Tencent Cloud TCLake

Operation Guide

Product Documentation

Contents

Operation Guide
Data Catalog

Last updated：2026-05-15 10:25:45

A Catalog serves as the top-level logical entity for user-facing metadata within TCLake, organizing metadata resources into a hierarchical structure. It enables metadata isolation and fine-grained access control across different business units and users. This document provides a guide to basic catalog operations.

Data Catalog Hierarchy Model

Within TCLake, all metadata is registered and stored in an underlying metadata store (Metalake), which is completely transparent to the user. The hierarchical structure of metadata objects within a unified catalog is divided into three distinct levels. When referencing tables, volumes, models, or functions, they are represented using a three-level namespace format (e.g., catalog.schema.table).

Level 1: Catalog

A catalog organizes data assets across various formats. Currently, TCLake supports the following catalog categories:

Category	Subcategory	Description
Built-in Data Catalog	Lakehouse Catalog	A structured data catalog that natively hosts open formats, including the TCIceberg batch-stream unified table format and the Lance multimodal format.
	Volume Catalog	An unstructured data catalog designed to directly manage files (e.g., images, video, and audio) stored in Cloud Object Storage (COS), enabling unified metadata governance for unstructured assets.
	Model Catalog	A built-in catalog for machine learning model artifacts. It allows you to register models trained in MLOps frameworks (e.g., MLflow) into TC-Catalog, delivering full-lifecycle management for ML models. (Currently under development)
External Data Catalog	e.g., MySQL, EMR, DLC, TCHouse	Establishes connections to external data sources via JDBC or similar protocols to retrieve their metadata in real time.

Level 2: Schema

A Schema (often synonymous with a database) is a second-level object within a catalog. Depending on the catalog type, a schema can contain concrete data resources such as tables, views, volumes, machine learning models, and functions. Schemas organize Data and AI assets into logical groupings, providing a more granular level of categorization than the catalog itself.

Level 3: Data Resources

The third level of the catalog hierarchy consists of concrete data entities, such as Tables, Volumes, Models, and Functions, depending on the specific catalog type.

Tables / Views

A table is a concrete dataset hosted within the TCLake service, organizing data into rows and columns. A view is a saved query built on top of one or more tables.

Volumes

A volume is a logical entity used to manage unstructured data stored in systems like COS or HDFS. For example, by mapping files under a COS path (e.g., `examplebucket.cos.ap-guangzhou.myqcloud.com/folder/` containing a.jpg and b.csv) to `MyCatalog.MySchema.MyVolume`, a compute engine can directly access the image using `MyCatalog.MySchema.MyVolume/a.jpg`.

> **Note:**
> A volume can only be created within a Volume-type catalog.

Models

A machine learning model registered into the Catalog from frameworks like MLflow. (Currently under development)

> **Note:**
> A model can only be created within a Model-type catalog.

Functions

A user-defined function (UDF) registered within the Catalog. It can return either a scalar value or a set of rows. (Currently in planning)

Creating a Data Catalog

1. Log in to the TCLake Console.

2. On the Data Catalog list page, ensure you are logged in as a user with the TCLake **Admin** role, then click **Create Data Catalog**.
3. Configure the following parameters in the dialog:

Parameter	Description
Catalog Name	Required. A globally unique identifier (no duplicates allowed). Must be 1 to 64 characters long and can only contain letters, digits, and underscores.
Description	Optional.
Storage Class	Currently, only the Standard storage class is supported.
Enable Multi-AZ Redundancy	**Notes:** This feature is currently in limited release. To request access, please submit a ticket. Disabled by default. When enabled, data is redundantly stored across multiple Availability Zones (AZs) within the same region, providing intra-city disaster recovery capabilities. Please note: Once enabled, Multi-AZ redundancy cannot be disabled. While it significantly enhances data reliability, it also incurs higher storage usage fees. We recommend enabling this feature only if your workloads demand elevated data reliability.

4. Review and agree to the TCLake Pricing Overview, then click to create the catalog.

Viewing a Data Catalog

In the left navigation pane, select **Data Catalog**. Using the catalog tree browser, you can select and explore specific catalogs along with their underlying levels, such as Schemas and Tables.

Editing a Data Catalog

1. On the Data Catalog list page, find the target catalog and click **Edit** in the Actions column.
2. In the configuration dialog, modify the catalog settings as needed.

Deleting a Data Catalog

On the Data Catalog list page, find the target catalog and click **Delete** in the Actions column.

> **Note:**
> To prevent accidental data loss, for built-in catalogs (Lakehouse, Volume, and Model) hosted in TCLake, you must manually delete all underlying metadata resources—except the default schema—before you can delete the catalog itself.

Schema

Last updated：2026-05-15 10:25:45

A Schema (often synonymous with a database) is a second-level metadata object within the three-level namespace of TCLake. This guide explains how to create a Schema via the TCLake console. Alternatively, you can create schemas directly using compute engines such as EMR or DLC connected to TCLake.

Creating a Schema

1. Log in to the TCLake Console.
2. On the Data Catalog list page, click a catalog name to enter its details page.

> **Note:**
> Creating schemas for Connection-type catalogs is currently not supported via the console.

3. Click Create Schema, configure the following parameters, and click OK:

Parameter	Description
Name	Required. Must be unique within the same catalog. Cannot start with a number. Maximum length is 64 characters. Allowed characters: letters, digits, and underscores (_).
Description	Optional.

Viewing a Schema

In the Schema list, click a schema name to view its basic information (e.g., name, description, creator, owner, creation time, and last metadata update time) as well as the resources it contains.

Deleting a Schema

1. In the Schema list, click Delete in the Actions column.
2. In the confirmation dialog, click OK.

> **Notes:**
> 1. To prevent accidental data loss, you must manually delete all metadata resources within a schema before deleting the schema itself.
> 2. A default schema is automatically created within each Lakehouse catalog and cannot be deleted.

Data Table
TCIceberg Table

Last updated：2026-05-15 10:25:45

Table Types

TCIceberg tables are a batch-stream unified table format developed by Tencent Cloud based on Apache Iceberg. They include the native Apache Iceberg V1/V2 table versions, and extend upon them with a Mix version that supports near real-time lakehouse construction capabilities. All versions support the Apache Iceberg Rest API specification protocol and can be read from and written to using the Apache Iceberg Client:

TCIceberg Table Version:

- It supports the native Apache Iceberg V1/V2 versions and is fully compatible with Apache Iceberg syntax operations.
- The TCIceberg extended Mix table version consists of two Iceberg tables: Base Store and Change Store. It supports reading streaming incremental data via CDC and provides Merge On Read read-time merging to ensure data latency in data analysis scenarios. Both tables can be read from and written to directly using the Apache Iceberg Client.

Table Data Management: TCIceberg table data is managed by TCLake, which includes metadata and data files. Users do not need to concern themselves with the underlying storage system. When a table is deleted, its metadata and data are deleted together.

Data Optimization: The data optimization service is managed by TCLake, including small file merging, expired snapshot cleanup, orphaned file cleanup, and more.

Creating a table

1. Log in to the TCLake Console.
2. On the Data Catalog list page, select a LakeHouse catalog, click the **catalog name**, and go to the catalog details page.
3. In the Schema list, click the **Schema name** to go to the table list page.
4. In the table list, click **Create Table**.
5. Configure the following parameters and click **OK**.

Parameter	Description
Table Name	Required. Must be unique within the same catalog. Cannot start with a number. Maximum length is 64 characters. Allowed characters: letters, digits, and underscores (_).

Table Format	Select TCIceberg.
Table Version	Required. V1/V2 are native Apache Iceberg versions. The Mix version is Tencent Cloud's extended batch-stream unified format, optimized for near real-time lakehouse scenarios.
Description	Optional.
Columns	Define the table columns, including column name, type, data type, primary key status, and description.
Partitioning	Disabled by default. When enabled, you can define partition columns and transform strategies for the table.
Custom Properties	Optional. You can add custom table properties. For a list of supported configuration properties, see the official Apache Iceberg documentation.

Viewing a Table

1. Go to a LakeHouse Catalog. In the Schema list, click the **Schema name** to view the table list.
2. In the table list, click the **table name** to view its columns, partition columns, and properties.
3. Click the **Data Optimization** tab to configure a dedicated data optimization policy for the TCIceberg table.
4. Click the **Version History** tab to view the snapshot version history of the TCIceberg table.

Deleting a Table

> **Note:**
> After a TCIceberg table is deleted, the managed data is retained by default for 7 days and is automatically deleted after that period. Please proceed with caution.

1. In the Schema list, click the **Schema name** to view the table list.
2. In the table list, click **Delete** in the Actions column.
3. In the confirmation dialog, click OK.