicated VPC and VSwitch, and is billed on a pay-as-you-go basis or subscribed monthly. For more billing info, see [Billing Center](#). For operations and notes, see [Data Lake Compute User Guide](#).

Create resource

Bill query

Renewal management

Engine Name/ID	Auto-renewal	Start and stop policy	Cluster description	Auto-granting
<div>自动化专用常稳拨测_勿用</div> <div>DataEngine-lwxhwnud</div>	No	Manual start, Manual suspension	Private engine	No
<div>at_data_engine_presto</div> <div>DataEngine-p3d2xfq1</div>	--	Auto-start, Manual suspension	Private engine	No
<div>public-engine</div> <div>DataEngine-public-1313074...</div>	--	Manual start, Manual suspension	Public engine	No

Total items: 3

Quick Start with Partition Table

Last updated : 2024-07-17 15:25:14

Data Lake Compute Partition Table

With the partition catalog feature, you can store data with different characteristics in different catalogs. In this way, when exploring data, you can filter data by partition through the `where` condition. This greatly reduces the scanned data volume and improves the query efficiency.

Note:

Partitions in the same table should adopt the same data type and format.

Internal tables in Data Lake Compute are implemented as implicit partitions, so you don't need to care about the partition catalog structure.

Creating a Partition Table

Specify the partition field through the `PARTITIONED BY` parameter in the table creation statement.

Example: Creating the `test_part` partition table

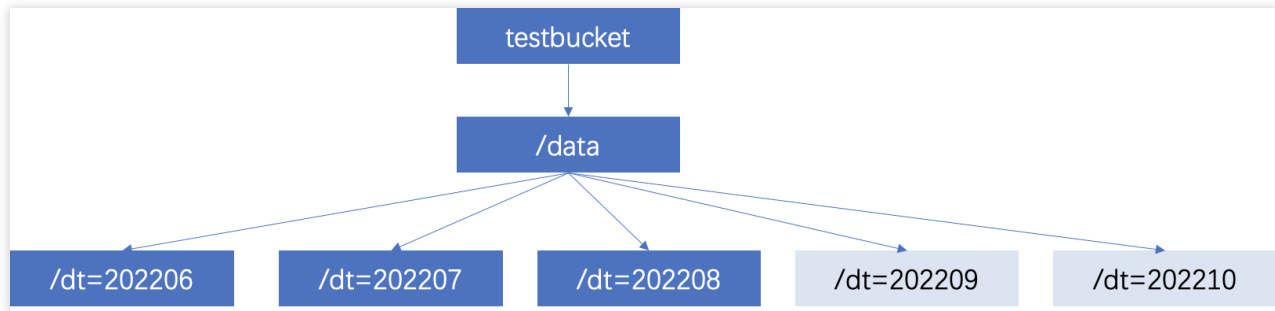


```
CREATE EXTERNAL TABLE IF NOT EXISTS `DataLakeCatalog`.`test_a_db`.`test_part` (  
  `_c0` int,  
  `_c1` int,  
  `_c2` string,  
  `dt` string  
) USING PARQUET PARTITIONED BY (dt) LOCATION 'cosn://testbucket/data/';
```

Adding a Partition

Adding a partition through `ALTER TABLE ADD PARTITION`

If your data partition catalog uses the Hive partitioning rule (partition column name=partition column value), the rule can be used to add partitions. The catalog is organized as follows:



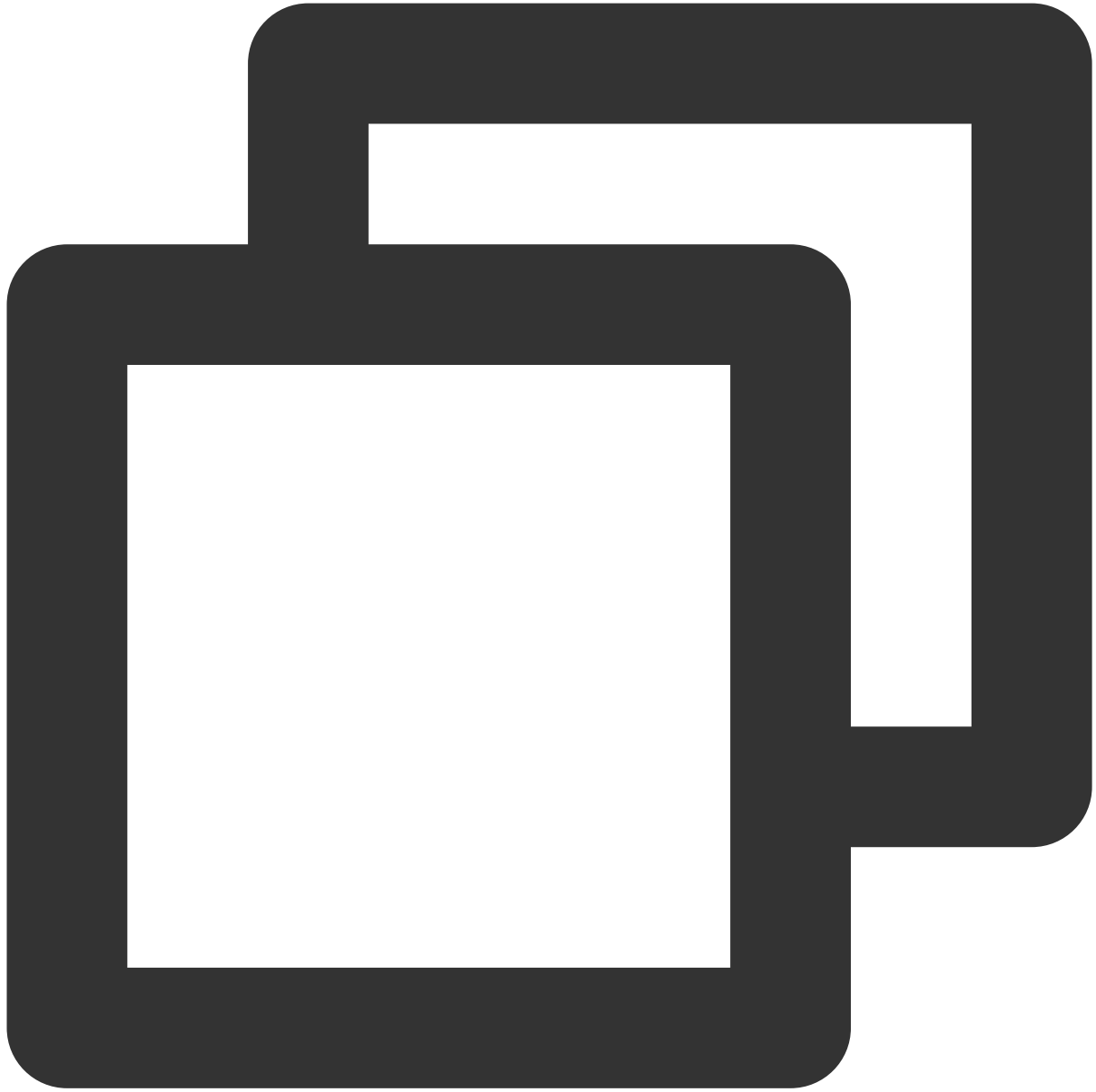


```
ALTER TABLE `DataLakeCatalog`.`test_a_db`.`test_part` add PARTITION (dt = '202206')
ALTER TABLE `DataLakeCatalog`.`test_a_db`.`test_part` add PARTITION (dt = '202207')
ALTER TABLE `DataLakeCatalog`.`test_a_db`.`test_part` add PARTITION (dt = '202208')
ALTER TABLE `DataLakeCatalog`.`test_a_db`.`test_part` add PARTITION (dt = '202209')
ALTER TABLE `DataLakeCatalog`.`test_a_db`.`test_part` add PARTITION (dt = '202210')
```

Adding a partition by specifying the location through `ALTER TABLE`

If your data adopts a general COS catalog (not in the "partition column name=partition column value" format), you can specify a catalog when adding a partition.

Sample SQL:

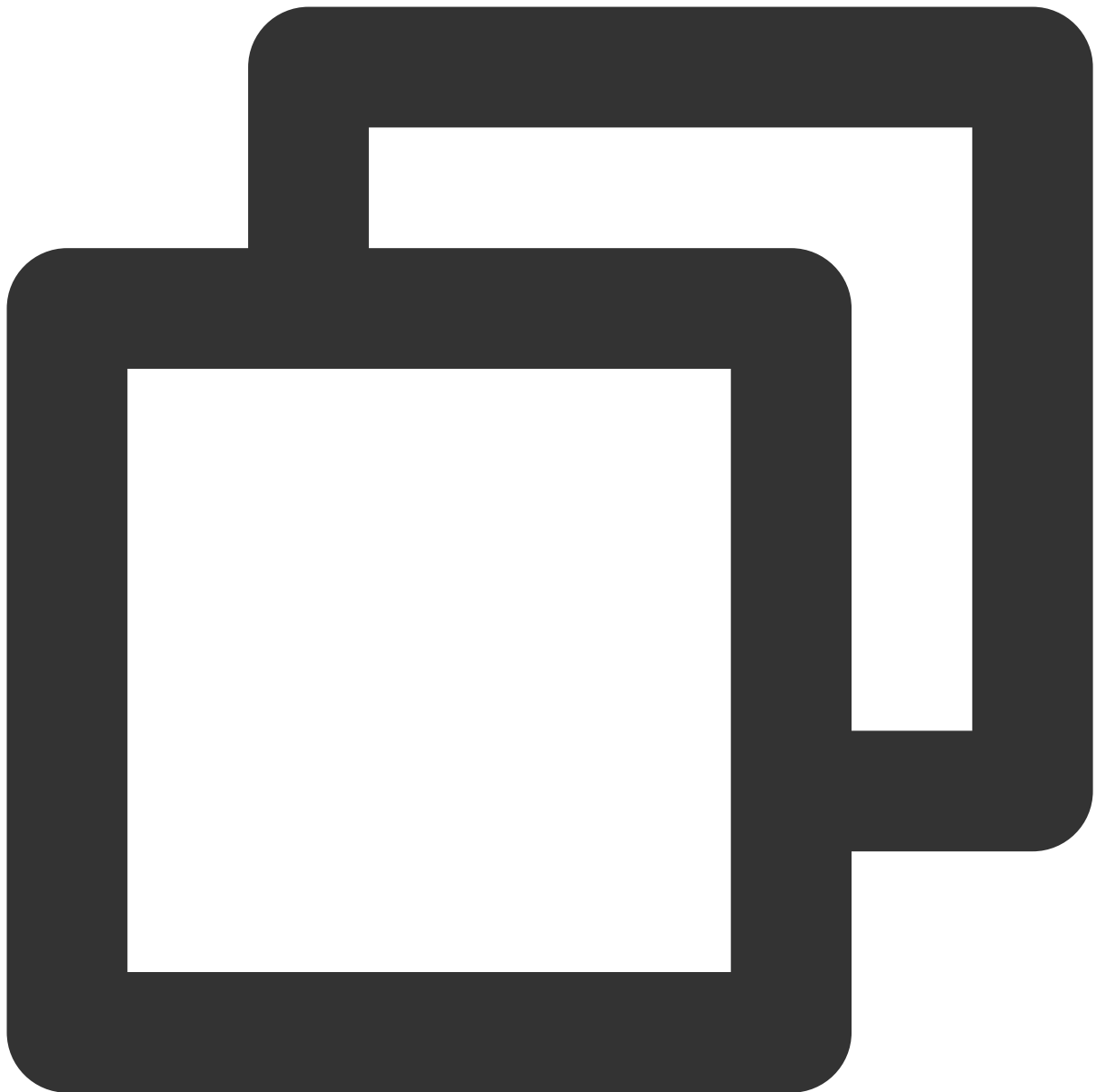


```
ALTER TABLE `DataLakeCatalog`.`test_a_db`.`test_part` add PARTITION (dt = '202211')
ALTER TABLE `DataLakeCatalog`.`test_a_db`.`test_part` add PARTITION (dt = '202212')
```

Automatically adding a partition through **MSCK REPAIR TABLE**

Use the **MSCK REPAIR TABLE** statement to scan the data catalog specified during table creation. If there is a new partition catalog, the system will automatically add the partitions to the metadata of the data table.

Sample SQL:



```
MSCK REPAIR TABLE `DataLakeCatalog`.`test_a_db`.`test_part`
```

We recommend you use `ALTER TABLE` to add a partition preferably, as automatic adding through `MSCK REPAIR TABLE` has the following restraints:

`MSCK REPAIR TABLE` only adds partitions to the metadata of the data table but does not delete them.

`MSCK REPAIR TABLE` is not recommended if the data volume is large, as it will scan all the data, which may cause a timeout.

If your partition catalog doesn't use the Hive partitioning rule (partition column name=partition column value), `MSCK REPAIR TABLE` cannot be used.

Quick Start with UDFs

Last updated : 2022-08-16 09:41:59

UDF description

You can write a user-defined function (UDF), package it into a JAR file, and define it as a function in Data Lake Compute for use in query analysis. Currently, UDFs in Data Lake Compute are in the Hive format and inherit

`org.apache.hadoop.hive.ql.exec.UDF` to implement the `evaluate` method.

Example: A simple array UDF

```
public class MyDiff extends UDF {
    public ArrayList<Integer> evaluate(ArrayList<Integer> input) {
        ArrayList<Integer> result = new ArrayList<Integer>();
        result.add(0, 0);
        for (int i = 1; i < input.size(); i++) {
            result.add(i, input.get(i) - input.get(i - 1));
        }
        return result;
    }
}
```

Sample POM file

```
<dependencies>
<dependency>
<groupId>org.slf4j</groupId>
<artifactId>slf4j-log4j12</artifactId>
<version>1.7.16</version>
<scope>test</scope>
</dependency>
<dependency>
<groupId>org.apache.hive</groupId>
<artifactId>hive-exec</artifactId>
<version>1.2.1</version>
</dependency>
</dependencies>
```

Creating a function

If you are familiar with the SQL syntax, you can create a function by running the `CREATE FUNCTION` statement on the **Data Explore** page. You can also create a function visually in the following steps:

Note :

The data management page of Data Lake Compute is currently in beta test. To try it out, [submit a ticket](#) for application.

1. Log in to the [Data Lake Compute console](#) and select the service region.
2. Select **Data management** on the left sidebar and select the target database.
3. Click **Function** to enter the function management page.
4. Click **Create function**.

You can also upload a local UDF program package or select a COS path (which requires COS permissions). The following example shows how to create a function by selecting a COS path.

The function class name includes the package information and the function execution class name.

Using a function

1. Log in to the [Data Lake Compute console](#) and select the service region.
2. Select **Data Explore** on the left sidebar and select a compute engine to use a SQL function.

Enabling Data Optimization

Last updated : 2024-07-31 17:23:30

In big data scenarios, frequent fragmented writes generate a large number of small files, which significantly slow down performance. Based on extensive production practice experience, DLC offers you efficient, simple, and flexible data optimization capabilities that can handle near real-time scenarios with large data volumes.

Note:

1. In Upsert scenarios, a large number of small files and snapshots will be generated. You need to configure data optimization before writing to avoid the need for extensive resource processing of historical backlog of small files after writing.
2. Currently, data optimization capability only supports DLC native tables.
3. The initial execution of data optimization tasks may be slow, depending on the stock data volume size and the selected engine resource specifications.
4. It is recommended to separate the data optimization engine from the business engine to avoid the situation where data optimization tasks and business tasks compete for resources, causing delays in business tasks.

Configure data optimization through the DLC console

DLC data optimization strategies can also be set in the data directory, database, and data table. When data optimization strategies are not specifically set for a database or data table, they will inherit the optimization strategy from the previous level. When configuring data optimization, users need to select an engine. To execute data optimization tasks, if the user currently does not have a data engine, they may refer to [Purchasing Dedicated Data Engine](#) to make a purchase. DLC data governance supports Spark SQL Engine and Spark Job Engine.

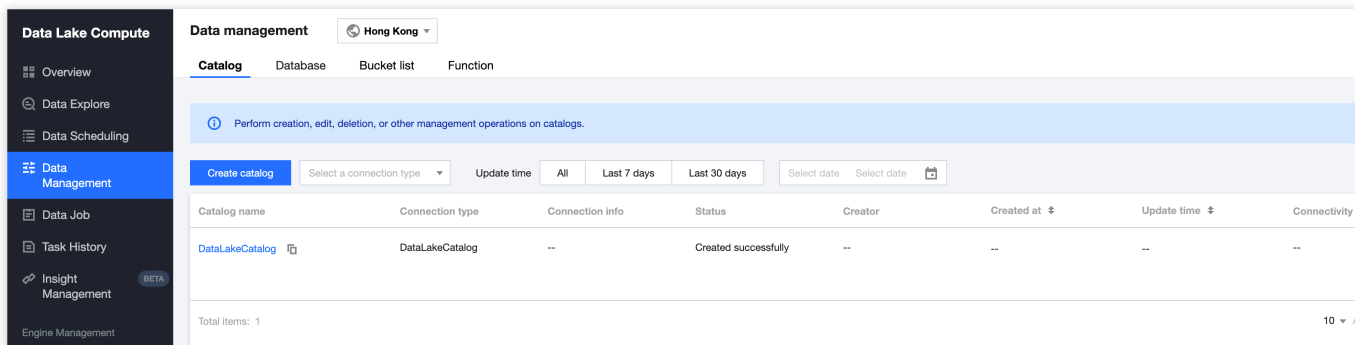
Note:

1. If a user chooses the Spark Job Engine as the data optimization resource, DLC will create data optimization tasks on that engine. Depending on the size of the cluster, the optimized task data created will vary. For instance, if the cluster size is smaller than 32 CU, one data optimization task will be created to execute all optimization tasks. If the cluster size is larger than 32 CU, two data optimization tasks will be created to separately execute write optimization and data deletion optimization.
2. When choosing a Spark Job as a data optimization resource, some resources need to be reserved. If the optimization tasks queue exceeds 50, DLC will launch temporary data optimization tasks to quickly process the backlog of optimization tasks.

Data Directory Configuration Steps

You can use DLC's Data Catalog Editing Feature to configure data optimization capabilities for your data directory.

1. Go to the Data Management Module in the [DLC Console](#), enter the **Data Management** page, and click **Data Optimization**.



2. Open the **Data Optimization** page of the data directory, configure the corresponding data optimization resources and policies. Once confirmed, the data optimization feature will automatically apply to that data directory.

Edit database ✕

Governance resources

Select a data engine ▼

Only SparkSQL engines are supported

Governance rules

☐ Smart ? ☒ Custom ?

Governance rules ▲

Merge small files ?

☒

Min file count

— 5 +

Max size of file to merge

— 128 + M

Scheduling interval

— 60 + Minute

Data files threshold ?

— 20 +

Delete files threshold ?

— 20 +

Equality delete files threshold ?

— 1000 +

Position delete files threshold ?

— 1000 +

Delete expired snapshots ?

☒

Retained snapshot count

— 5 +

Deletion time slot

— 2 + Day

Concurrency

— 4 +

Scheduling interval

— 600 + Minute

Confirm

Cancel

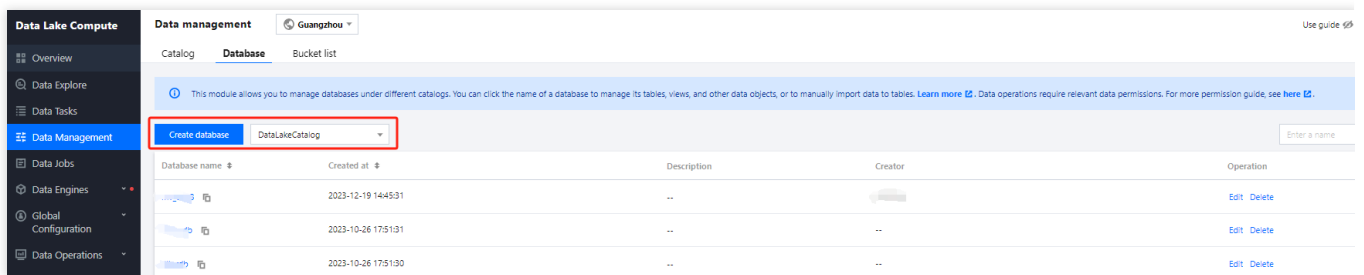
Note:

Only supports configuring data optimization for the DataLakeCatalog data directory.

Database Configuration Steps

If you want to configure a data optimization strategy for a specific database individually, you can use the database editing capabilities of DLC to configure data optimization capabilities for the database.

1. Enter the [DLC console](#) Data Management Module, enter the **Database** page, enter the database list under DataLakeCatalog.



2. Open the database page, click **Data Optimization Configuration**. Once confirmed, the data optimization strategy will automatically apply to that database.

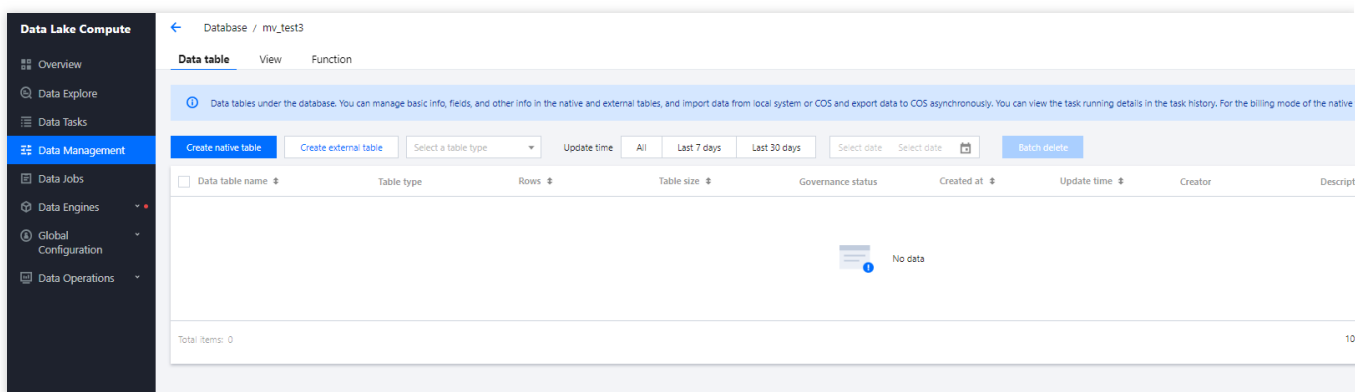
Note:

When creating a database and editing data, the default to show data optimization strategy inherits the data optimization strategy of the superior data directory. If you want to customize the data optimization strategy, you need to select **Custom Configuration** and configure data optimization resources and policies.

Data Table Configuration Steps

If you want to configure a data optimization strategy for a specific data table individually, you can use the data table editing capabilities of DLC to configure data optimization capabilities for the data table.

1. Enter the [DLC console](#) Data Management Module, enter the **Database** page, select a database, then enter the **Data Table** list page, and click **Create Native Table**.



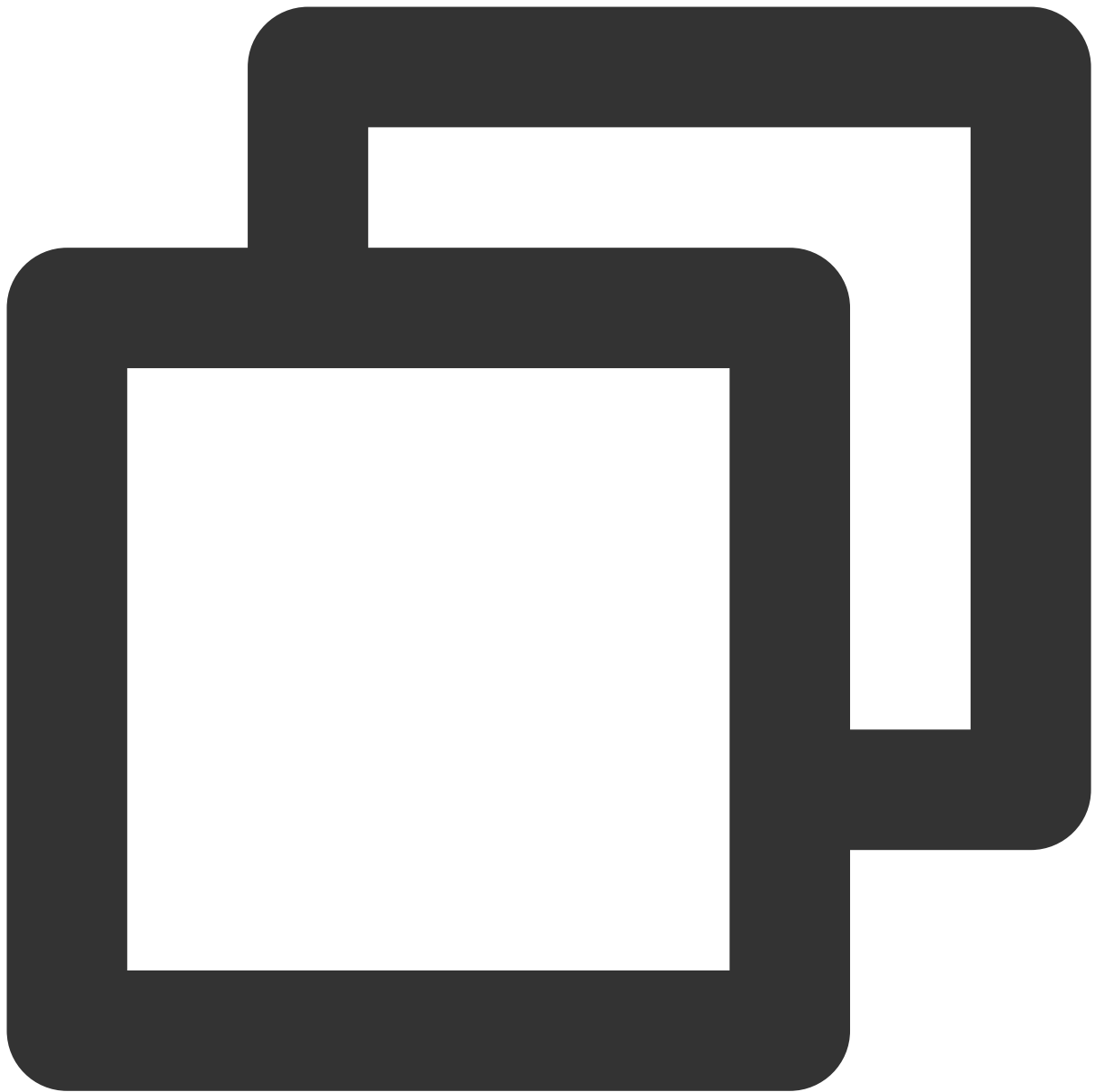
2. Open the Create Native Table page, configure the corresponding optimization resources, and once confirmed, the data optimization strategy will automatically apply to that data table.
3. For already created tables, you can click **Data Optimization Configuration** to edit the existing data table's data optimization strategy.

Note:

When creating or editing a data table, the default data optimization strategy displayed inherits from the parent data table's data optimization strategy. If you want to customize the data optimization strategy, you need to select **Custom Configuration** and configure data optimization resources and policies.

Optimize data through attribute field configuration

Besides the above visualization method for configuring data optimization, you can also manually specify library and table field attributes for configuration. For example:



```
// for table govern policy
ALTER TABLE
    `DataLakeCatalog`.`wd_db`.`wd_tb`
SET
TBLPROPERTIES (
    'smart-optimizer.inherit' = 'none',
    'smart-optimizer.written.enable' = 'enable'
)
// for database govern policy
ALTER DATABASE
    `DataLakeCatalog`.`wd_db`
```

```
SET
DBPROPERTIES (
  'smart-optimizer.inherit' = 'none',
  'smart-optimizer.written.enable' = 'enable'
)
```

The attribute values for data optimization can be modified via the ALTER statement. The attribute value definitions are as follows:

Attribute Value	Meaning	Default Value	Value Description
smart-optimizer.inherit	Whether to Inherit from the Parent Strategy	default	none: Does not inherit default: Inherit
smart-optimizer.written.enable	Whether Write Optimization is Enabled	disable	disable: Not Enabled enable: Enabled
smart-optimizer.written.advance.compact-enable	(Optional) Advanced Write Optimization Parameters, Whether to Start Small File Merge	enable	disable: Not Enabled enable: Enabled
smart-optimizer.written.advance.delete-enable	(Optional) Advanced Write Optimization Parameters, Whether to Start Data Cleanup	enable	disable: Not Enabled enable: Enabled
smart-optimizer.written.advance.min-input-files	(Optional) Merge Minimum Number of Input Files	5	When the number of files in a table or partition exceeds the minimum number of files, the platform will automatically check and initiate file optimization merge. File optimization merge can effectively improve analyze query performance. The larger the minimum number of files, the higher the resource load. The smaller the minimum number of files, the more flexible the

			execution, and tasks will be more frequent. It is recommended to set the value to 5.
smart-optimizer.written.advance.target-file-size-bytes	(Optional) Merge Target Size	134217728 (128 MB)	During file optimization merge, files will be combined to meet the target size as much as possible. It is recommended to set the value to 128M.
smart-optimizer.written.advance.retain-last	(Optional) Snapshot Expiration Time, Unit Days	5	When the snapshot retention time exceeds this value, the platform will mark the snapshot as expired. The longer the snapshot expiration time, the slower the snapshot cleanup speed, and the more storage space is occupied.
smart-optimizer.written.advance.before-days	(Optional) Number of Expired Snapshots to Retain	2	Expired snapshots exceeding the retention count will be cleaned up. The more expired snapshots retained, the more storage space is occupied. It is recommended to set the value to 5.
smart-optimizer.written.advance.expired-snapshots-interval-min	(Optional) Snapshot Expiration Execution Cycle	600(10 hour)	The platform will periodically scan snapshots and expire them. The shorter the execution cycle, the more sensitive the snapshot expiration will be, but it may consume more resources.
smart-optimizer.written.advance.cow-compact-enable	(Optional) Enable Merge for COW Tables (V1 Table or V2 Non-Upert Table)	disable	Once this configuration item is enabled, the system will automatically generate file merge tasks for COW tables. Note: COW tables usually have a large data volume, and file merging may consume a lot of resources. You can choose whether to enable file merging for COW tables based on resource availability and table size.
smart-optimizer.written.advance.strategy	(Optional) File Merge Strategy	binpack	binpack (default merge strategy): Merges data files that meet the

			merge conditions into larger data files using the append method. sort: The sort strategy merges files based on specified fields. You can choose query condition fields that are frequently used in your business scenarios as the sorting fields. Merging in this way can improve query performance.
smart-optimizer.written.advance.sort-order	(Optional) When the file merge strategy is sort, the configured sort collation	-	If you haven't configured a sorting strategy, the Upsert Table will sort using the configured upsert key values (by default, the first two key values) in an ASC NULLS LAST manner. If a sorting strategy cannot be found for COW Table during a sort merge, the binpack default merge strategy will be used.
smart-optimizer.written.advance.remove-orphan-interval-min	(Optional) Period for Removing Orphan Files	1440(24 hour)	The platform will periodically scan and clean up orphan files. The shorter the execution cycle, the more sensitive the cleanup of orphan files will be, but it may consume more resources.

Optimization Suggestions

The DLC backend regularly statistics native table metric items and combine these metrics with best practices to provide optimization suggestions for native tables. There are four categories of optimization suggestion items, including basic configuration for table usage scenarios, data optimization recommendations, and recommendations for data storage distribution items.

Optimization recommendation check items	Sub-check item	Meaning	Business Scenario	Optimization Suggestions
Basic attribute configuration check of the table	Metadata governance enabled	Check whether metadata governance is enabled to prevent metadata volume expansion due to frequent table writes	append/merger into/upsert	Recommended to enable

	Bloom filter set	Check if the bloom filter is set. After enabling the bloom filter for MOR tables, it quickly filters the deletes files, speeding up MOR table queries and deletes file merges	upsert	Must enable
	Metrics key attributes configured	Check if metrics are set to full. Once this attribute is enabled, it will record all metrics information, preventing incomplete metrics information recording due to excessively long table locations	append/merger into/upsert	Must enable
Data optimization configuration check	Small File Merge	Check if small file merging is enabled	merge into/upsert	Must enable
	Snapshot Expiration	Check if snapshot expiration is enabled	append/merge into/upsert	Recommended to enable
	Remove orphaned files	Check if removing orphaned files is enabled	append/merge into/upsert	Recommended to enable
Recent governance task check items	Recent governance task check items	If data governance is enabled, the system will track the execution of data governance tasks. If multiple tasks in a row time out or fail, it will be deemed in need of optimization	append/merger into/upsert	Recommended to enable
Data Storage Distribution	Average File Size	Collect summary information from snapshots, calculate the average file size, and if the average file size is less than 10MB, it will be deemed in need of optimization	append/merger into/upsert	Recommended to enable
	MetaData Meta	Collect table	append/merger	Recommended

	File Size	metadata.json Meta File Size, if the file size exceeds 10MB, it will be deemed in need of optimization	into/upsert	to enable
	Number of Table Snapshots	Collect Number of Table Snapshots, if the number of snapshots exceeds 1000, it will be deemed in need of optimization	append/merger into/upsert	Recommended to enable

Optimization Suggestions for Basic Configuration Items of Table Attributes

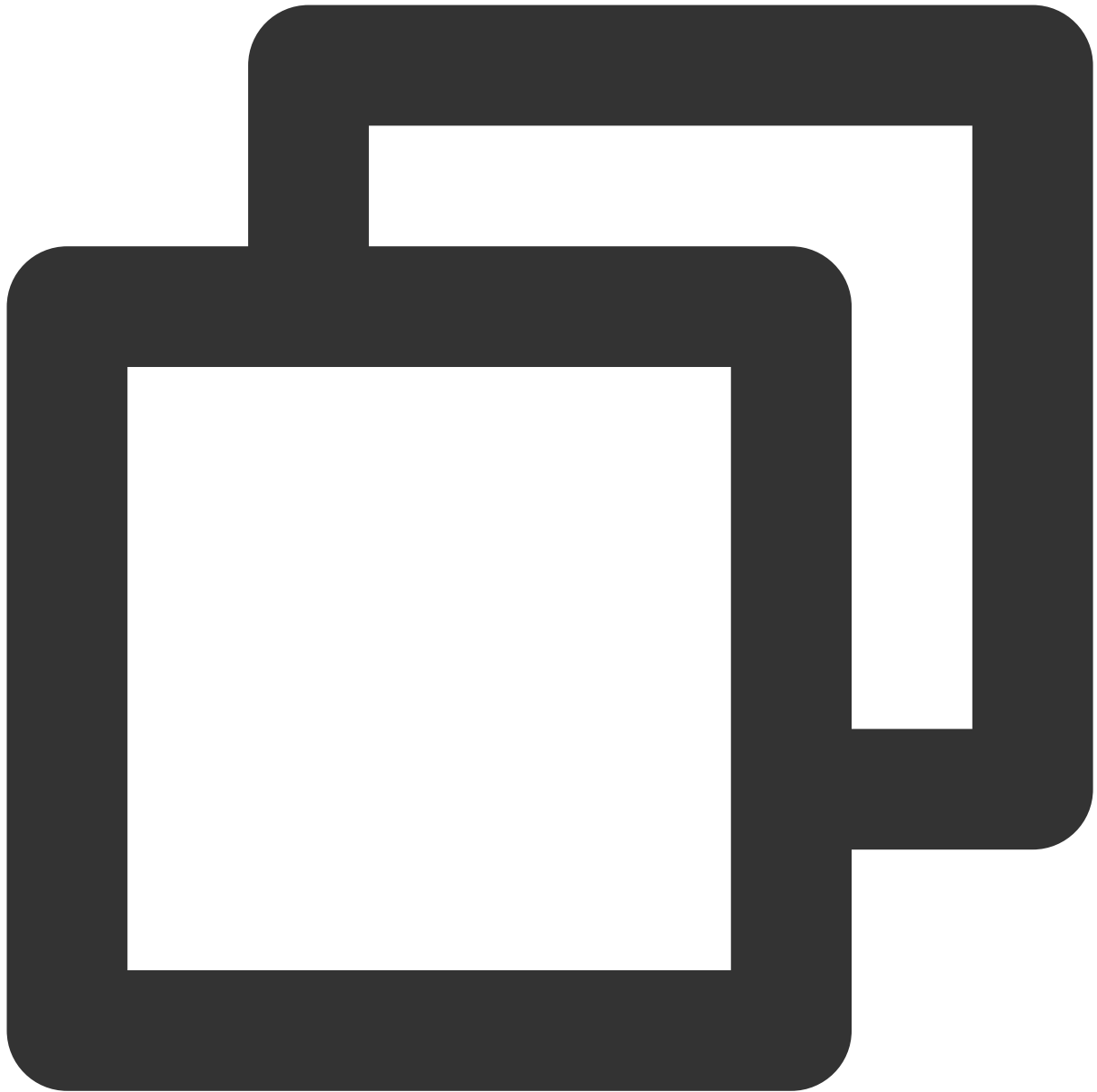
Check and configure Metadata Governance Method

Step1 Inspection Method

Use 'show TBLPROPERTIES' to view table attributes and check if "write.metadata.delete-after-commit.enabled", "write.metadata.previous-versions-max" are configured.

Step2 Configuration Method

If Step1 finds that it's not configured, you can configure it using the following Alter table DDL, with the method referenced below.



```
ALTER TABLE
  `DataLakeCatalog`.`axitest`.`upsert_case`
SET
  TBLPROPERTIES (
    'write.metadata.delete-after-commit.enabled' = 'true',
    'write.metadata.previous-versions-max' = '100'
  );
```

Note:

To enable automatic metadata governance, "write.metadata.delete-after-commit.enabled" should be set to true. The number of historical metadata to retain can be set according to the actual situation, for example, setting "write.metadata.previous-versions-max" to 100 will retain up to 100 historical metadata.

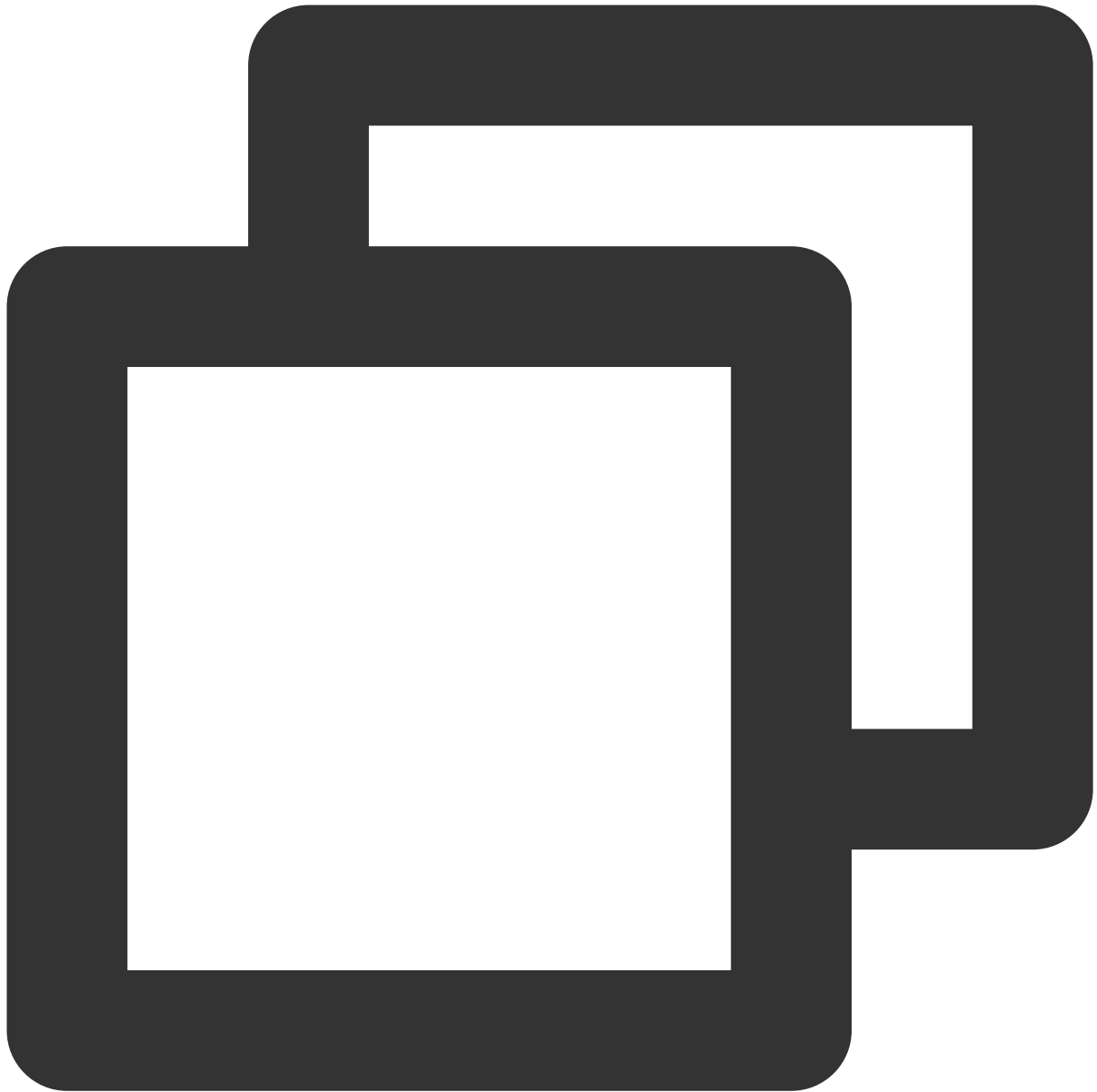
Inspecting and Setting Bloom Filter Method

Step1 Inspection Method

Use show TBLPROPERTIES to view table attributes, and check if "write.parquet.bloom-filter-enabled.column.{column}" is set to true.

Step2 Configuration Method

If Step1 finds that it's not configured, you can configure it using the following Alter table DDL, with the method referenced below.



```
ALTER TABLE
  `DataLakeCatalog`.`axitest`.`upsert_case`
SET
  TBLPROPERTIES (
    'write.parquet.bloom-filter-enabled.column.id' = 'true'
  );
```

Note:

It is recommended to enable bloom in upsert scenarios, and configure it based on the upsert primary key. If there are multiple primary keys, it is advisable to set it for the first two primary key fields.

After updating the bloom fields, if there are upstream writes from inlong/oceans/flink, you must restart the upstream import job.

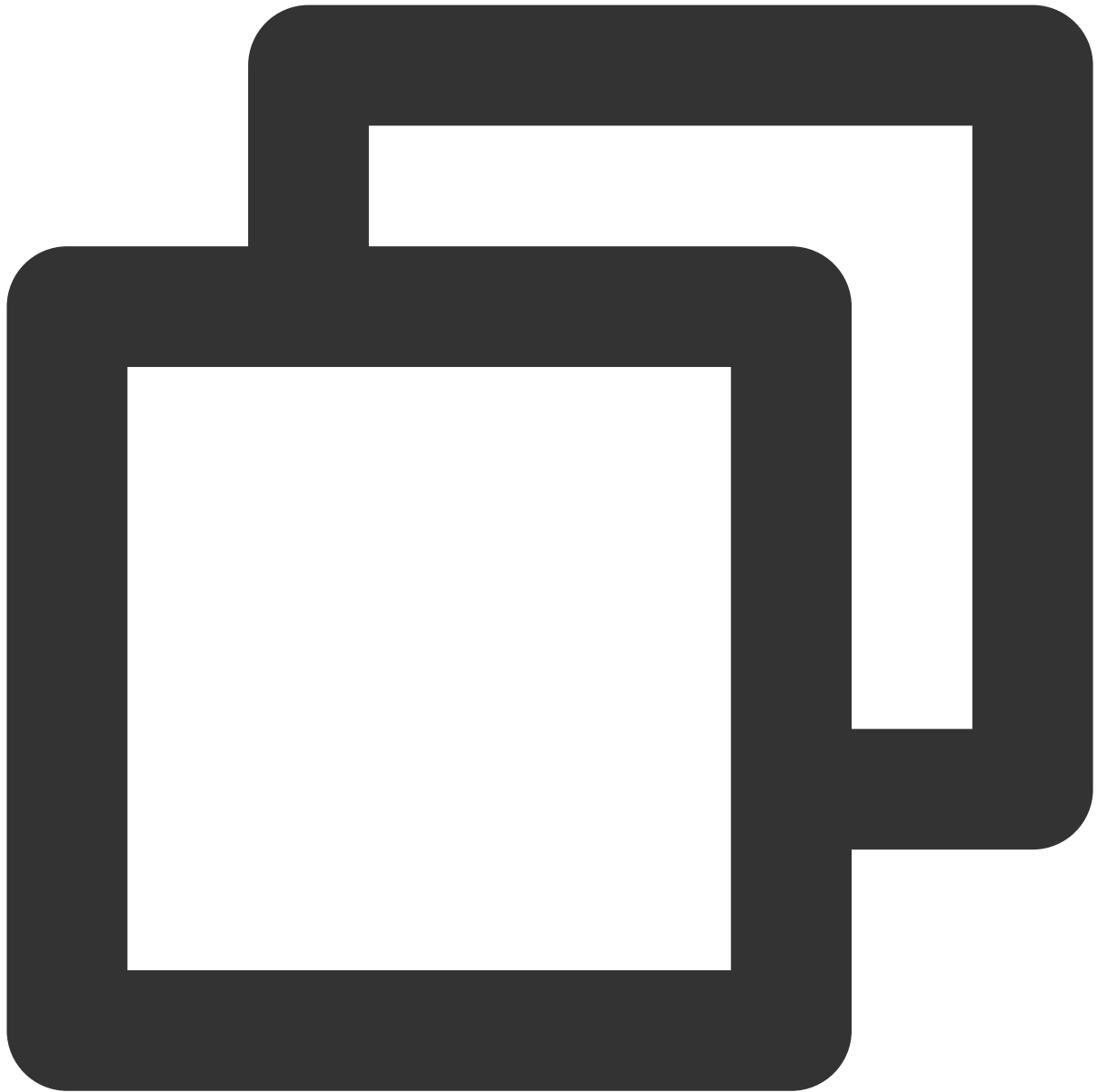
Check and configure table key attributes metrics

Step1 Inspection Method

View table properties using ``show TBLPROPERTIES`` and check if `"write.metadata.metrics.default"` is configured as `"full"`.

Step2 Configuration Method

If Step1 finds that it's not configured, you can configure it using the following Alter table DDL, with the method referenced below.



```
ALTER TABLE
  `DataLakeCatalog`.`axitest`.`upsert_case`
SET
  TBLPROPERTIES('write.metadata.metrics.default' = 'full');
```

Data Optimization Configuration Recommendations

Step1 Inspection Method

Check using SQL

View table properties using `show TBLPROPERTIES` and check if data optimization is configured. Refer to [DLC Native Table Core Capabilities](#) for the attribute configuration values for data optimization.

Visual inspection through the DLC Console

Go to the Data Management Module in the [DLC Console](#), enter the **Database** page, select a database to access the **Data Table** list page, choose the table to inspect, and proceed to **Data Optimization Configuration**.

Step2 Configuration Method

Follow the guidance to enable data optimization.

Recent recommendations for data governance optimization task items

Check if data governance is functioning properly

Step1 Inspection Method

Enter the [DLC Console](#) Data Management Module, enter the **Database** page, select a database and then enter the **Data Table** list page, click on the data table name, enter **Optimized Monitoring**, choose **Optimization Task** then select **Today's Optimization**, check for tasks that failed in the last three hours, if there are any, the check is not passed. Select the failed task, in **View Details** look at the **Execution Results**.

Step2 Fix Methods

Summary of Reasons and Solutions for Failed Scenario Data Optimization Tasks.

1. Data Governance Configuration Error led to failure.

Sort Merge Strategy was enabled, but the collation was incorrectly configured, or a nonexistent rule was set.

The configuration for the data governance engine has changed, leading to the inability to find an appropriate engine when running governance tasks.

2. Task Execution Timed Out.

Note:

After repairing the recent data optimization task performance, it is necessary to wait three hours before checking if it has recovered.

Data Storage Distribution Item Optimization Suggestions

Note:

Failure in this scenario check is usually due to large data volume. It's recommended to handle it manually before considering addition to Data Optimization Governance.

It is recommended to use the more efficient Spark job engine.

When manually merging small files, configure the target-file-size-bytes parameter based on the business scenario. For upsert operations, it is advised not to exceed 134217728, i.e., 128M. For append/merge into operations, it is advised not to exceed 536870912, i.e., 512M.

When using the Spark job engine to handle snapshot expiration, the max_concurrent_deletes parameter can be increased.

Average Data File Size Check Failure Handling Method

Step1 Summary of Reasons

The average size of data files is too small, usually occurring in the following scenarios:

The table is partitioned too finely, resulting in each partition having only a small amount of data.

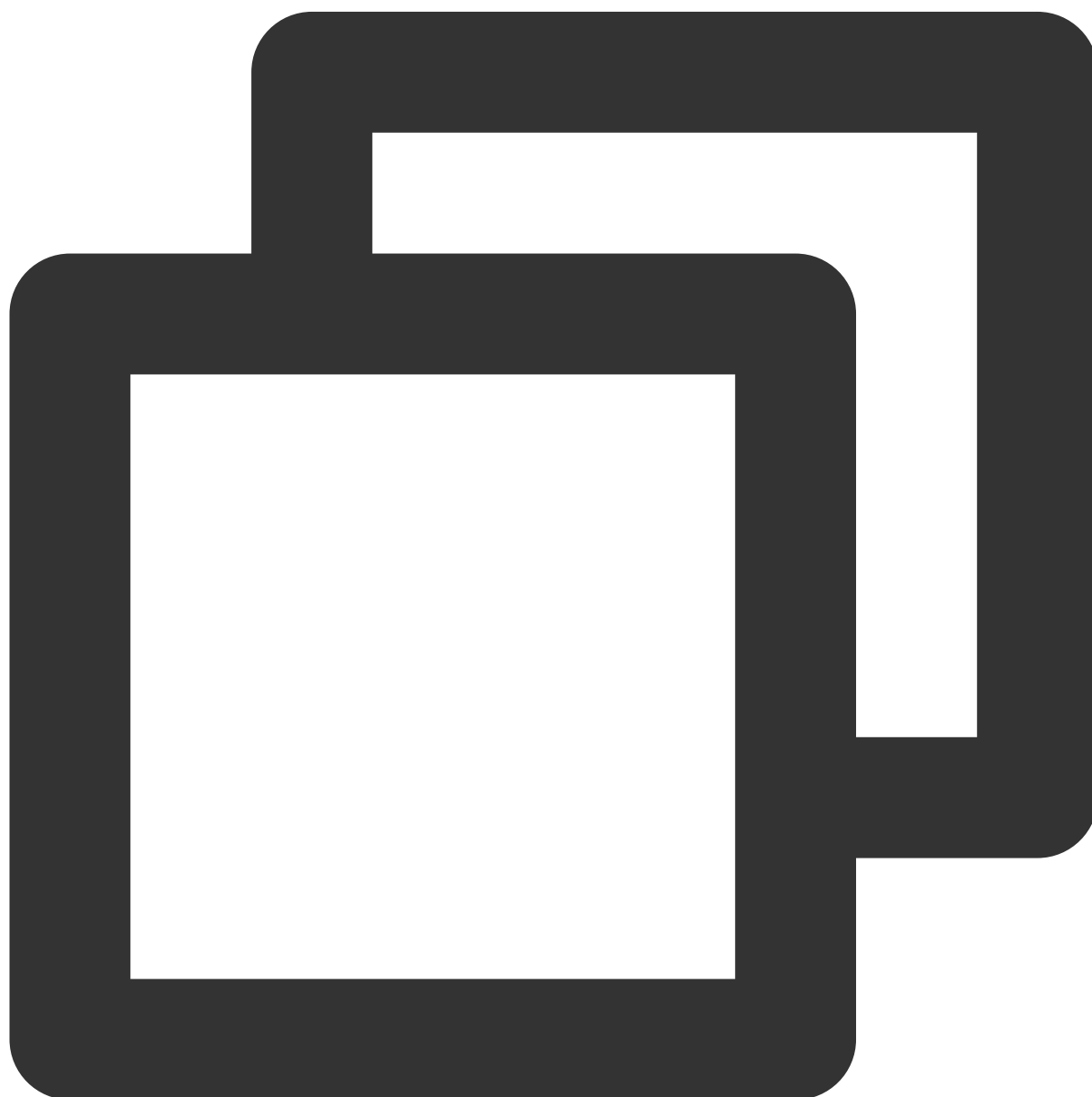
When tables are written using the Insert into/overwrite method, the upstream data is dispersed, such as when the upstream data is also from a partitioned table with little data within partitions.

The table is written to the MOR Table using the merge into method, but small file merging has not been performed.

The table is written using the upsert method, but small file merging has not been performed.

Step2 Fix Methods

Refer to the following SQL to manually perform small file merging.




```
CALL `DataLakeCatalog`.`system`.`rewrite_data_files` (  
  `table` => 'test_db.test_tb',  
  `options` => map(  
    'delete-file-threshold',  
    '10',  
    'max-concurrent-file-group-rewrites', --Subject to actual resource conditions,  
    '5',  
    'partial-progress.enabled',  
    'true',  
    'partial-progress.max-commits',  
    '10',  
    'max-file-group-size-bytes',  
    '10737418240',  
    'min-input-files',  
    '30',  
    'target-file-size-bytes',  
    '134217728'  
  )  
)
```

MetaData Meta File Size Check Failure Handling Method

Step1 Summary of Reasons

MetaData file size is too large, usually caused by an excessive number of data files, mainly due to the following reasons:

The table has been written to using the append method for a long time, and each write involves a large number of scattered files.

The table has the attributes of an MOR table and has been written to long-term using the merge into method, but small file merging is not enabled.

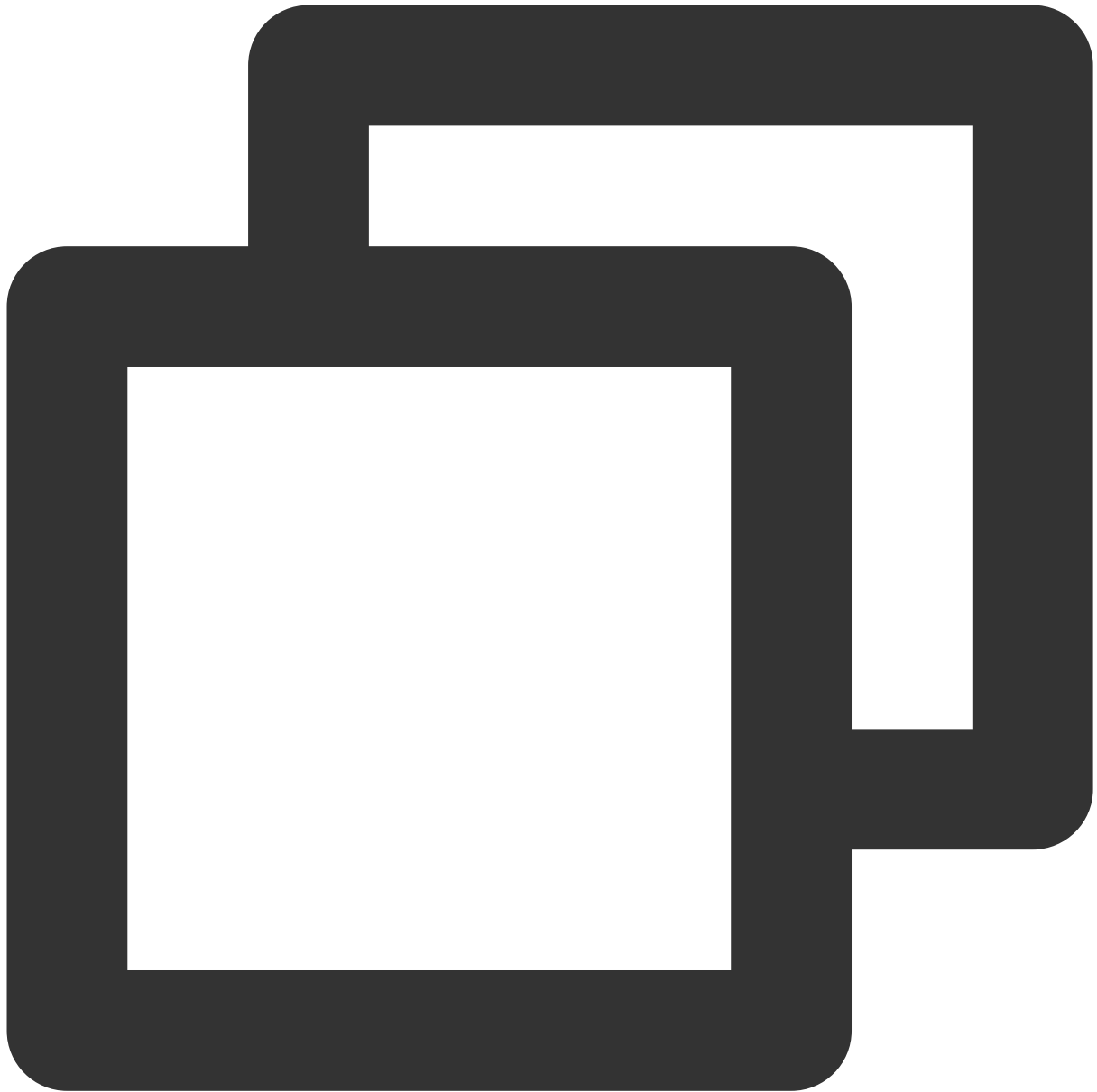
The table has not undergone snapshot expiration for an extended period, maintaining a large number of historical snapshot data files.

The table partitions are large, and each partition contains a large number of small files.

Step2 Fix Methods

Refer to manually perform small file merging.

Refer to the following SQL to manually execute the expired snapshot SQL and clean up historical snapshots.



```
CALL DataLakeCatalog.system.rewrite_data_files(  
  table => 'test_db.test_tb',  
  options => map(  
    'delete-file-threshold',  
    '10',  
    'max-concurrent-file-group-rewrites', --The higher the concurrency, and the fa  
    '5',  
    'partial-progress.enabled',  
    'true',  
    'partial-progress.max-commits',
```

```
'10',  
'max-file-group-size-bytes',  
'10737418240',  
'min-input-files',  
'30',  
'target-file-size-bytes',  
'134217728'  
)  
)
```

Based on the service scenario, the written files are aggregated to a certain extent to avoid scattered files.

If the data is written into insert into/insert overwrite, you can automatically add a repartition in either of the following ways.

1. This parameter takes effect when both of the following parameters are true. In this case, you can use the preceding parameters to control the number or size of automatically adapted partitions after repartition.

`spark.sql.adaptive.enabled` : This parameter must be true. The default value is true for cluster creation.

`spark.sql.adaptive.insert.repartition` : This parameter must be true. The default value is false for cluster creation.

2. Specify the following parameters to take effect. This case repartition spark. The partition number after SQL. The adaptive. Insert. The repartition. ForceNum the specified value.

`spark.sql.adaptive.insert.repartition.forceNum` : This parameter specifies the value of the partition to be partitioned. It is left blank by default when the cluster is created.

Check the number of snapshots. This operation fails to pass the check

Step1 Cause summary

Snapshots do not expire for a long time.

The upsert writes data to the checkpoint interval improperly, resulting in a large number of snapshots.

Step2 Repair method

See Snapshot expiration SQL to perform snapshot expiration operations.

Adjust the flink write checkpoint interval. It is recommended that the checkpoint interval of DLC native table upsert be 3 to 5 minutes.

Cross-Source Analysis of EMR Hive Data

Last updated : 2024-07-17 15:27:21

Data Lake Compute allows you to configure an EMR Hive data source for multi-source federated data analysis.

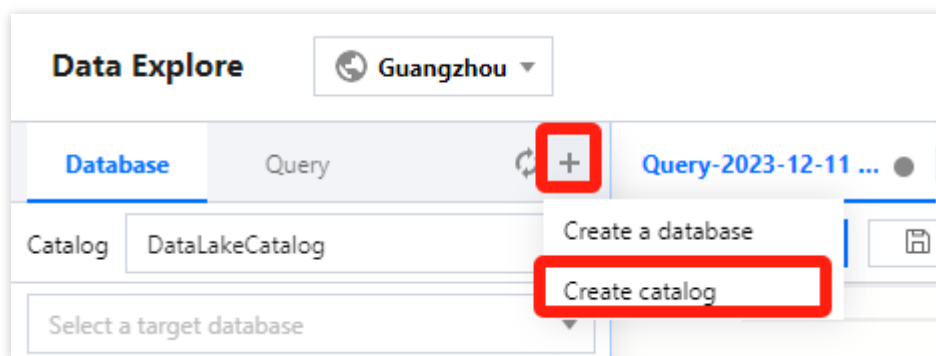
Preparations

Get the EMR Hive address.

Use an account with the permission to create data catalogs. For more information on permissions, see [Permission Overview](#).

Creating an EMR Hive data source

1. Log in to the [Data Lake Compute console](#) and select the service region.
2. Select **Data Explore** on the left sidebar, click **+** in the **Database & table** column, and select **Create data catalog**.



3. Select **EMR Hive (HDFS)** for **Connection type** and select the target EMR instance. The VPC information will be populated by default after the instance is selected. **EMR versions supported by EMR Hive are 2.3.5, 2.3.7, 3.1.1, and 3.1.2.**

Note:

Relevant permissions are required for you to select the EMR Hive instance.

Create catalog

1 Catalog configuration

>

2 Network configuration

Connection type *

EMR Hive(HDFS)

Connection name *

hdfs_demo

Description

hdfs_demo

EMR instance *

Data source VPC *

Ha setting *

HA


Non-HA


Hive version *

2.3.5

Hive access address *

Example: thrift://ip:port, metastore. The address can be queried in the [EMR console](#)

Cluster name 

Node  *

Back

Next

4. Select the **Run cluster**. Currently, you can only select a private data engine of Presto. If there is no engine, create one on the **Data engine** page. For more information on the purchase process, see [Purchasing Private Data Engine](#).

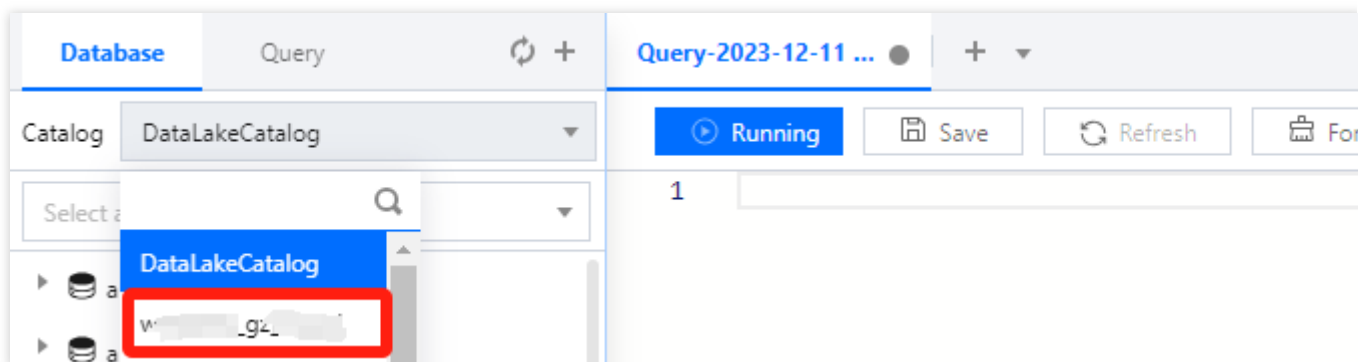
Note:

The IP range of the selected data engine cannot be the same as that of the EMR instance; otherwise, a network conflict will occur, and you cannot query or analyze data.

5. Click **Confirm**.

Querying the EMR Hive data

After the data catalog is created, you can switch to it from the **Data catalog** menu on the **Data Explore** page.



At this point, you can query and analyze the data catalog with SQL statements.

Select the data engine bound when the data catalog is created and click **Run** to get the query result.

Note:

You can only query the data catalog with its bound data engine. To change the bound engine, click the set icon next to the data catalog.

