数据湖计算

快速入门

产品文档

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

文档目录

快速入门
- 新用户开通全流程
- DLC 数据导入指引
- 一分钟入门 DLC 数据分析
- 一分钟入门 DLC 权限管理
- 一分钟入门分区表
- 一分钟入门 UDF 函数
- 开启数据优化
- 跨源分析 EMR Hive 数据

版权所有：腾讯云计算（北京）有限责任公司

快速入门
新用户开通全流程

最近更新时间：2024-07-31 17:22:59

前期准备

注册账号

注意：

仅针对管理员，可能会涉及到以下操作。

1. 进入 数据湖计算 DLC 控制台，单击**前往访问管理**，授权数据湖计算。

2. 在角色管理中，单击**同意授权**。

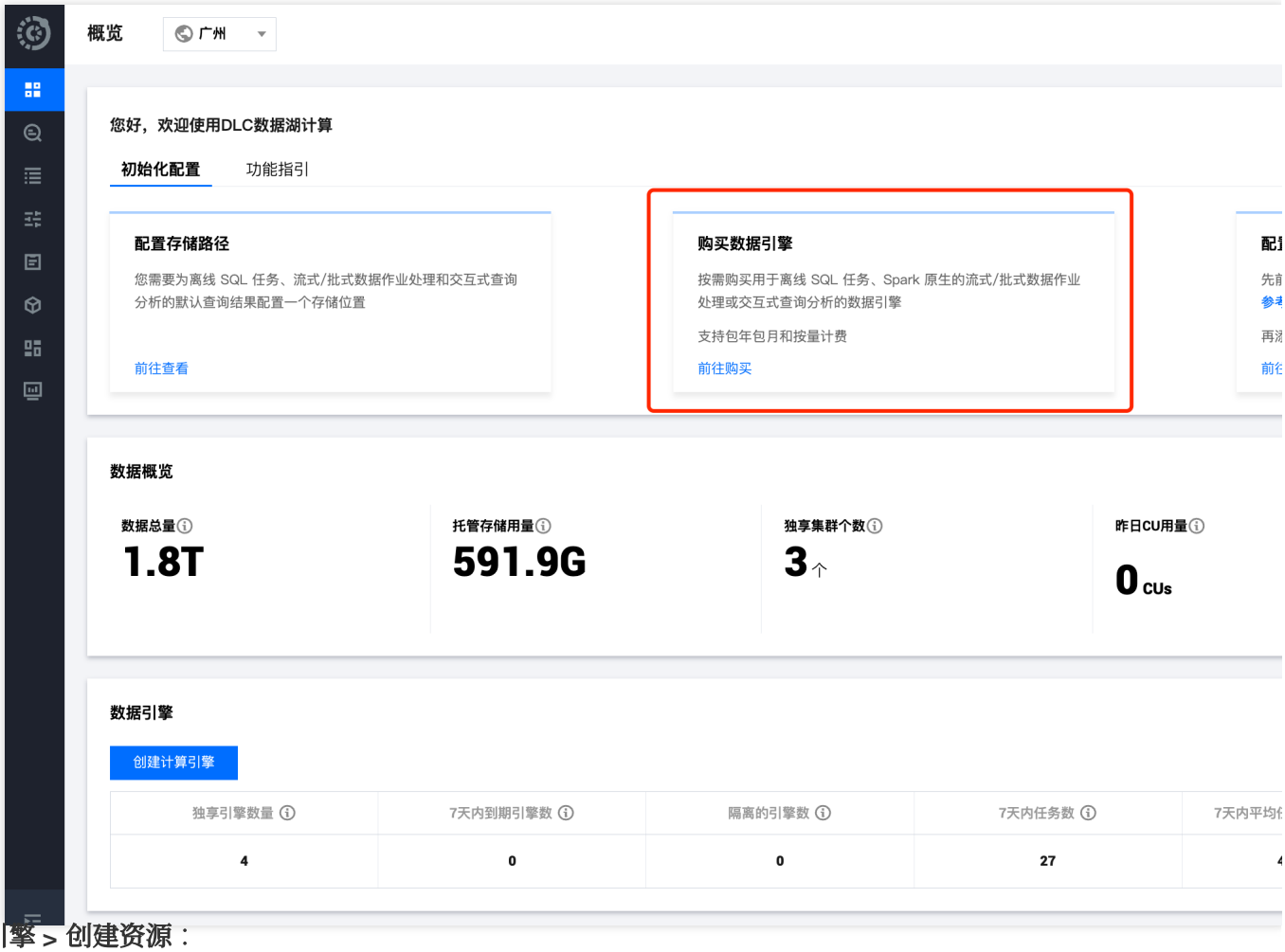
购买引擎

注意：

需要在访问管理 CAM 中拥有财务权限。

1. 在 数据湖计算 DLC 控制台 上，可在**概览**页及**数据引擎**页中前往引擎购买页购买引擎。

概览页 > 初始化配置 > 购买数据引擎：



数据引擎 > 创建资源：

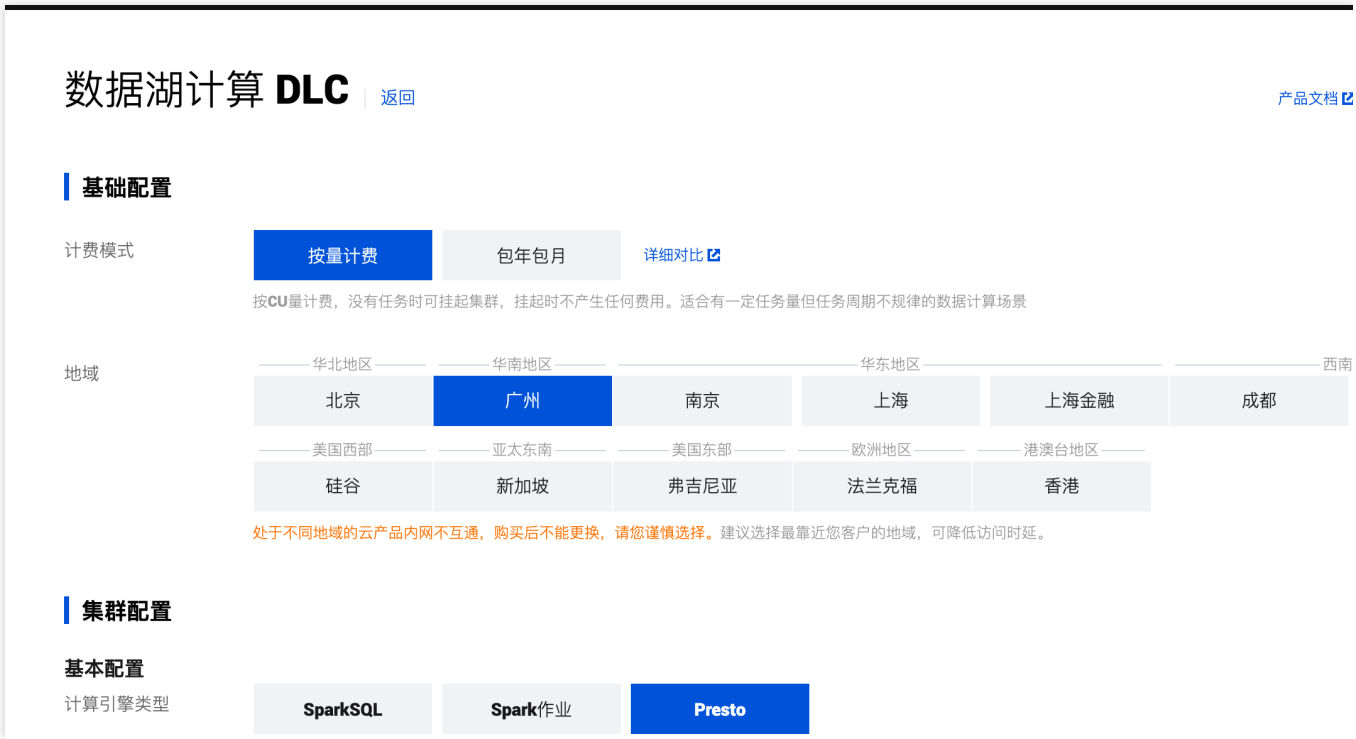
2. 在购买页中选择需要购买的引擎类型：

说明：

SparkSQL：适用于稳定高效的离线 SQL 任务。

Spark 作业：适用于 Spark 原生的流式/批式数据作业处理。

Presto：适用于敏捷、快速的交互式查询分析。



注意：

6CUs的集群规模较小，建议仅用于测试场景，真实生产场景建议选择购买64CUs以上规格的集群。

引擎操作类权限自动授权

数据湖计算 DLC 支持默认开启引擎操作类权限，开启后，所有用户默认拥有该引擎的下列权限：

使用：使用该引擎进行任务执行。

操作：操作引擎的暂停、挂起。

监控：针对引擎的使用情况监控运维。

注意：

1. 关闭后，管理员默认继续拥有引擎各权限，普通用户需管理员在权限管理页添加权限。
2. 原有普通用户拥有权限不受影响，可前往权限管理页删除。
3. 后续新建普通用户无使用权限，需在权限管理页手动添加。

如何开启、关闭自送授权引擎权限

引擎默认开启/关闭操作类权限入口有两个：

入口一：引擎购买页 > 高级配置项

入口二：进入 数据引擎 页面，单击编辑授权引擎权限。

设置引擎权限后，单击**确定**。

团队账号开通

如果您有多账号协同使用产品的诉求，可按照以下操作建议进行开通：

1. 每个地域的权限不通用，不同地域需要分别设置对应权限。
2. 数据湖计算 DLC 的权限快捷开通入口：

开通子账号访问数据湖计算 DLC 的权限，请前往 CAM 控制台 进行配置。

开通子账号对数据湖计算 DLC 产品内的数据与引擎读写权限，请前往 DLC 控制台 配置。

开通子账号与访问 DLC 权限

主账户默认拥有 DLC 全部操作权限。主账号通过访问管理 CAM 将 DLC 的访问权限授予给子用户，使子用户拥有对应的 DLC 操作权限：**QcloudDLCFullAccess（DLC 的全部操作权限）**。

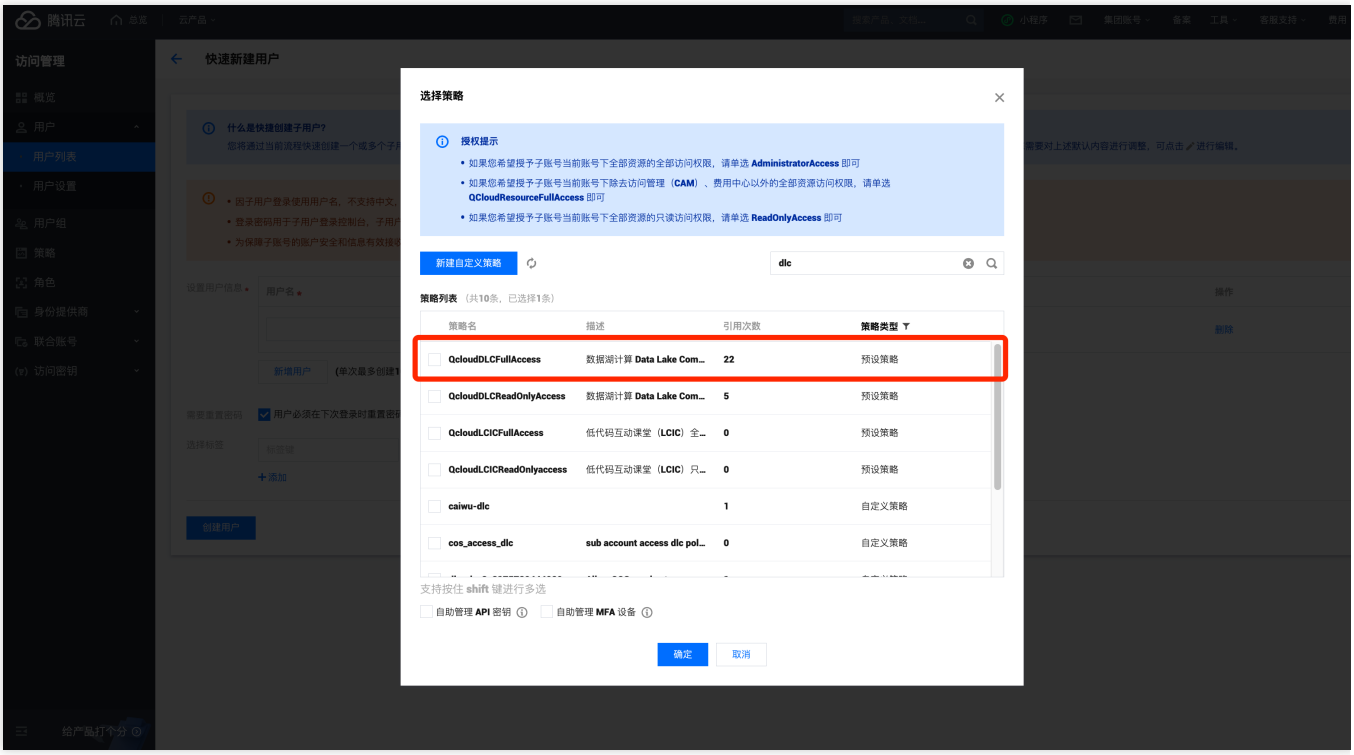
操作步骤

1. 登录CAM 控制台 创建子用户，详细操作请参见 创建子账号并授权 。

2. 为子账号添加预设策略：**QcloudDLCFullAccess（DLC 的全部操作权限）**。

可在用户列表搜索需授权的用户，单击**授权**。

策略列表中选择 **QcloudDLCFullAccess（DLC 的全部操作权限）**。



开通子账号在 DLC 中数据与引擎权限

添加用户至 DLC 权限管理

说明：

1. 请确认用户权限生效地域。
2. 关于用户划分：

管理员：拥有所有资源的权限。

普通用户：需要授予具体的权限，或者绑定工作组来获取权限。

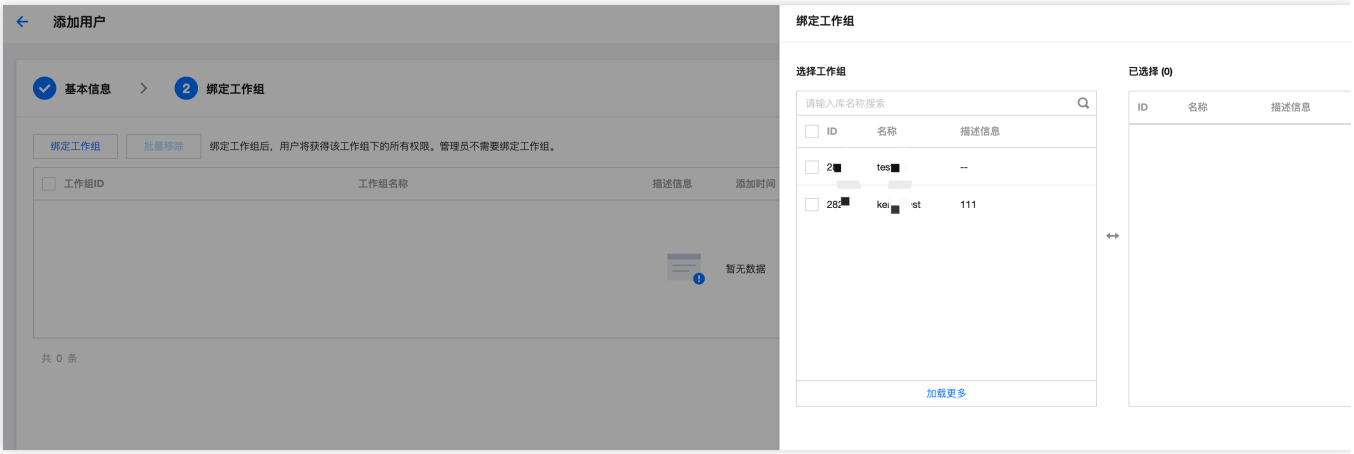
1. 登录 数据湖计算 DLC 控制台，进入权限管理页面，选择**对应服务地域**，进入权限管理页。单击**添加用户**。

2. 通过子用户的 **CAM ID** 将该账号添加进 DLC 内进行管理。用户类型请按需选择。

3. 绑定用户至工作组（非必选步骤）。

说明：

如需要管理多个用户的使用权限，可通过绑定工作组，待创建完工作组后，再进行添加，具体创建流程请参见 一分钟入门 DLC 权限管理。



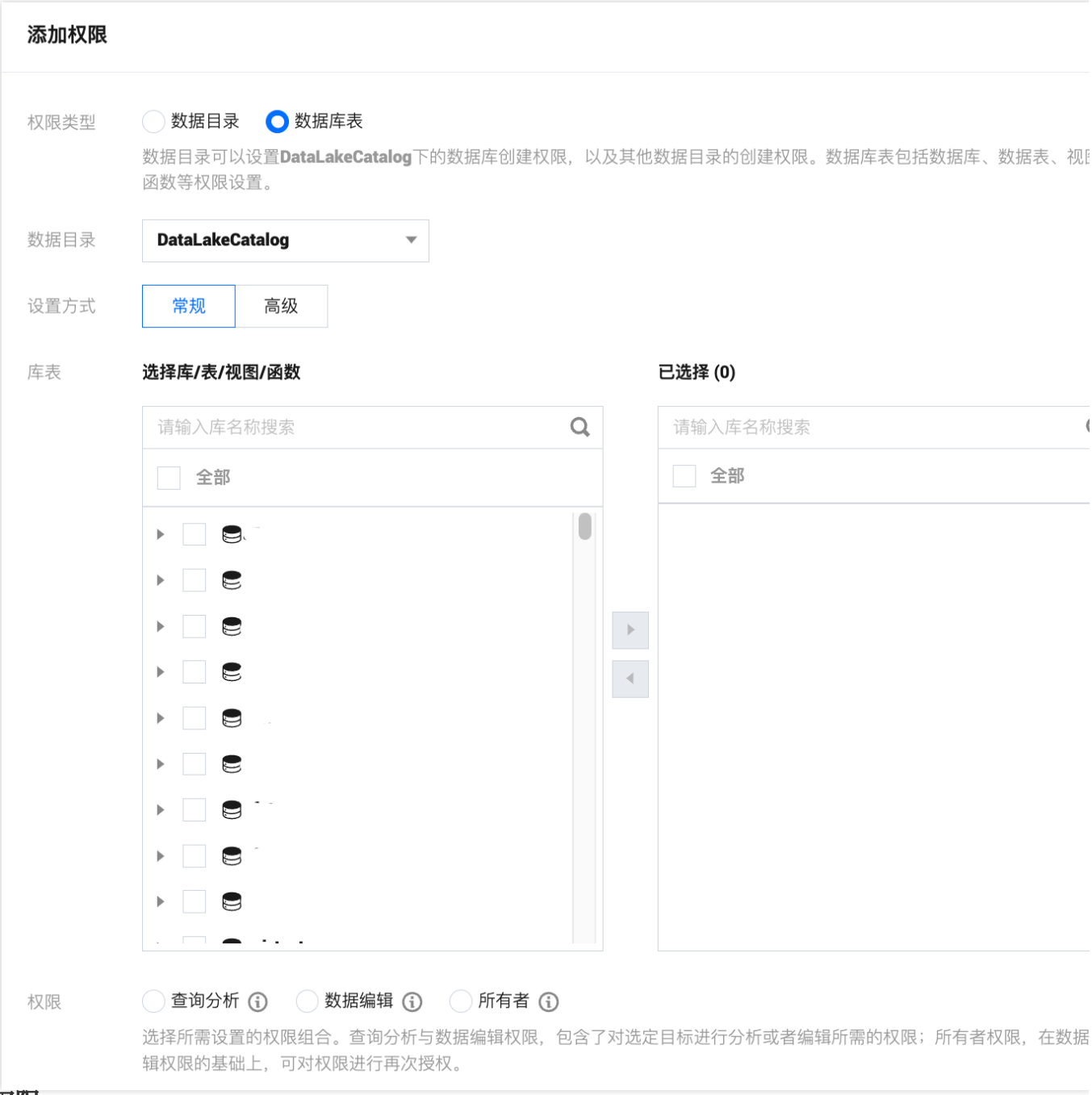
添加引擎与数据权限

创建用户或工作组后单击列表中的授权操作，为工作组添加权限，包括数据权限和引擎权限。

数据权限

数据目录权限：包括在数据目录下创建数据库和创建数据目录两种权限。

数据库表权限：可授予库表级别的细粒度权限，包括对库、表、视图、函数的查询和编辑等权限。



引擎权限

根据用户或工作组的使用场景，勾选引擎的权限策略。

说明：

使用：使用该引擎进行任务执行。

修改：修改引擎的配置参数，如引擎的规格变配。

操作：操作引擎的暂停、挂起。

监控：针对引擎的使用情况监控运维。

删除：删除引擎。

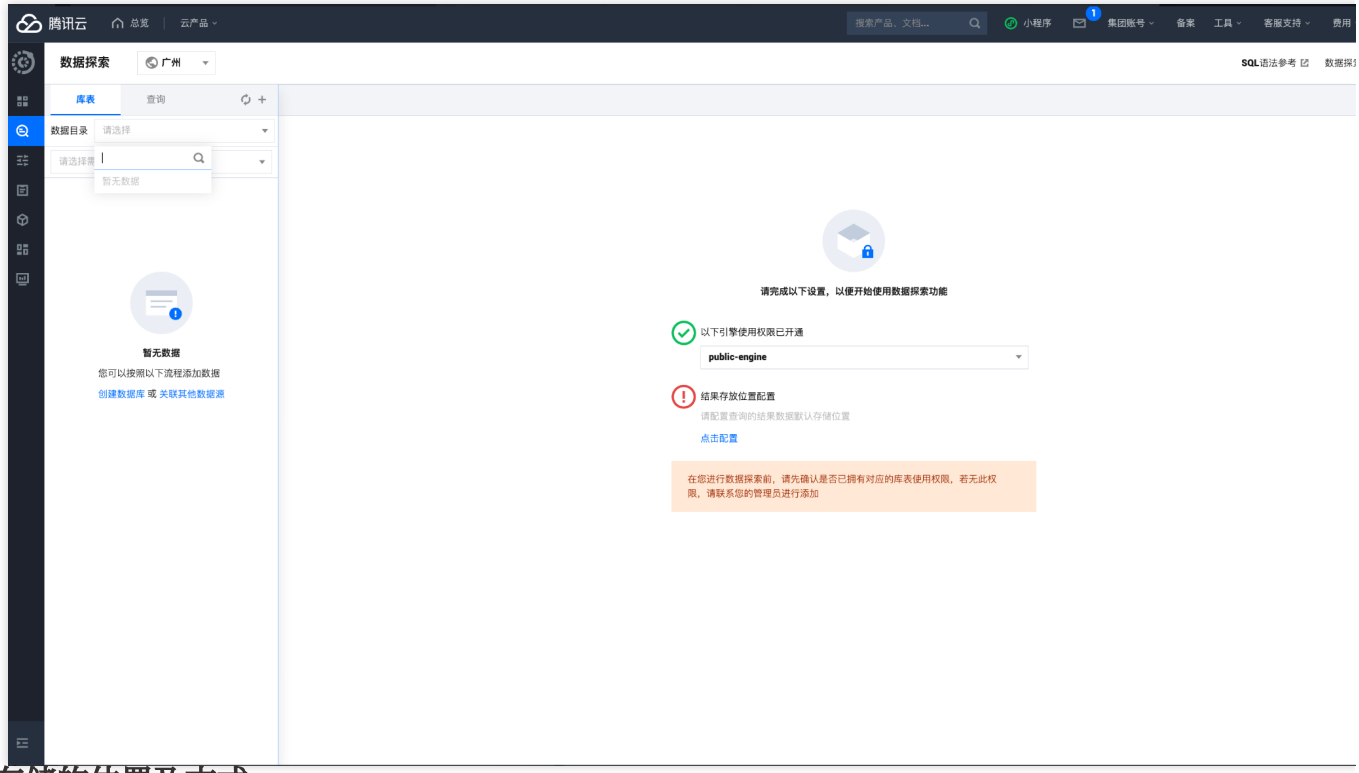
可授权：勾选后，该子用户或工作组下的所有成员拥有对引擎的授权权限。

配置结果存储位置

在使用数据探索功能前，需要先配置查询结果路径，配置完毕后，查询结果会保存到该 COS 路径下或 DLC 的托管存储。详细操作请参见 配置查询结果路径操作指南。

配置结果存储位置

进入数据湖计算 DLC 控制台，选择数据探索功能，选择结果存放位置配置，单击**点击配置**。



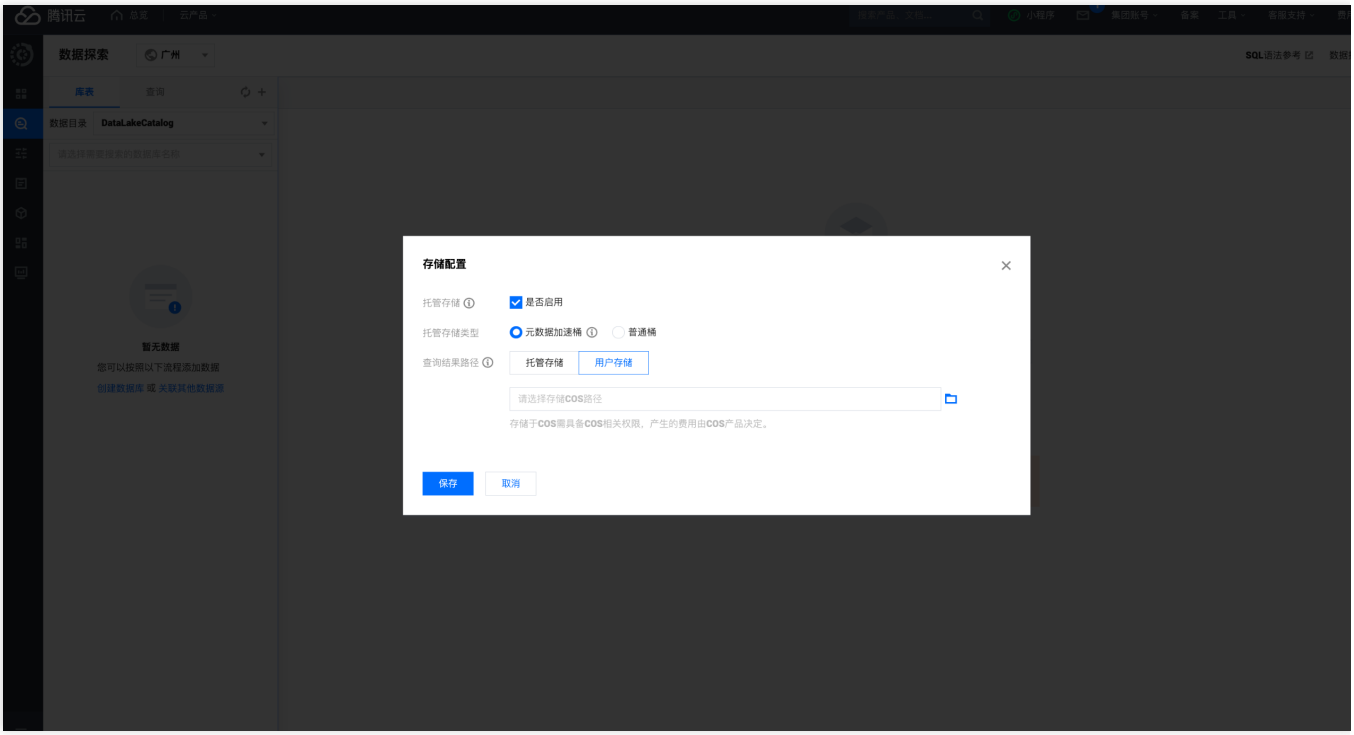
选择存储的位置及方式

说明：

元数据加速桶：在当前地域，可更好的提升查询分析性能。 内表可直接开启，外表需要确认引擎权限是否允许开启。

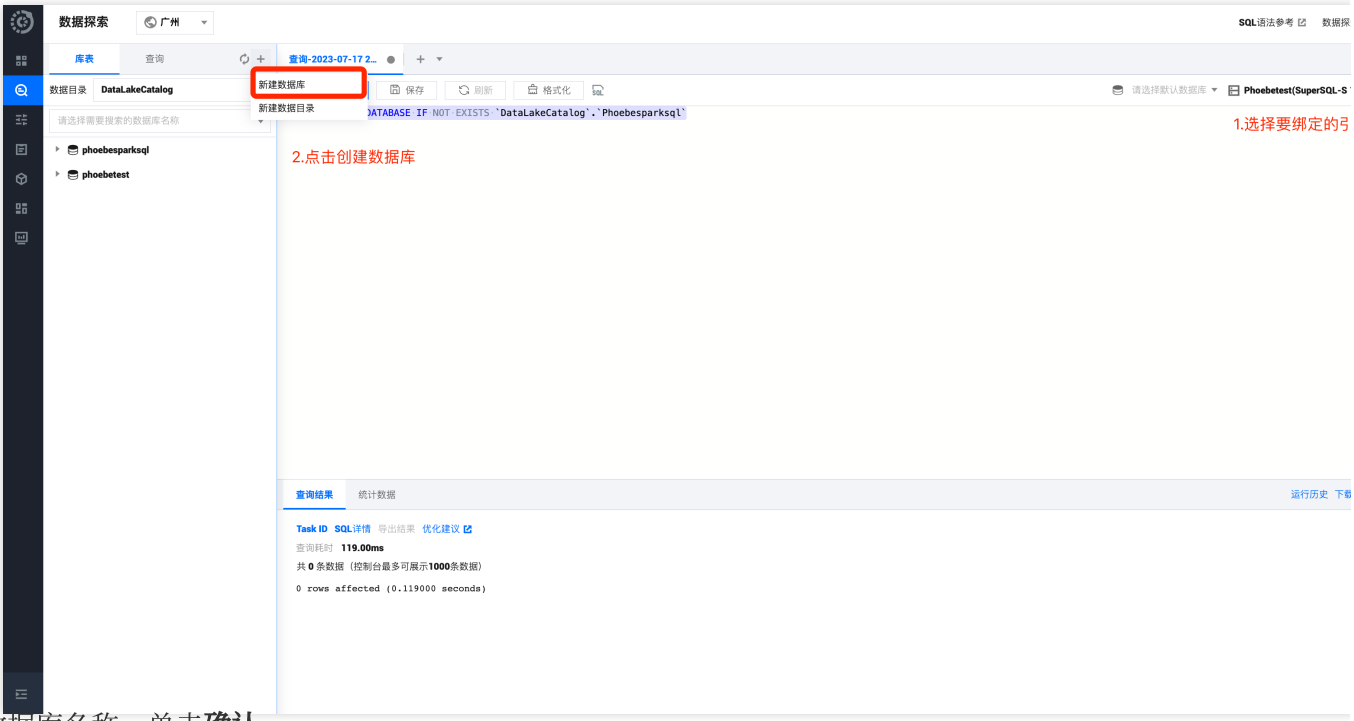
请注意：共享引擎无法绑定元数据加速桶。当用户选择用户存储路径时，独享引擎需要先绑定元数据加速桶后，再做查询才能生效。

用户存储：用户存储为您在 COS 上的存储桶路径。

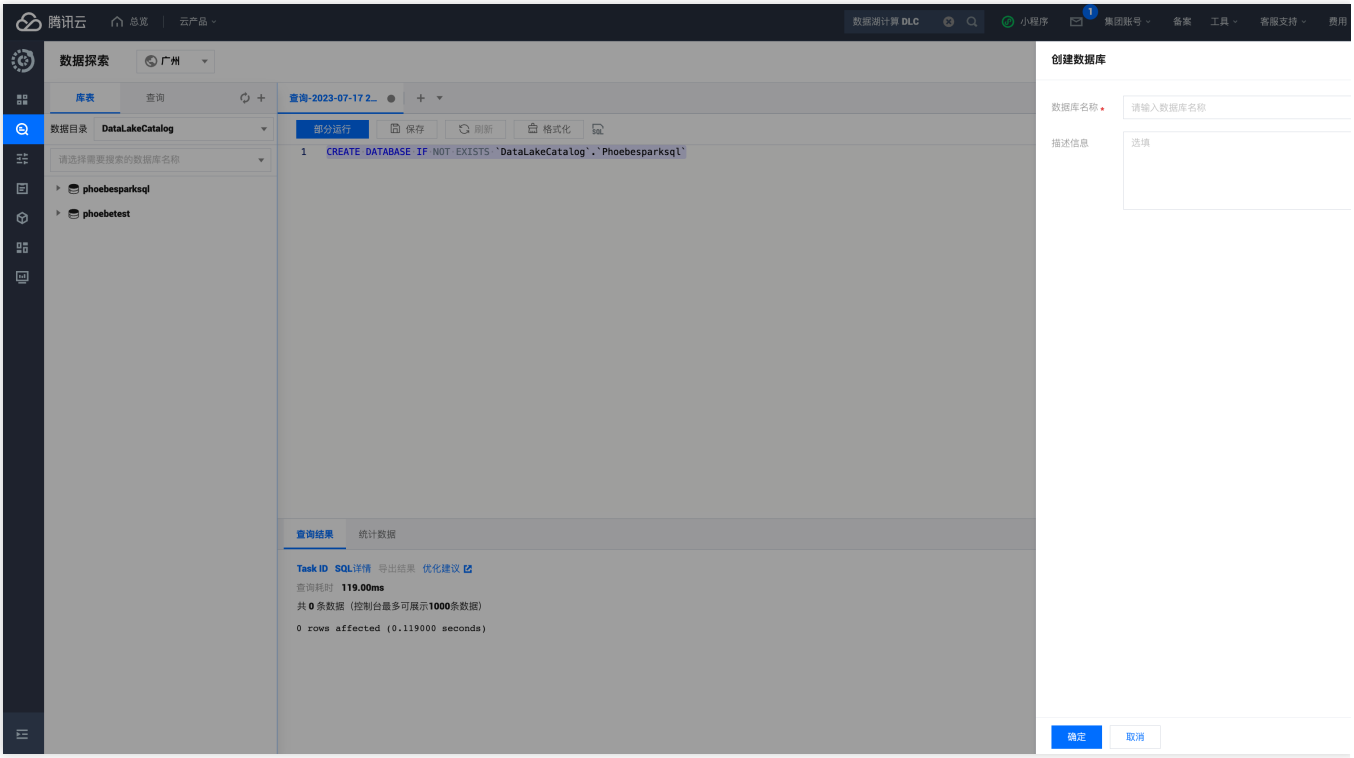


创建库

创建数据库前，选择使用消耗的引擎。

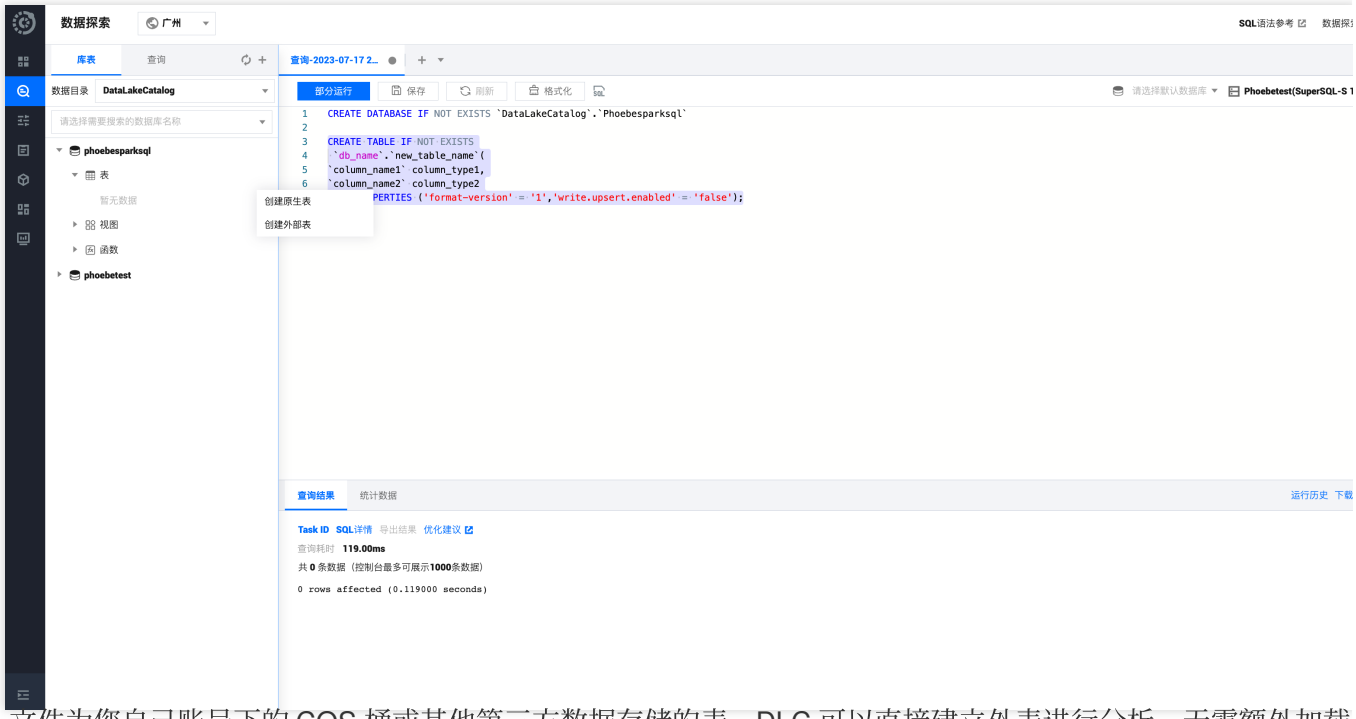


输入数据库名称，单击**确认**。

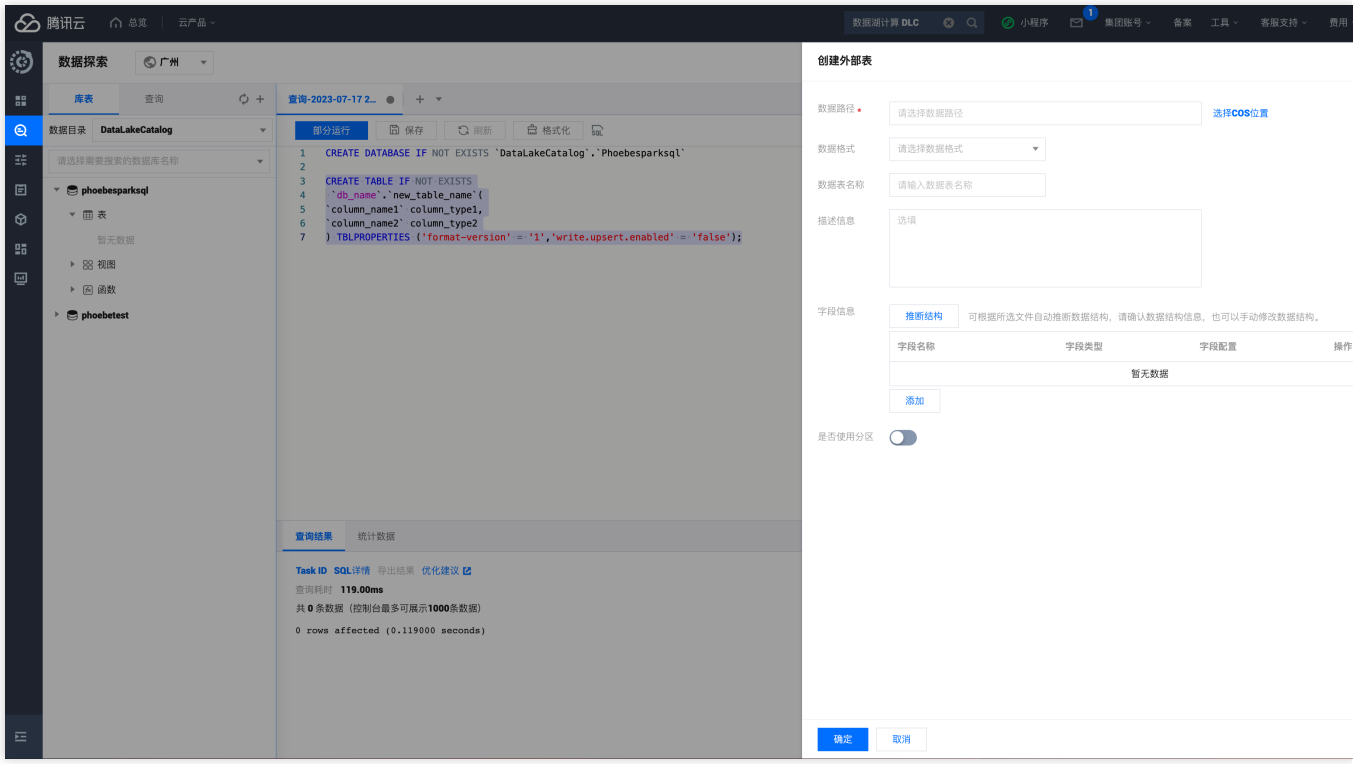


创建表

原生表：是您存放于 DLC 托管存储上的表，默认为 Iceberg 格式。使用原生表无需关注 Iceberg 底层文件，而且具备数据优化等能力帮助构建数据湖。



外表：文件为您自己账号下的 COS 桶或其他第三方数据存储的表。DLC 可以直接建立外表进行分析，无需额外加载数据。

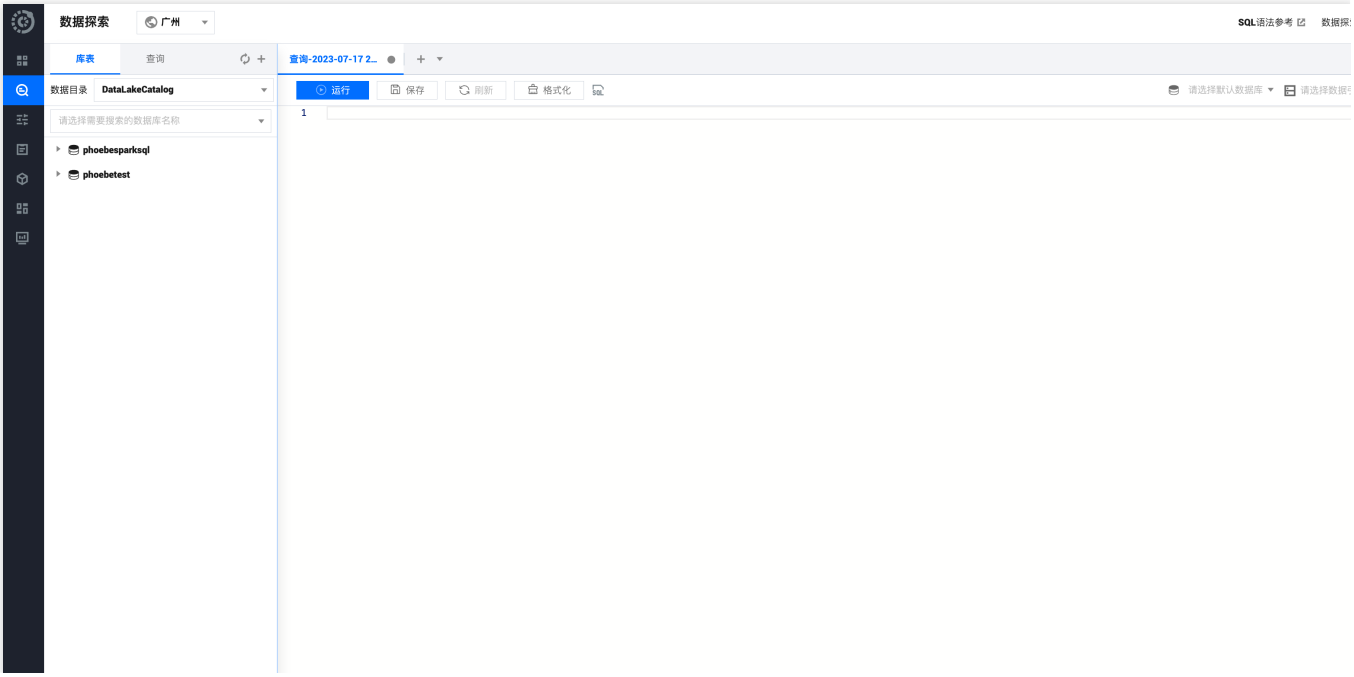


注意：

创建原生表后，需要刷新浏览器才能使用。

数据查询

进入数据湖计算 DLC 控制台-数据探索。在分析页面可创建 SQL 查询。功能支持全部运行、部分运行，下载结果，物化视图。



DLC 数据导入指引

最近更新时间：2024-07-31 17:23:18

通过 COS 进行外部表数据导入

数据湖计算 DLC 支持在不迁移数据的情况下，直接对对象存储 COS 上的数据进行查询分析，所以您只需将数据导入对象存储 COS 即可开始使用数据湖计算 DLC 进行无缝数据分析，实现数据存储与计算完全解耦。目前支持上传 orc、parquet、arvo、json、csv 和文本文件等多种格式。

目前，对象存储 COS 提供了丰富的数据导入方式，您可根据自身情况选择以下方式进行数据导入。

登录 对象存储 COS，直接进行文件上传，相关操作步骤可参见 上传对象。

通过对象存储 COS 服务提供的多种上传工具将数据进行导入，工具支持列表可参见 工具概览。

通过对象存储 COS 服务提供的 SDK 或 API 对数据进行导入，服务相关说明可参见 上传接口文档。

如您需要将日志服务 CLS 内的日志导入进行分析，可按照分区直接将日志投递到对象存储 COS 后通过数据湖计算 DLC 直接进行分析查询。相关操作可参见 使用 DLC（Hive）分析 CLS 日志。

如您需要将其他云服务（如数据库 CDB 等）数据导入对象存储 COS ，可以使用数据集成 DataInLong 进行导入。创建数据同步链路时，数据源选择需导出的云服务，目的端选择对象存储 COS 即可完成数据的导入。

您在进行数据导入时，如遇到问题，可以通过 提交工单 咨询我们为您提供解决方案。

将数据导入到 COS 后，您可通过数据湖计算 DLC 控制台、API、SDK 进行 SQL 查询，实现建表、分析、结果导出等操作，详细操作可参见 一分钟入门 DLC 数据分析。

数据导入原生表

为了提供更好的数据查询性能，数据湖计算 DLC 也支持将数据导入到原生表后进行查询分析。 DLC 原生表基于 Iceberg 表格式排布数据，在导入数据过程中对数据进行优化，如您有以下使用场景，建议使用原生表进行数据查询分析。

基于数据仓库分析场景下，希望借助 Iceberg 索引获得更好的分析性能。

对数据有更新需求，希望通过 DLC 服务通过 SQL 或数据作业实现 UPSERT 操作。

数据通过数据集成 DataInLong、Flink、流计算 Oceanus、Spark Streaming 实时写入更新，读写同时进行，需要事务性保障的数据处理业务。

希望使用 Iceberg 表相关特性，如时间旅行、多版本快照、隐藏式分区、分区进化等高级数据湖特性。

如您需要将数据导入到原生表，可根据自身情况选择以下方式进行数据导入。

通过 数据湖计算 DLC 控制台 直接导入。

注意

通过控制台进行数据导入时，存在一定使用限制，主要用于快速测试，不建议生产使用。

如您的原始数据在 MySQL、Kafka 等业务中，需要将 MySQL binlog 、消息中间件数据分钟级实时写入/更新到 DLC ，可通过数据集成 DataInlong 的实时导入能力实现。或通过流计算 Oceanus 、Flink写入，如需操作指导可通过 工单 联系我们。

如原始数据在 MySQL、Kafka、MongoDB 等数据业务中，可通过数据集成 DataInLong 的离线同步任务将数据转存到原生表。数仓建模过程中，将外部表作为原始数据贴源层，把数据转存到原生表的过程中，可以通过构建稀疏索引等方式结合业务重新排布数据分布，以获得优异的原生表查询分析性能，如需要指导可以 联系我们。

通过 SQL 语法 SELECT INSERT 的方式将外部表的数据查询后写入原生表。如：在数据湖计算 DLC 创建一张与外部表表结构相同的原生表后，通过 SparkSQL 引擎执行 SQL 语法完成转存。语法示例如下：

```
--- 外部表表名：outtertable，原生表表名：innertable
```

```
insert into innertable select * from outtertable
```

您在进行数据导入时，如遇到问题，可以通过 提交工单 咨询我们为您提供解决方案。

多数据源联邦查询分析

如您不想将数据导出到对象存储 COS 或数据湖计算 DLC 的原生表中，数据湖计算 DLC 同时提供了数据联邦查询分析能力，支持您在不搬迁数据的情况下，通过 SQL 快速关联分析多个数据源的数据，目前已支持 MySQL、SQLServer、clickhouse、postgreSQL、EMR on HDFS、EMR on COS 等多种数据源。
使用联邦分析时需要数据源与数据引擎在同一个网络下，网络打通、管理可参见 引擎网络配置。
通过数据湖计算 DLC 联邦分析 EMR 的数据时，查询性能将会持平甚至超过 EMR 的性能，适用于生产环境，可以在不搬迁 EMR 服务的情况下充分利用 DLC 的全托管弹性能力降本增效。
联邦分析可快速联合多个数据源的数据进行分析，为数据洞察、快速分析提供了便利的方式，同时依托于 DLC 的全托管弹性能力，可有效降低使用成本。同时也可以支持使用 INSERT INTO/INSERT OVERWRITE 语法将联邦数据写入到 DLC 原生表，完成数据导入。
联邦分析其他数据源时，由于计算过程中需要将数据同步到 DLC 侧进行分析，相比原始数据源直接查询，性能存在一定的损耗，如对查询性能要求较高，可将数据导入到原生表后进行分析，操作方式可见 数据导入原生表。

一分钟入门 DLC 数据分析

最近更新时间：2024-07-17 14:48:55

使用数据湖计算 DLC，您仅需一分钟即可完成对象存储 COS 上的数据分析查询，目前支持 CSV、ORC、PARQUET、JSON、ARVO、文本文件等多个格式。

前置准备

在进行查询前，您需要开通 DLC 内部权限及配置查询结果路径。

步骤1：设置必要 DLC 内部权限

说明：

如果用户已经有权限，或者为主账户管理员，可忽略此步骤。

若您是首次登录的子账号，除了必要的 CAM 授权，还需要请任意 DLC 管理员或主账号管理员在 DLC 控制台左侧权限管理菜单，为您授予必要的 DLC 权限（详细权限说明请参见 DLC 权限概述）。

1. 库表权限：可授予对应的 catalog、database、table，view 等读写操作权限。
2. 引擎权限：可授予计算引擎的使用、监控、修改等权限。

说明：

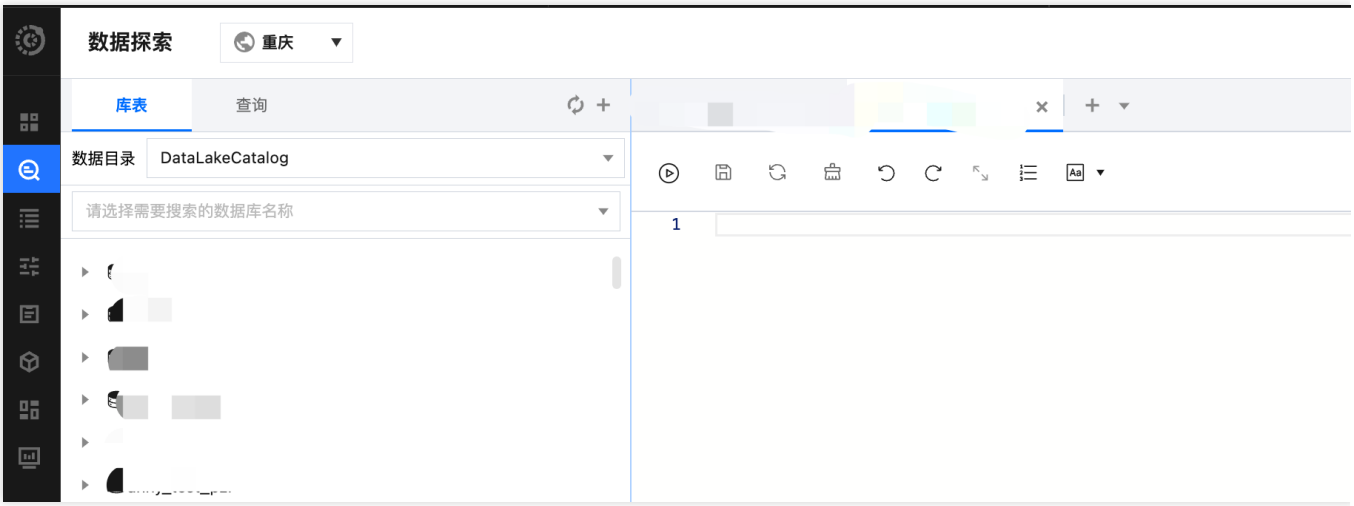
系统会默认为每个用户开通基于 presto 内核的共享 public-engine，方便您可以快速试用，无需先购买独享集群。

详细权限授予步骤请参见 子账号权限管理。

步骤2：配置查询结果路径

首次使用 DLC，需要先配置查询结果路径，配置完毕后，查询结果会保存到该 COS 路径下。

1. 登录 数据湖计算 DLC 控制台，选择服务地域。
2. 左侧导航菜单进入数据探索。
3. 在库表页面下，单击存储配置配置查询结果路径。



配置查询结果路径。如 DLC 内部存储，将 SELECT 查询结果存储在 DLC 产品内的存储空间，存储底层为对象存储。结果存储有效期为36小时。

如果您的账号中没有可用的 COS 存储桶，可在对象存储控制台创建。

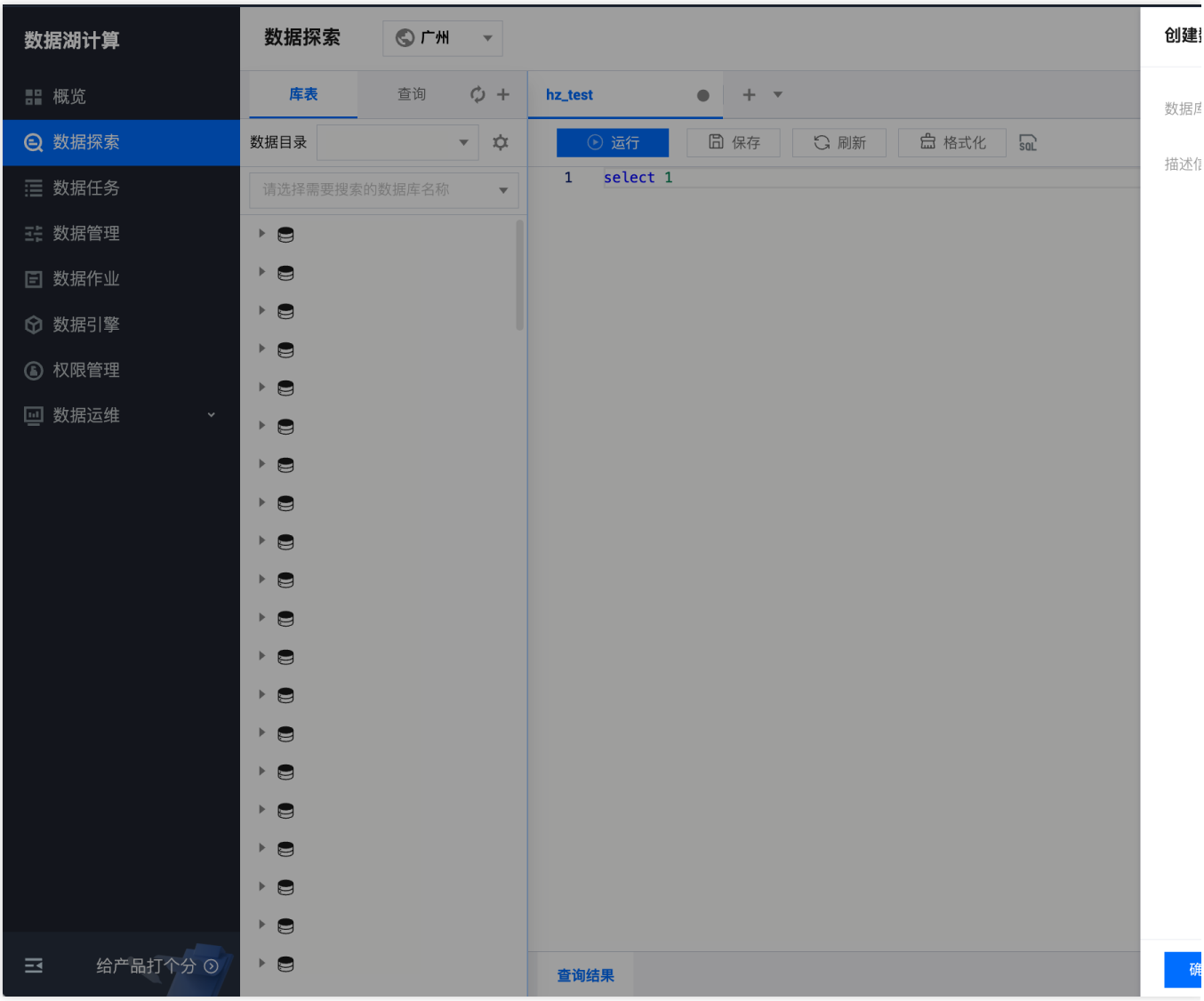
分析步骤

步骤1：创建数据库

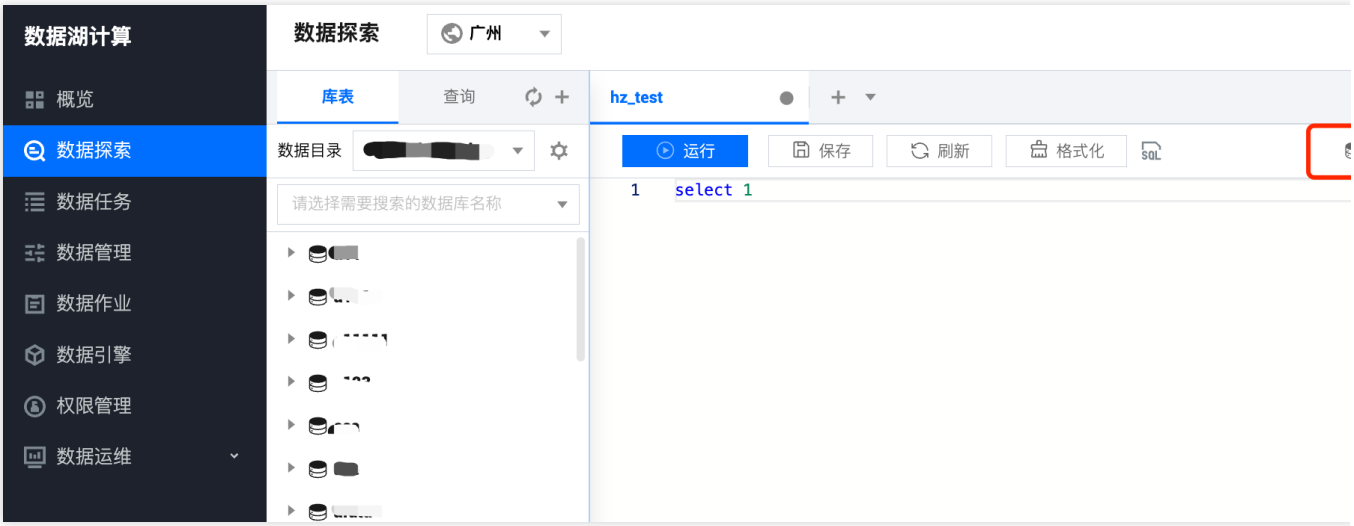
如果您对 SQL 语句熟悉，可直接在查询中编写 create database 语句，跳过创建向导。

1. 登录 数据湖计算 DLC 控制台，选择**服务地域**。
2. 左侧导航菜单进入**数据探索**。
3. 选择**库表**，单击“+”，选择**创建数据库**进行数据库新建。如下图所示：

填写数据库名称和描述信息。



4. 右上角选择执行引擎后，执行生成的 create database 语句，完成建库。



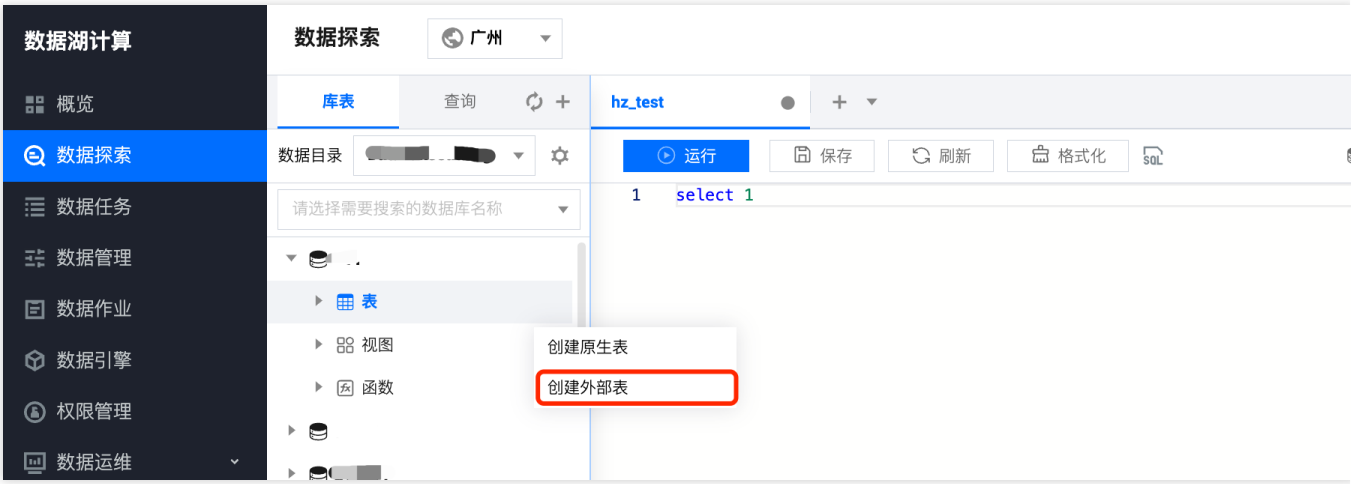
步骤2：创建外表

如果您对 SQL 语句熟悉，可直接在查询中编写 create table 语句，跳过创建向导。

1. 登录 数据湖计算 DLC 控制台，选择服务地域。
2. 左侧导航菜单进入**数据探索**。
3. 选择库表，选中当前创建的表后，右键单击，选择**创建外表**。

说明：

外表一般指数据文件放到您自己账号下的 COS 桶，DLC 可以直接建立外表进行分析，无需额外加载数据。基于外表的特性，例如在执行 drop table 等动作时，DLC 并不会删除您的原始数据，只会删除 table 的元信息。



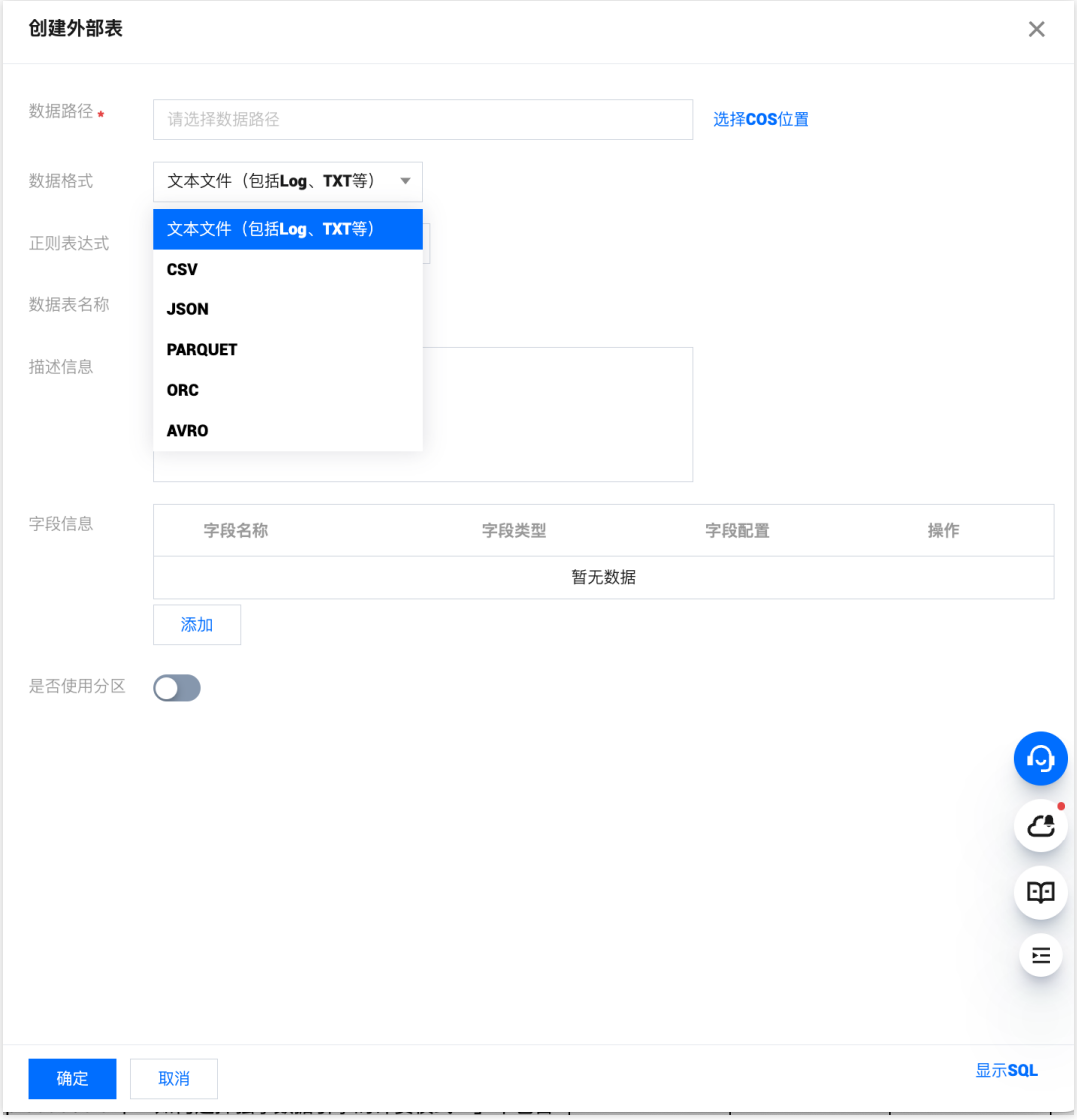
4. 按照向导生成创表语句，按照**基本信息 > 数据格式 > 编辑列 > 编辑分区**，完成各个步骤。

step1：选择数据文件存放的 COS 路径（路径必须是 COS 桶下的目录，不能直接建立到 COS 桶），此处也提供快速上传文件到 COS 的快捷方式。操作需具备 COS 相关的权限。

目前DLC支持创建：文本文件、csv、json、parquet、orc、orc、avrd。

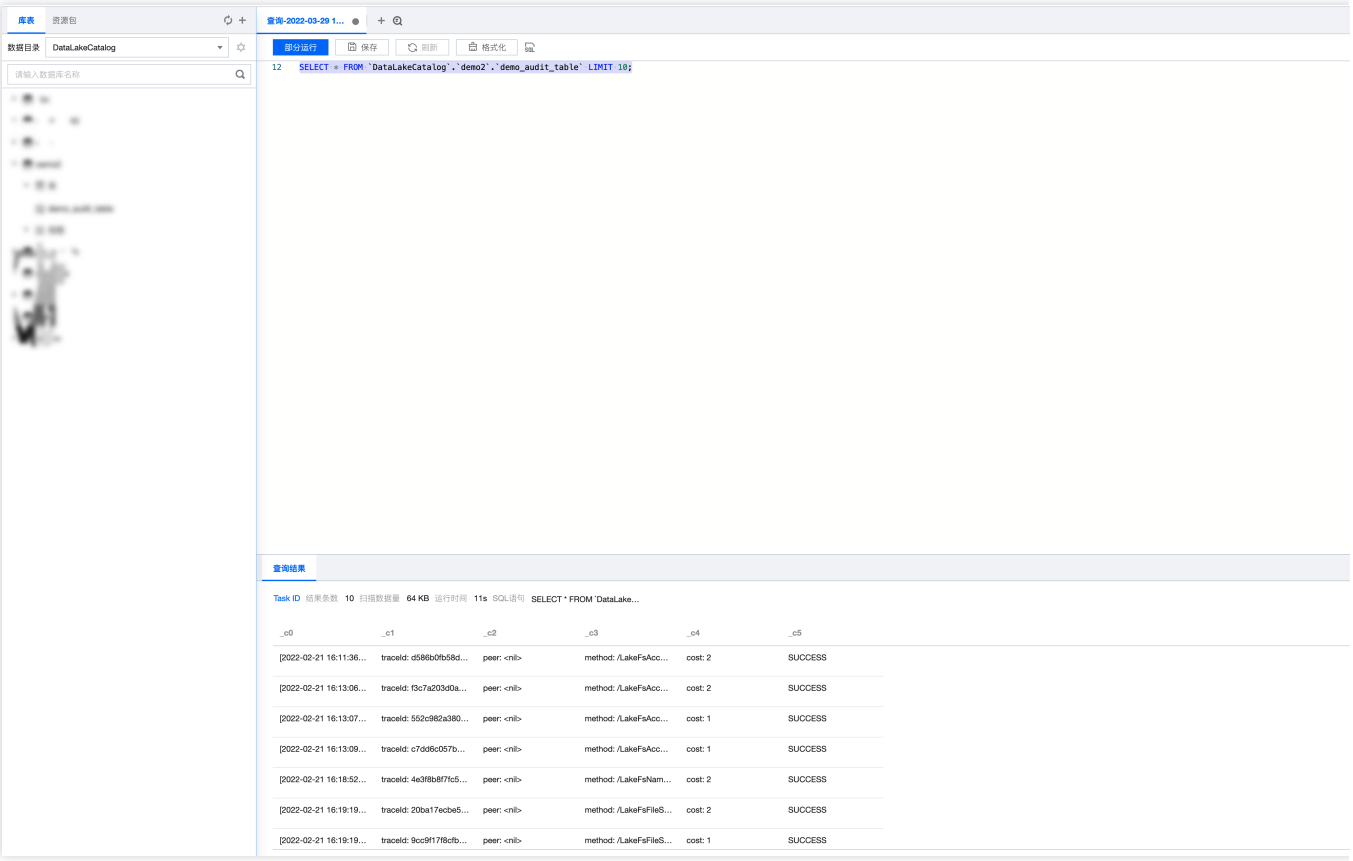
说明：

结构推断为建表辅助工具，不能保证100%正确，仍需您进行复查核对字段名、类型是否符合预期，根据实际情况编辑修改为正确的信息。



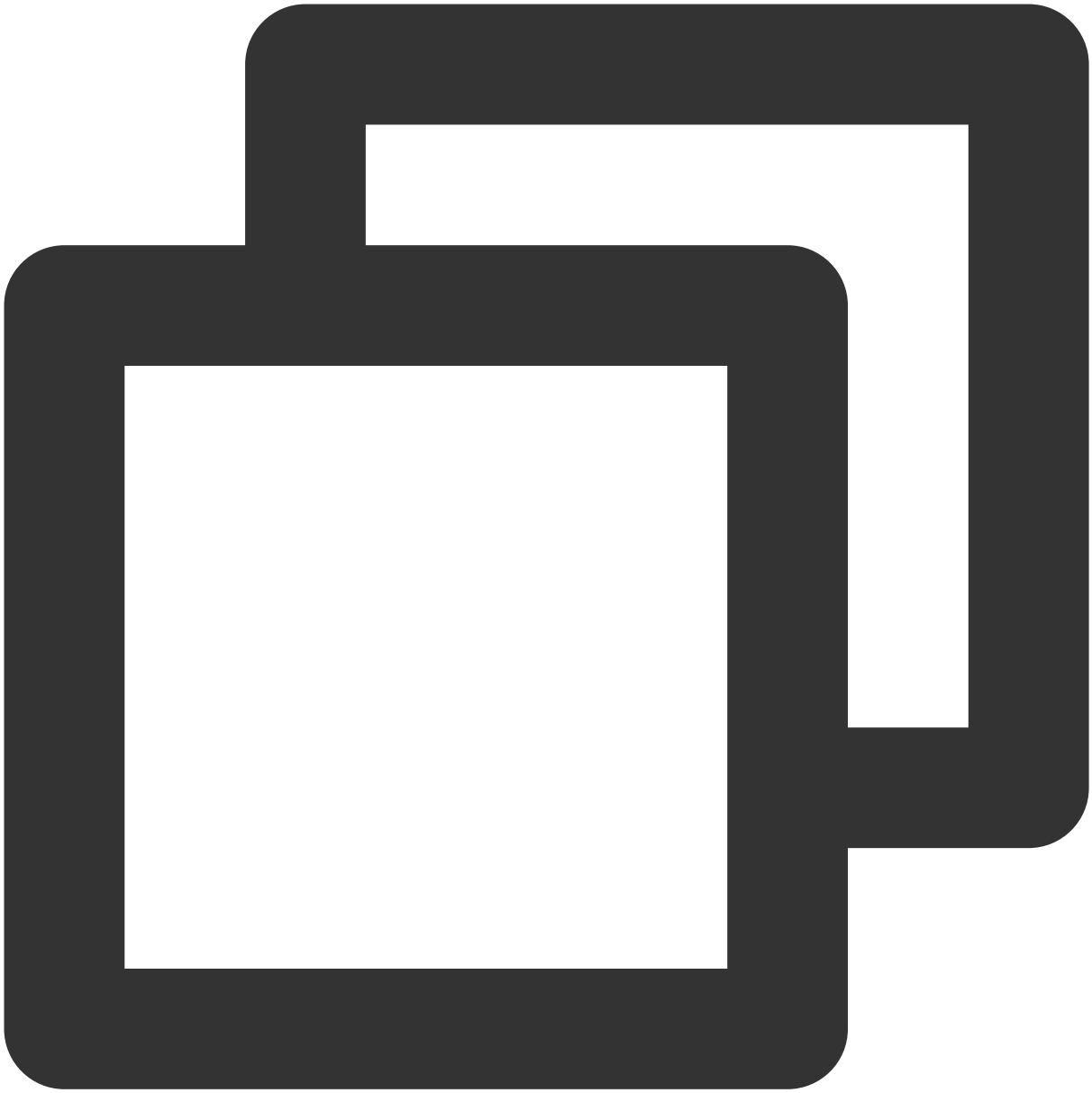
步骤3：执行 sql 分析

数据准备完备后，您就可以开始书写 SQL 分析语句，选择合适的计算引擎，开始数据分析。



示例

编写数据查询所有结果为 SUCCESS 记录的 SQL 语句，选择计算引擎后执行。



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

执行结果如下：

查询结果

Task ID 结果条数 10 扫描数据量 64 KB 运行时间 5s SQL语句 SELECT * FROM `DataLake...

_c0	_c1	_c2	_c3	_c4	_c5
[2022-02-21 16:11:36...	traceId: d586b0fb58d...	peer: <nil>	method: /LakeFsAcc...	cost: 2	SUCCESS
[2022-02-21 16:13:06...	traceId: f3c7a203d0a...	peer: <nil>	method: /LakeFsAcc...	cost: 2	SUCCESS
[2022-02-21 16:13:07...	traceId: 552c982a380...	peer: <nil>	method: /LakeFsAcc...	cost: 1	SUCCESS
[2022-02-21 16:13:09...	traceId: c7dd6c057b...	peer: <nil>	method: /LakeFsAcc...	cost: 1	SUCCESS
[2022-02-21 16:18:52...	traceId: 4e3f8b8f7fc5...	peer: <nil>	method: /LakeFsNam...	cost: 2	SUCCESS
[2022-02-21 16:19:19...	traceId: 20ba17ecbe5...	peer: <nil>	method: /LakeFsFileS...	cost: 2	SUCCESS
[2022-02-21 16:19:19...	traceId: 9cc9f17f8cfb...	peer: <nil>	method: /LakeFsFileS...	cost: 1	SUCCESS

一分钟入门 DLC 权限管理

最近更新时间：2024-07-17 14:50:49

在使用 数据湖计算 DLC 的过程中，如果您需要给企业中的员工设置不同的访问权限，以达到不同员工之间的权限隔离，可以使用权限管理功能，针对用户、工作组进行精细的权限管理。

注意：

权限的策略与产品的使用有高度关联关系，建议在正式使用产品功能前，管理员提前配置好工作组、子用户等角色的策略。

1. **不同地域，需要管理员重新配置该地域下的 DLC 的成员管理与权限管理。**

CAM 授权

数据湖计算 DLC 具备完善的数据访问权限机制。如果您有子账号的管理需求，请在访问管理控制台中授予对应子账号QcloudDLCFullAccess（**数据湖计算 Data Lake Compute(DLC) 全读写访问**）策略。

数据湖计算 DLC 产品具备粒度细化到数据表行级列级的权限，您无需担心此操作会越权。

用户与工作组

DLC 通过对用户授权和绑定工作组授权两种方式来管理用户权限。

工作组：DLC 可以将一批用户绑定到工作组，并授予该组数据、引擎等资源权限，来批量管理用户权限，在同一个工作组的用户具有相同的权限。

用户：CAM 中的用户，包括子账号、协作者账号。

说明：

当用户被赋予的权限与所在工作组权限不同时，两者权限取并集。

默认情况下，管理员创建的普通用户没有任何权限，需要将其加入工作组，并授予工作组相应的权限策略，才能使该工作组中的用户获得对应权限。

添加用户

数据湖计算 DLC 使用腾讯云账号 ID 作为用户默认 ID，用户类型分为管理员与普通用户，管理员拥有所有资源的权限，普通用户需要授予具体的权限，或者绑定工作组来获取权限。

1. 添加用户并绑定工作组。

登录 DLC 控制台，选择权限管理，单击用户 > 添加用户，添加新用户。

2. 填写基本信息：填写用户 ID、用户名称和描述，选择用户类型。

说明：

选择用户类型为“普通用户”时，可通过个人授权或获取指定工作组的所有权限，选择用户类型为“管理员”时不需要绑定工作组，即可获取所有权限。

3. 绑定工作组：选择工作组进行绑定（非必选）。

用户授权

在用户列表为用户单独进行授权，授权包含“数据权限”和“引擎权限”，权限策略与工作组涉及权限策略一致。更多详细操作参见 子账号权限管理。

添加工作组

1. 数据湖计算 DLC 左侧导航栏选择权限管理，单击工作组 > 添加工作组来创建用户的工作组，创建工作组时可以选择用户进行绑定或创建一个空工作组。详细操作可参见 用户和用户组。

2.填写基本信息：填写工作组名称和描述。

2. 绑定用户：绑定的用户将获得该工作组下的所有权限。

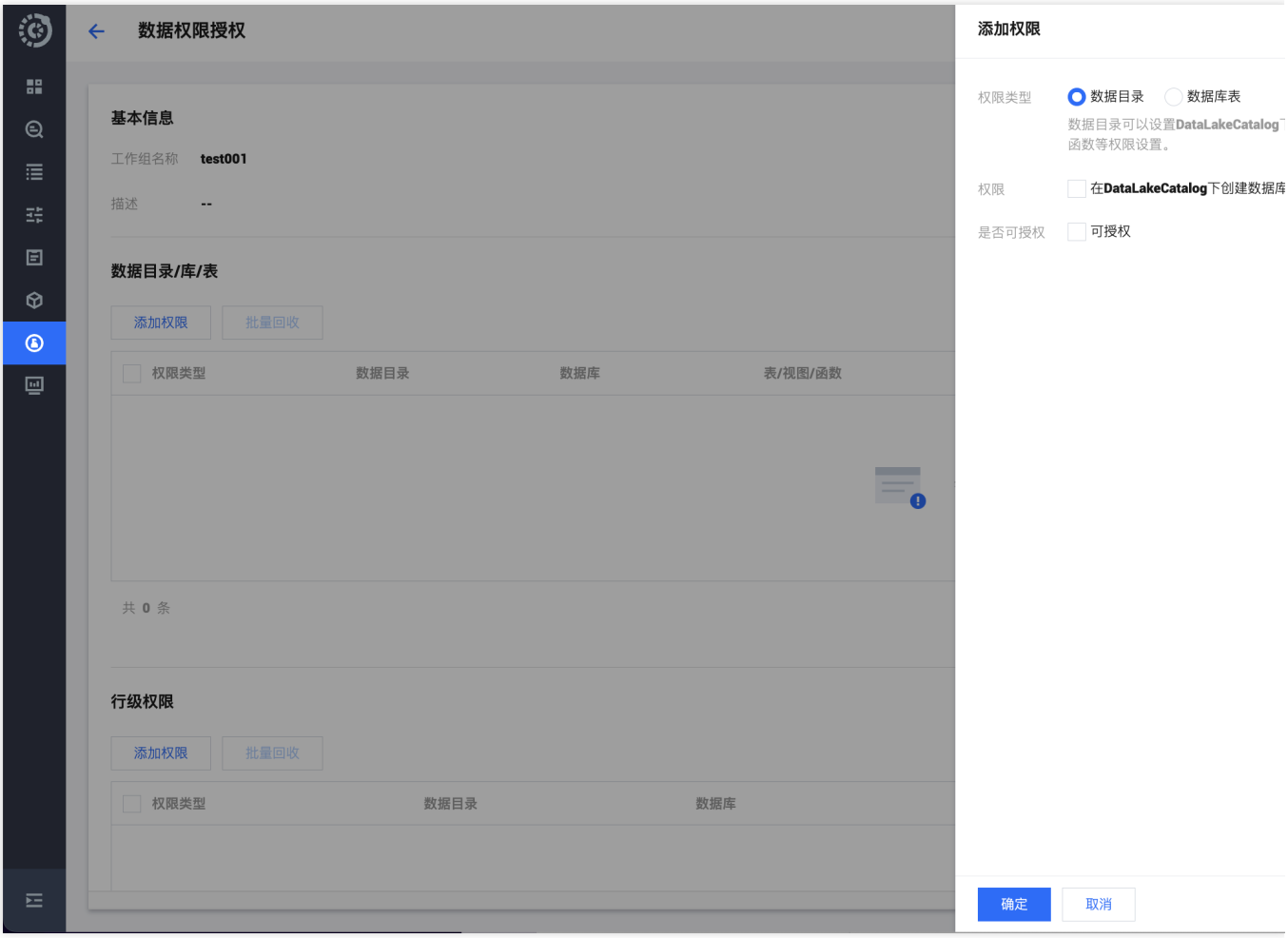
工作组授权

创建工作组后单击列表中的授权操作，为工作组添加权限，包括数据权限和引擎权限。

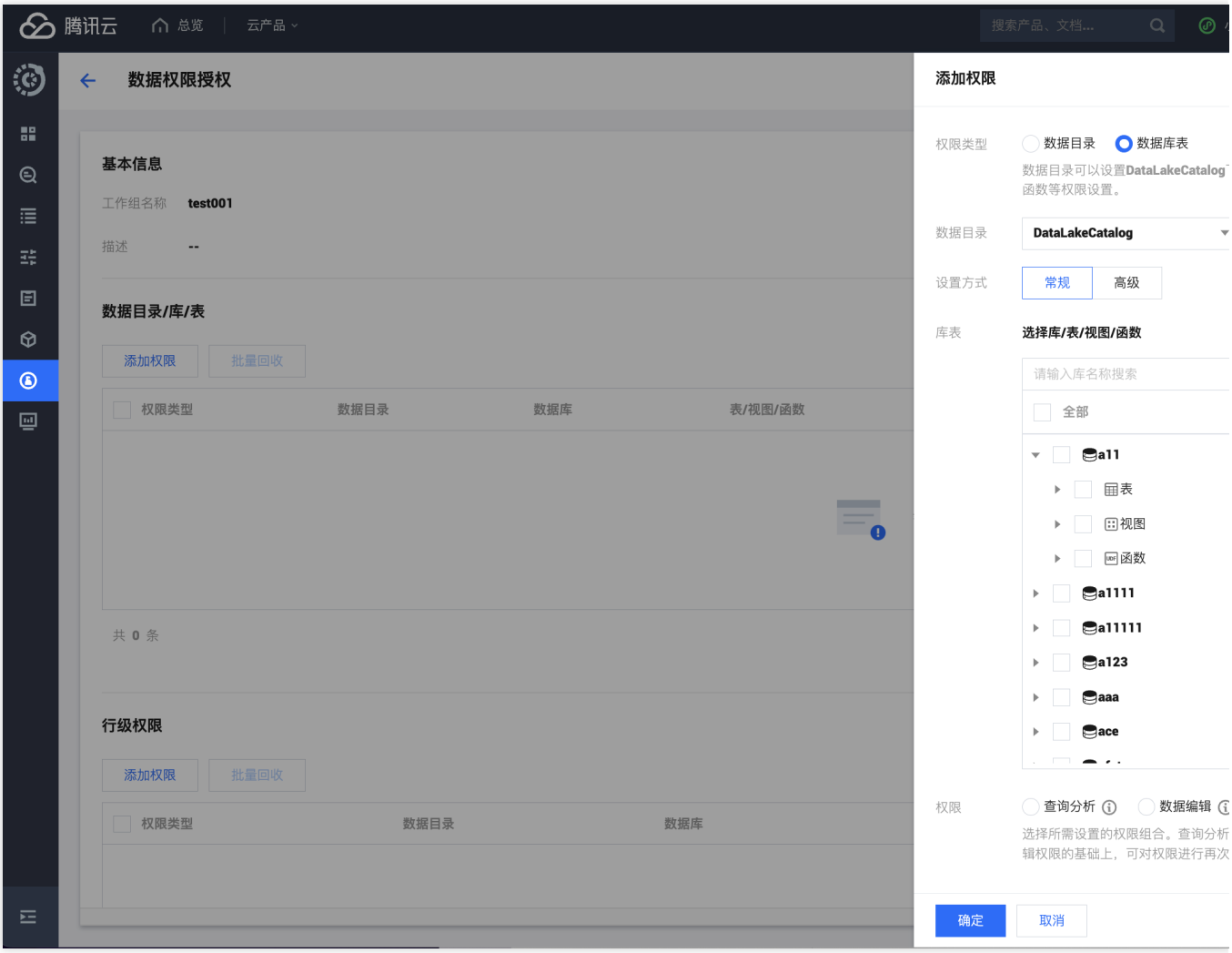
数据权限

数据权限包含：

数据目录权限：包括在数据目录下**创建数据库**和**创建数据目录**两种权限。



数据库表权限：可授予库表级别的细粒度权限，包括对库、表、视图、函数的查询和编辑等权限。



引擎权限

选择数据引擎并授予使用、修改、删除等权限。

引擎操作类权限自动授权

数据湖计算 DLC支持默认开启引擎操作类权限，开启后，所有用户默认拥有该引擎的下列权限：

使用：使用该引擎进行任务执行。

操作：操作引擎的暂停、挂起。

监控：针对引擎的使用情况监控运维。

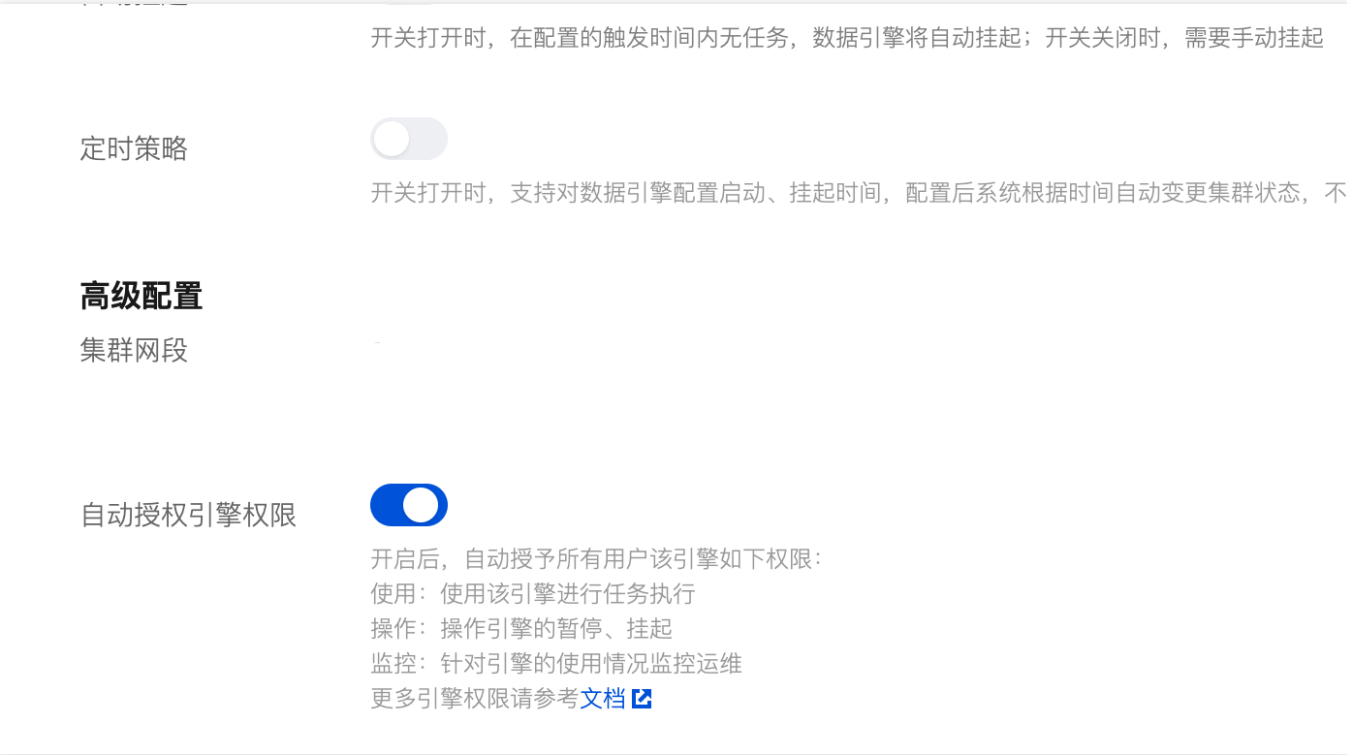
注意：

1. 关闭后，管理员默认继续拥有引擎各权限，普通用户需管理员在权限管理页添加权限。
2. 原有普通用户拥有权限不受影响，可前往权限管理页删除。
3. 后续新建普通用户无使用权限，需在权限管理页手动添加。

如何开启、关闭自动授权引擎权限

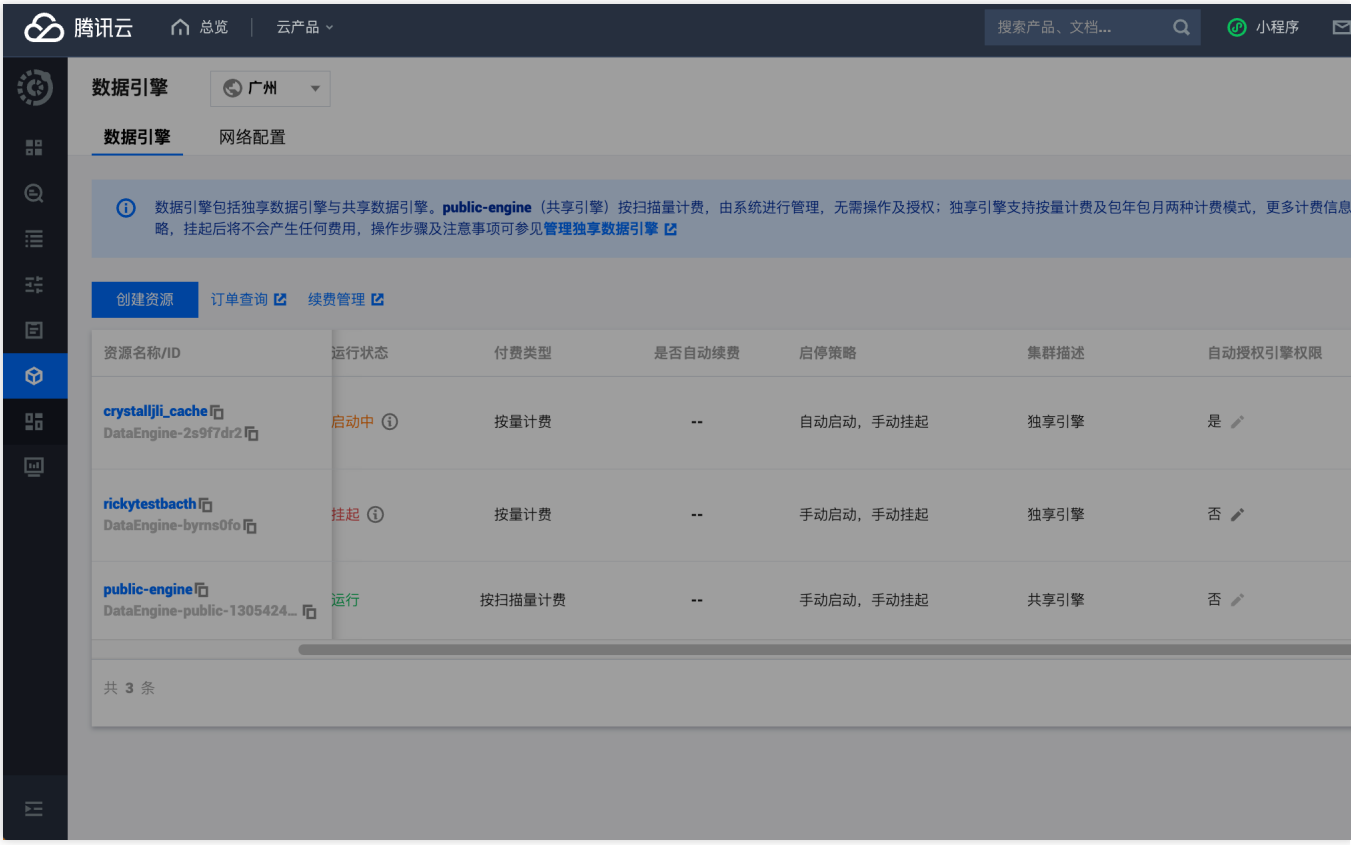
引擎默认开启/关闭操作类权限入口有两个：

入口一：引擎购买页 > 高级配置项



入口二：进入数据引擎 页面，单击编辑授权引擎权限。

引擎权限设置完成后，单击确定。



一分钟入门分区表

最近更新时间：2024-07-17 14:52:09

DLC 分区表

用户可以将数据按照分区目录的方式进行存储，将不同特征的数据存放在不同的目录下，在进行数据探索时，通过 where 条件按照分区进行过滤，DLC 的数据扫描量将大幅减少，提高查询效率。

注意：

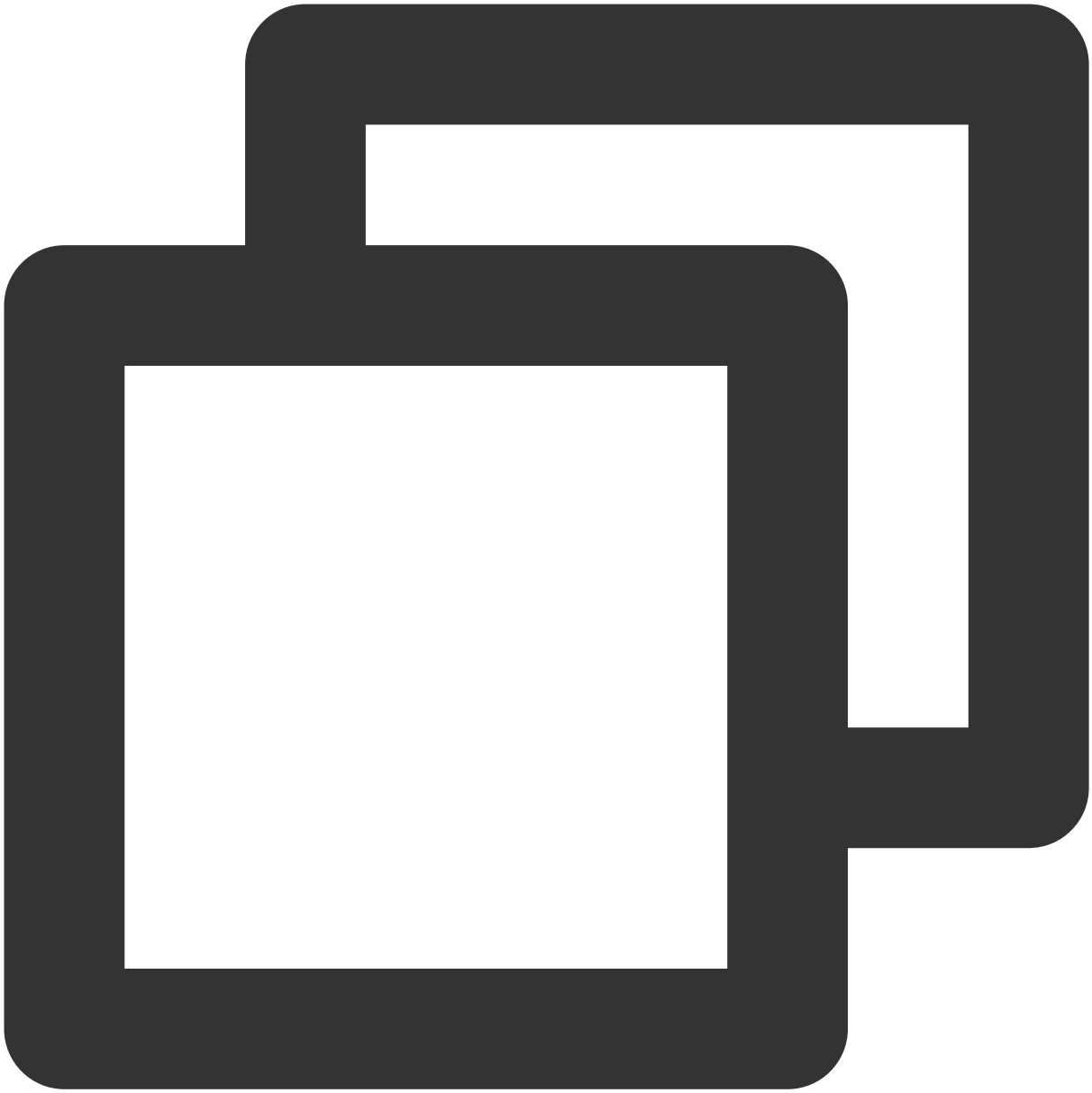
同一个表的分区应使用相同的数据类型及格式。

DLC 内表采用隐藏式分区实现，可不用关注分区目录结构。

创建分区表

创建表格语句中通过 PARTITIONED BY 指定分区字段。

示例：创建分区表 test_part

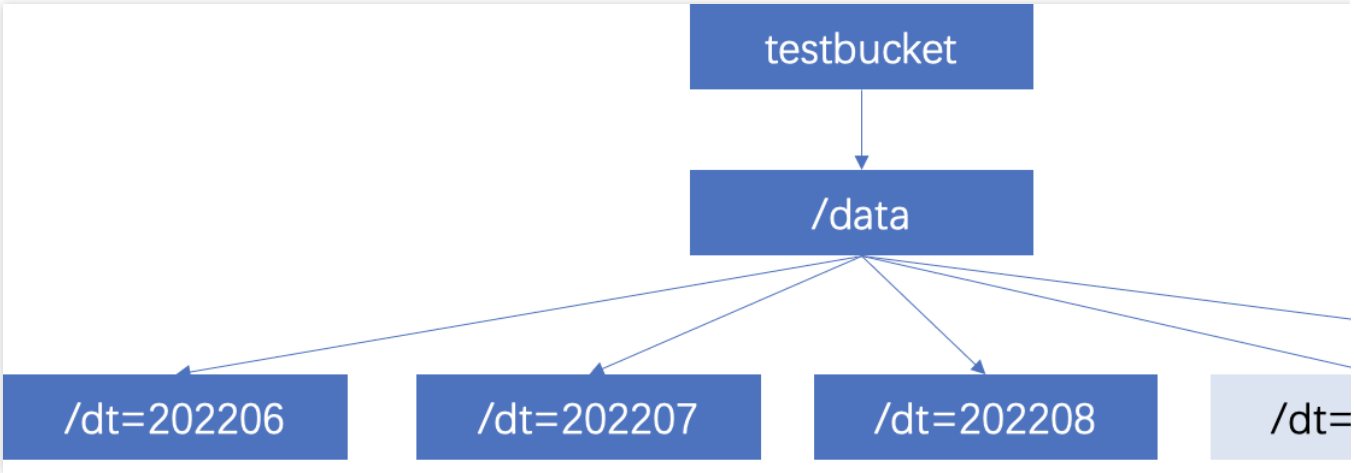


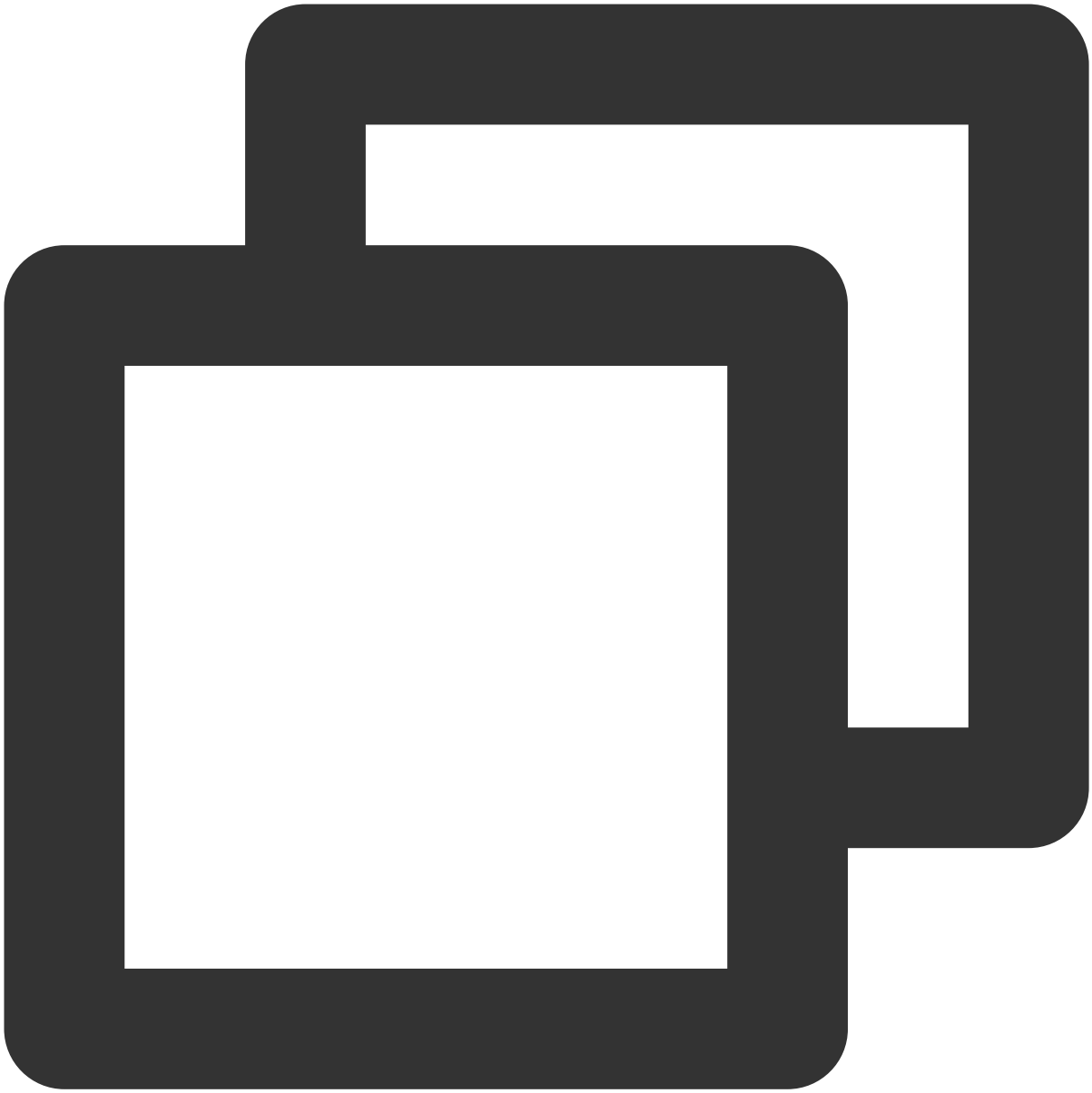
```
CREATE EXTERNAL TABLE IF NOT EXISTS `DataLakeCatalog`.`test_a_db`.`test_part` (
`_c0` int,
`_c1` int,
`_c2` string,
`dt` string
) USING PARQUET PARTITIONED BY (dt) LOCATION 'cosn://testbucket/data/';
```

添加分区

通过 alter table add partition 添加分区

如果用户的数据分区目录为 Hive 的分区规则：分区列名=分区列值，可采用这种方式添加分区，目录组织方式如图。

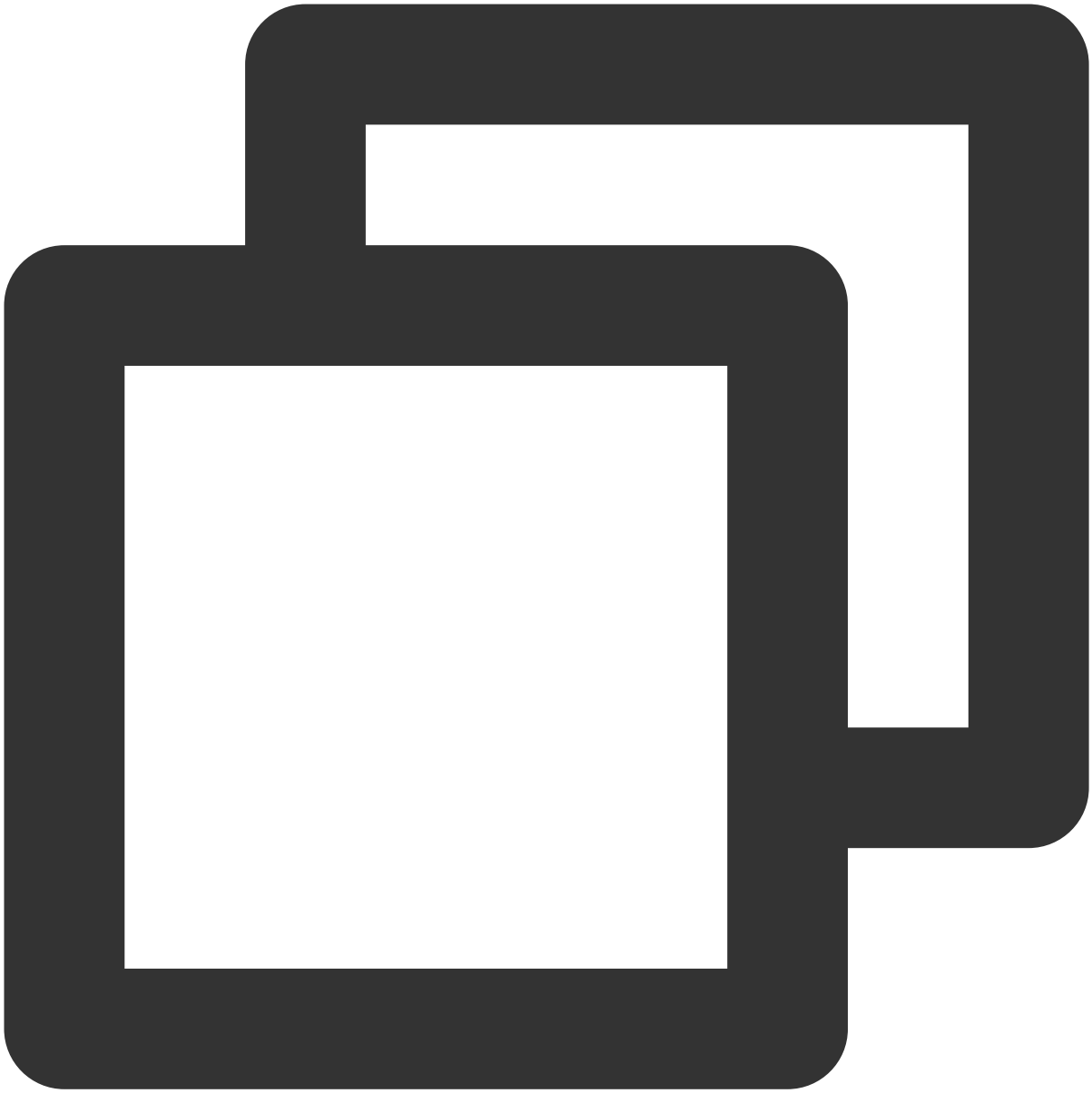




```
ALTER TABLE `DataLakeCatalog`.`test_a_db`.`test_part` add PARTITION (dt = '202206')
ALTER TABLE `DataLakeCatalog`.`test_a_db`.`test_part` add PARTITION (dt = '202207')
ALTER TABLE `DataLakeCatalog`.`test_a_db`.`test_part` add PARTITION (dt = '202208')
ALTER TABLE `DataLakeCatalog`.`test_a_db`.`test_part`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

通过 alter table 指定 location 添加分区

如果用户的数据组织为普通的 cos 目录（非“分区列名=分区列值”组织方式），可以在 add partition 时指定目录。SQL 参考：

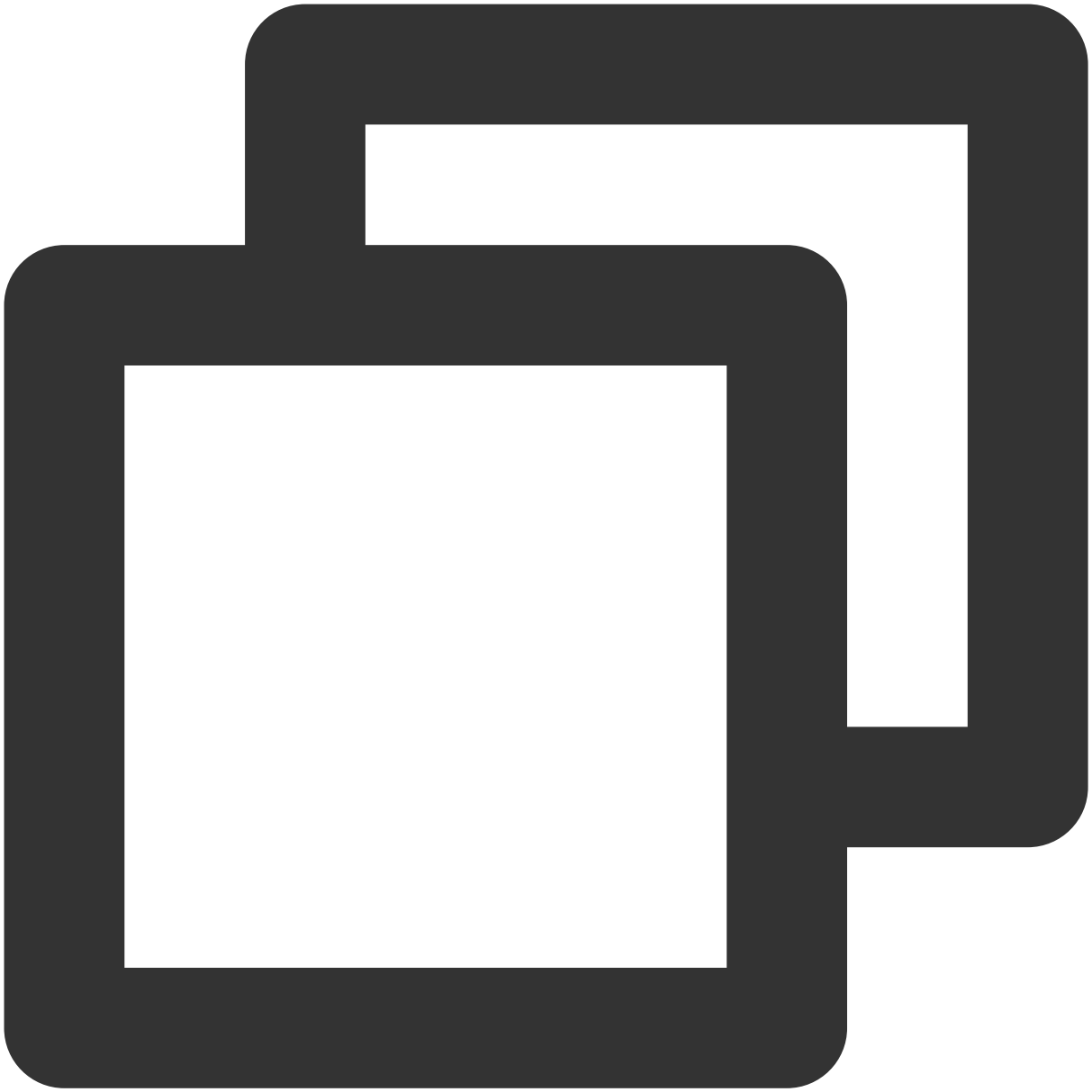


```
ALTER TABLE `DataLakeCatalog`.`test_a_db`.`test_part` add PARTITION (dt = '202211')
ALTER TABLE `DataLakeCatalog`.`test_a_db`.`test_part` add PARTITION (dt = '202212')
```

使用 MSCK REPAIR 自动添加分区

使用 MSCK REPAIR TABLE 语句，扫描在建表时指定的数据目录。若存在新的分区目录，则系统会自动将这些分区添加到数据表的元数据信息中。

SQL参考：



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

建议优先选择 alter table 的方式添加分区，如果采用 msck repair 自动添加分区，有如下约束条件：

MSCK REPAIR TABLE 仅向数据表元数据添加分区，不会删除分区。

数据数据量较大时，不推荐使用 MSCK REPAIR TABLE 的方式，该方式会扫描全部数据量，可能会导致超时。

如果分区目录不为 Hive 的分区规则：分区列名=分区列值，不能采用 MSCK REPAIR TABLE 方式。

一分钟入门 UDF 函数

最近更新时间：2022-08-16 09:41:59

UDF 说明

用户可通过编写 UDF 函数，打包为 JAR 文件后，在数据湖计算定义为函数在查询分析中使用。目前数据湖计算 DLC 的 UDF 为 HIVE 格式，继承 org.apache.hadoop.hive.ql.exec.UDF，实现 evaluate 方法。
示例：简单数组 UDF 函数

```
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;
}
}
```

pom 文件参考

```
<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>
```

创建函数

若您了解 SQL 语法，可通过**数据探索**执行 CREATE FUNCTION语法完成函数创建，或通过可视化界面创建，流程如下：

> 说明：
> 数据湖计算 DLC 的数据管理页目前处于邀测阶段，如需免费体验可 提交工单 进行申请。

1. 登录 数据湖计算控制台 ，选择服务地域。
2. 通过左侧导航菜单进入**数据管理**，选择需要创建的函数的数据库。
3. 单击**函数**进入函数管理页面。
4. 单击**创建函数**进行创建。

UDF 的程序包支持本地上传或选择 COS 路径（需具备 COS 相关权限），示例为选择 cos 路径创建。
函数类名包含“包信息”及“函数的执行类名”。

函数使用

1. 登录 数据湖计算控制台，选择服务地域。
2. 通过左侧导航菜单进入数据探索，选择计算引擎后即可使用 SQL 调用函数。

开启数据优化

最近更新时间：2024-07-31 17:25:23

大数据场景下，频繁地碎片写入会产生大量的小文件，大量的小文件会严重拖慢性能。数据湖计算 DLC 根据大量的生产实践经验，为您提供高效、简单、灵活的数据优化能力，可应对大数据量下的近实时场景。

说明：

1. Upsert 场景下会产生大量小文件和快照，您需要在写入之前就配置好数据优化，避免因写入后再开启需要大量资源处理历史积压的小文件。
2. 目前数据优化能力仅支持 DLC 原生表。
3. 初次执行数据优化任务可能较慢，这取决于存量数据量大小和所选的引擎资源规格。
4. 建议数据优化引擎和业务引擎分开，避免数据优化作业和业务作业相关占用资源导致业务作业受阻。

通过 DLC 控制台配置数据优化

DLC 数据优化策略也可以配置在数据目录、数据库和数据表，当不单独针对数据库、数据表设置数据优化策略时，将继承上一级数据优化策略。用户在配置数据优化时，需要选择一个引擎，用户执行数据优化任务，如用户当前无数据引擎，可参考 购买独享数据引擎 进行购买。DLC 数据治理支持 Spark SupperSQL 引擎和 Spark 作业引擎。

说明：

1. 如用户选择 Spark 作业引擎作为数据优化资源，DLC 会在该引擎上创建数据优化作业，根据集群的规模不同，创建的优化作业数据也有所不同，如集群规模小于32CU时，将创建一个数据优化作业用于执行所有的优化任务，当集群规模大于32CU时，将创建两个数据优化作业，分作业执行写入优化和数据删除优化。
2. 选择 Spark 作业作为数据优化资源时，需要预留一些资源，当优化任务排队超过50个时，DLC 会拉起临时的数据优化作业，快速消费堆积的优化任务。

数据目录配置步骤

您可以通过 DLC 的数据目录编辑功能，针对数据目录配置数据优化能力。

1. 进入 DLC 控制台 数据管理模块，进入**数据目录**页面，单击**数据优化**。

2. 打开数据目录的**数据优化**页面，配置对应的数据优化资源和策略，确认后数据优化功能会自动应用于该数据目录上。

说明：

仅支持针对 DataLakeCatalog 数据目录进行数据优化配置。

数据库配置步骤

如果您要针对某个数据库单独配置数据优化策略，则可以通过 DLC 的数据库编辑能力，针对数据库配置数据优化能力。

1. 进入 DLC 控制台 数据管理模块，进入**数据库**页面，进入 DataLakeCatalog 下的数据库列表。

2. 打开数据库页面，点击**数据优化配置**，确认后数据优化策略会自动作用于该数据库上。

说明：

创建数据库、编辑数据时，默认显示数据优化策略继承上一级的数据目录的数据优化策略，如果要自定义数据优化策略时，需要选择**自定义配置**并配置数据优化资源和策略。

数据表配置步骤

如果您要针对某个数据表单独配置数据优化策略，则可以通过 DLC 的数据表编辑能力，针对数据表配置数据优化能力。

1. 进入 DLC 控制台 数据管理模块，进入**数据库**页面，选择数据库后进入**数据表**列表页面，单击**创建原生表**。

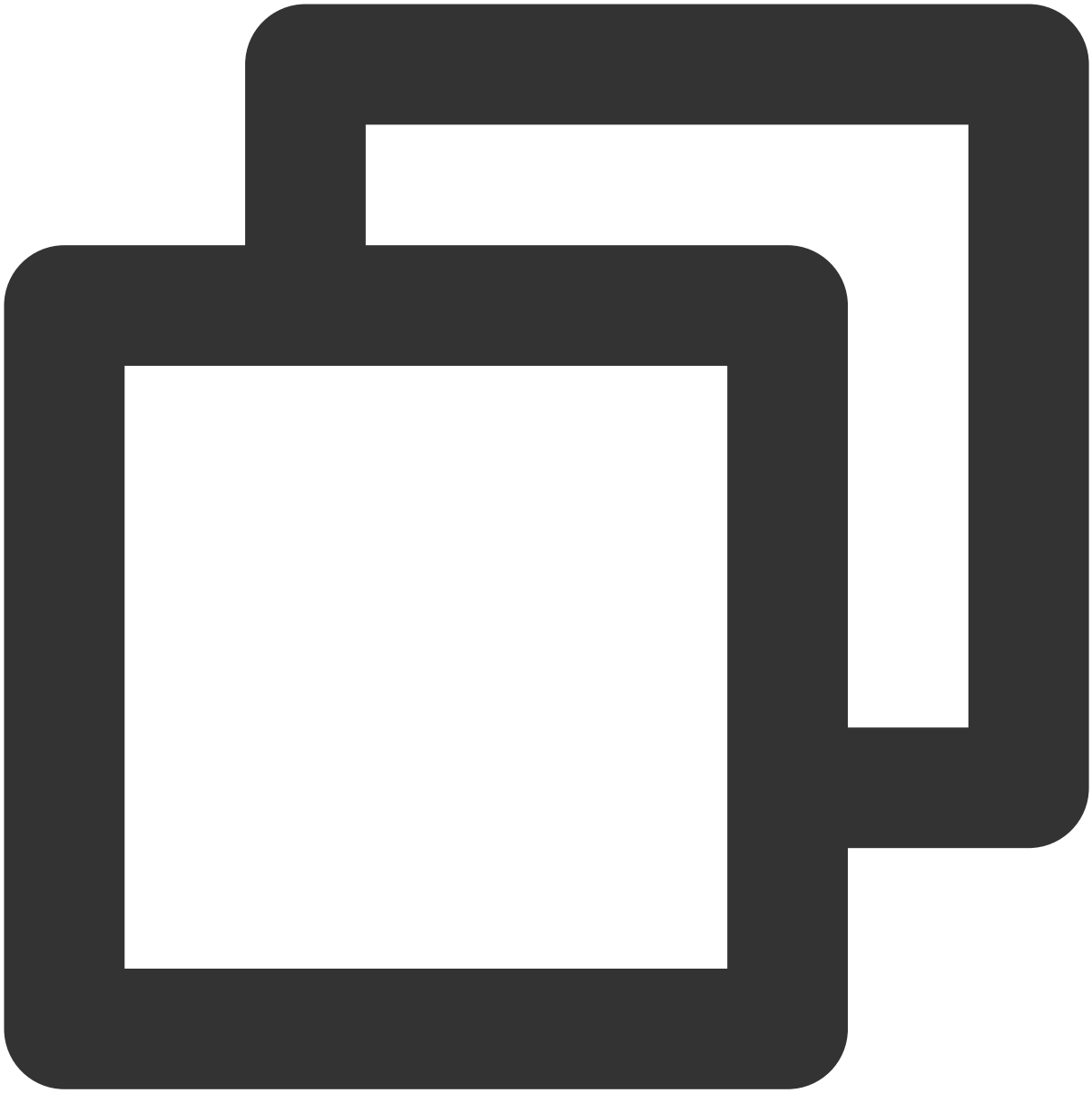
2. 打开创建原生表页面，配置对应的优化资源，确认后数据优化策略会自动作用于该数据表上。
3. 针对已经创建好的表，可以单击**数据优化配置**，编辑已存在的数据表的数据优化策略。

说明：

创建数据表、编辑数据表时，默认显示数据优化策略继承上一级的数据表的数据优化策略，如果要自定义数据优化策略时，需要选择**自定义配置**并配置数据优化资源和策略。

通过属性字段配置数据优化

除了以上可视化方式配置数据优化，您还可以手动指定库和表字段属性来进行配置。例如：



```
// 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'
)
```

其中，数据优化的属性值都可以通过ALTER语句进行修改，属性值定义如下：

属性值	含义	默认值	取值说明
smart-optimizer.inherit	是否继承上一级策略	default	none：不继承 default：继承
smart-optimizer.written.enable	是否开启写入优化	disable	disable：不开启 enable：开启
smart-optimizer.written.advance.compact-enable	（可选）写入优化高级参数，是否开始小文件合并	enable	disable：不开启 enable：开启
smart-optimizer.written.advance.delete-enable	（可选）写入优化高级参数，是否开始数据清理	enable	disable：不开启 enable：开启
smart-optimizer.written.advance.min-input-files	（可选）合并最小输入文件数量	5	当某个表或分区下的文件数目超过最小文件个数时，平台会自动检查并启动文件优化合并。文件优化合并能有效提高分析查询性能。最小文件个数取值较大时，资源负载越高，最小文件个数取值较小时，执行更灵活，任务会更频繁。建议取值为5。
smart-optimizer.written.advance.target-file-size-bytes	（可选）合并目标大小	134217728 (128 MB)	文件优化合并时，会尽可能将文件合并成目标大小，建议取值128M。
smart-optimizer.written.advance.retain-last	（可选）快照过期时间，单位天	5	快照存在时间超过该值时，平台会将该快照标记为过期的快照。快照过期时间取值越长，快照清理的速度越慢，占用存储空间越多。
smart-optimizer.written.advance.before-	（可选）保留过期快照	2	超过保留个数的过期快照将会被清理。保留的过期快照个数越多，存

days	数量		储空间占用越多。建议取值为5。
smart-optimizer.written.advance.expired-snapshots-interval-min	（可选）快照过期执行周期	600（10 hour）	平台会周期性扫描快照并过期快照。执行周期越短，快照的过期会更灵敏，但是可能消耗更多资源。
smart-optimizer.written.advance.cow-compact-enable	（可选）是否开启COW表（V1表或者V2非Upsert表）合并	disable	该配置项开启后，系统会自动针对 COW 表产生文件合并任务。 需注意：COW 表通常数据量都较大，进行文件合并时可能会占用较多的资源，您可根据资源情况及表大小选择是否开启 COW 表文件合并。
smart-optimizer.written.advance.strategy	（可选）文件合并策略	binpack	binpack（默认合并策略）：将文件以 append 的方式对满足合并条件的数据文件合并为更大的数据文件。 sort：sort 策略合并时会根据指定的字段进行排序，您可以根据业务场景选择使用频率较高的查询条件字段作为排序字段，合并后可提高查询性能。
smart-optimizer.written.advance.sort-order	（可选）当文件合并策略为sort时，配置的sort排序规则	-	如果您没有配置排序策略，Upsert 表将选择设置的 upsert 键值（默认取前两个键值）按照 ASC NULLS LAST 的方式排序。如果 COW 表 sort 合并时找不到排序策略，将采用 binpack 默认合并策略。
smart-optimizer.written.advance.remove-orphan-interval-min	（可选）移除孤立文件执行周期	1440（24 hour）	平台会周期性扫描并清理孤立文件。执行周期越短，清理孤立文件会更灵敏，但是可能消耗更多资源。

优化建议

DLC 后端定期统计原生表的各个指标项，并根据这些指标项结合最佳实践，给出原生表的优化建议。优化建议共分为4个优化建议项，包含表使用场景基本配置、数据优化建议及数据存储分布项建议。

优化建议检查项	子检查项	含义	业务场景	优化建议
表基本属性配置	是否开启元数据治	检查是否开启元数据治理，防	append/merger	建议开

检查	理	止表频繁写入引起元数据metadata数量膨胀	into/upsert	启
	是否设置 bloom filter	检查是否设置 bloom filter，在开启 bloom filter 表后，针对 MOR 表并快速过滤 deletes 文件，能加快 MOR 表查询和 deletes 文件合并	upsert	必须开启
	是否配置 metrics 关键属性	检查是否 metrics 在设置为 full，该属性开启后，会记录全部的 metrtics 信息，避免应表 location 过长导致 metrics 信息记录不全	append/merger into/upsert	必须开启
数据优化配置检查	小文件合并	检查是否开启小文件合并	merge into/upsert	必须开启
	快照过期	检查是否开启快照过期	append/merge into/upsert	建议开启
	移除孤立文件	检查是否开启移除孤立文件	append/merge into/upsert	建议开启
近期治理任务检查项	近期治理任务检查项	如表开启了数据治理，系统统计数据治理任务的执行情况，当出现连续多个任务执行超时或者失败时，将判断为需要优化	append/merger into/upsert	建议开启
数据存储分布	平均文件大小	采集快照的summary信息，计算平均文件大小，当平均文件大小小于10M时将判断为需要优化	append/merger into/upsert	建议开启
	MetaData 元文件大小	采集表 metadata.json 元文件大小，当该文件大小超过10M将判定为需要优化	append/merger into/upsert	建议开启
	表快照数量	采集表快照数量，当表快照数量超过1000个将判定为需要优化	append/merger into/upsert	建议开启

表属性基本配置项优化建议

检查并配置元数据治理方法

Step1 检查方法

通过 show TBLPROPERTIES 查看表属性，检查是否配置 “write.metadata.delete-after-commit.enabled”，“write.metadata.previous-versions-max”。

Step2 配置方法

如果 Step1检查结果为未配置，可通过如下 Alter table DDL 进行配置，参考方法如下。

```
ALTER TABLE
  `DataLakeCatalog`.`axitest`.`upsert_case`
SET
  TBLPROPERTIES(
```

```
    'write.metadata.delete-after-commit.enabled' = 'true',
    'write.metadata.previous-versions-max' = '100'
  );
```

说明：

开启元数据自动治理，需要将属性 “write.metadata.delete-after-commit.enabled” 设置为 true。可根据实际情况设置保留的历史 metadata 的个数，如 “write.metadata.previous-versions-max” 设置为100，则最多保留历史100个 metadata。

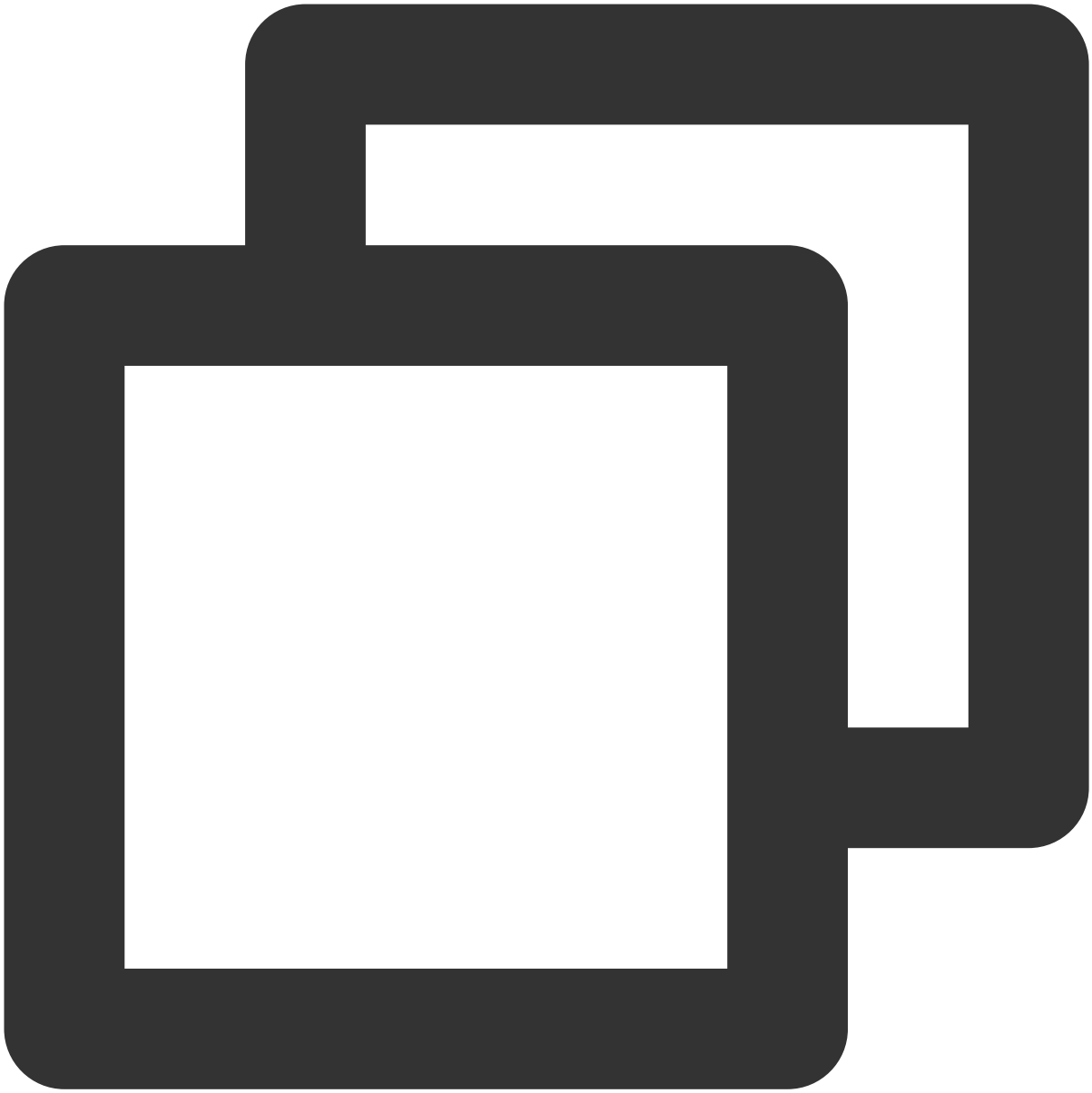
检查并设置 bloom filter 方法

Step1 检查方法

通过 show TBLPROPERTIES 查看表属性，检查是否设置了 “write.parquet.bloom-filter-enabled.column.{column}” 为 true。

Step2 设置方法

如果 Step1检查结果为未配置，可通过如下 Alter table DDL 进行配置，参考方法如下。



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

说明：

建议在 upsert 场景开启 bloom，且根据 upsert 主键进行设置，如有多个主键，建议针对前两个主键字段设置。

在更新 bloom 字段后，如上游有 inlong/oceans/flink 写入时，要重启上游导入作业。

检查并配置表关键属性 metrics

Step1 检查方法

通过 show TBLPROPERTIES 查看表属性，检查 “write.metadata.metrics.default” 是否配置为 “full”。

Step2 配置方法

如果 Step1检查结果为未配置，可通过如下 Alter table DDL 进行配置，参考方法如下。

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

数据优化配置优化建议

Step1 检查方法

通过 SQL 检查

通过 show TBLPROPERTIES 查看表属性，检查是否配置了数据优化，数据优化的属性配置值参考DLC 原生表核心能力。

通过 DLC 控制台可视化检查

进入 DLC 控制台 数据管理模块，进入**数据库**页面，选择数据库后进入**数据表**列表页面，选择要检查的表，进入**数据优化配置**。

Step2 配置方法

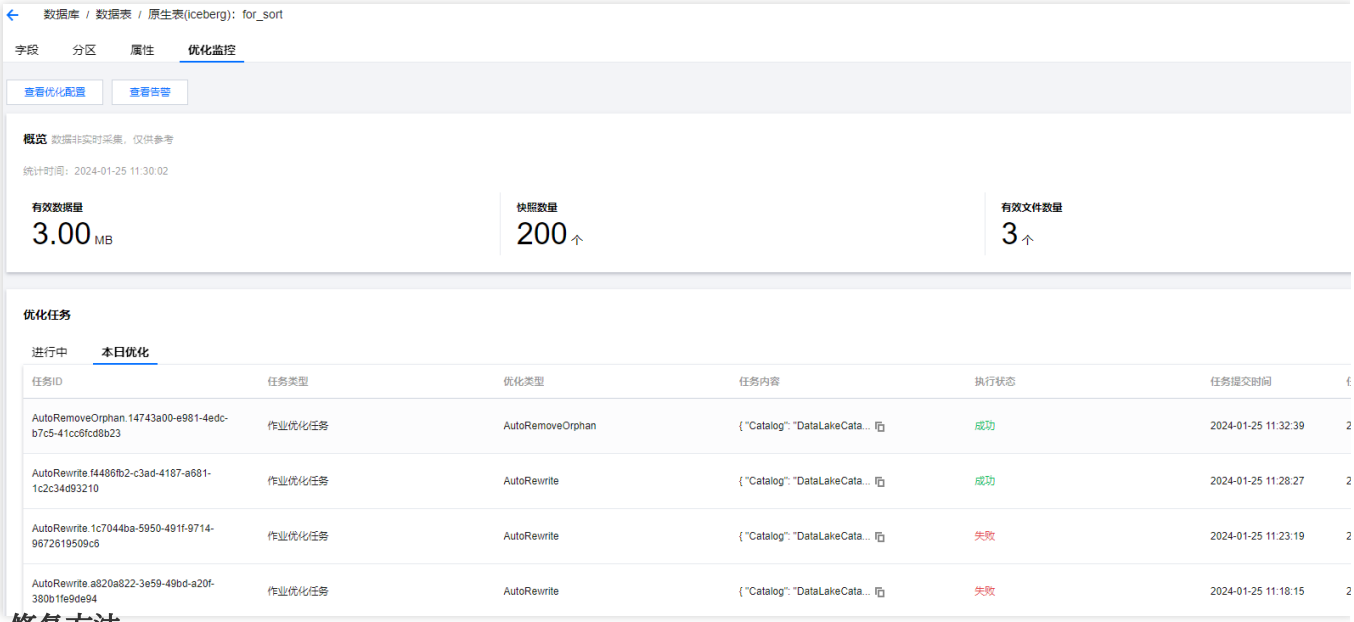
根据指导开启数据优化。

近期数据治理优化任务项优化建议

检查数据治理是否正常

Step1 检查方法

进入 DLC 控制台 数据管理模块，进入**数据库**页面，选择数据库后进入**数据表**列表页面，点击数据表名称，进入**优化监控**，选择**优化任务**中的**本日优化**，检查最近三小时有运行失败的任务，如存在则是本项检查不通过。选择失败的任务，**查看详情**中的**运行结果**。



Step2 修复方法

场景数据优化任务运行失败原因及处理方法汇总。

1. 数据治理配置错误导致运行失败。

开启了 sort 合并策略，但是配置排序规则配置错误，或者配置了不存在的排序规则。

配置为进行数据治理的引擎发生了变化，导致治理任务运行时找不到合适的引擎。

2. 任务运行超时。

说明：

近期数据优化任务运行情况检查在修复后，需要等待三个小时后观察是否恢复。

数据存储分布项优化建议

说明：

该场景检查不通过通常是数据量较大，建议先手动处理后在考虑添加到数据优化治理。

建议用效率更高效的 spark 作业引擎处理。

进行手动小文件合并处理时，target-file-size-bytes 参数根据业务场景配置，upsert 写入建议不超过134217728，即128M，append/merge into 写入建议不超过536870912，即512M。

采用 spark 作业引擎执行快照过期时，可调大 max_concurrent_deletes 参数。

数据文件平均大小检查不通过处理方法

Step1 原因汇总

数据文件平均大小过小，通常出现在如下几个情况：

表分区划分过细，导致每个分区只有少量数据。

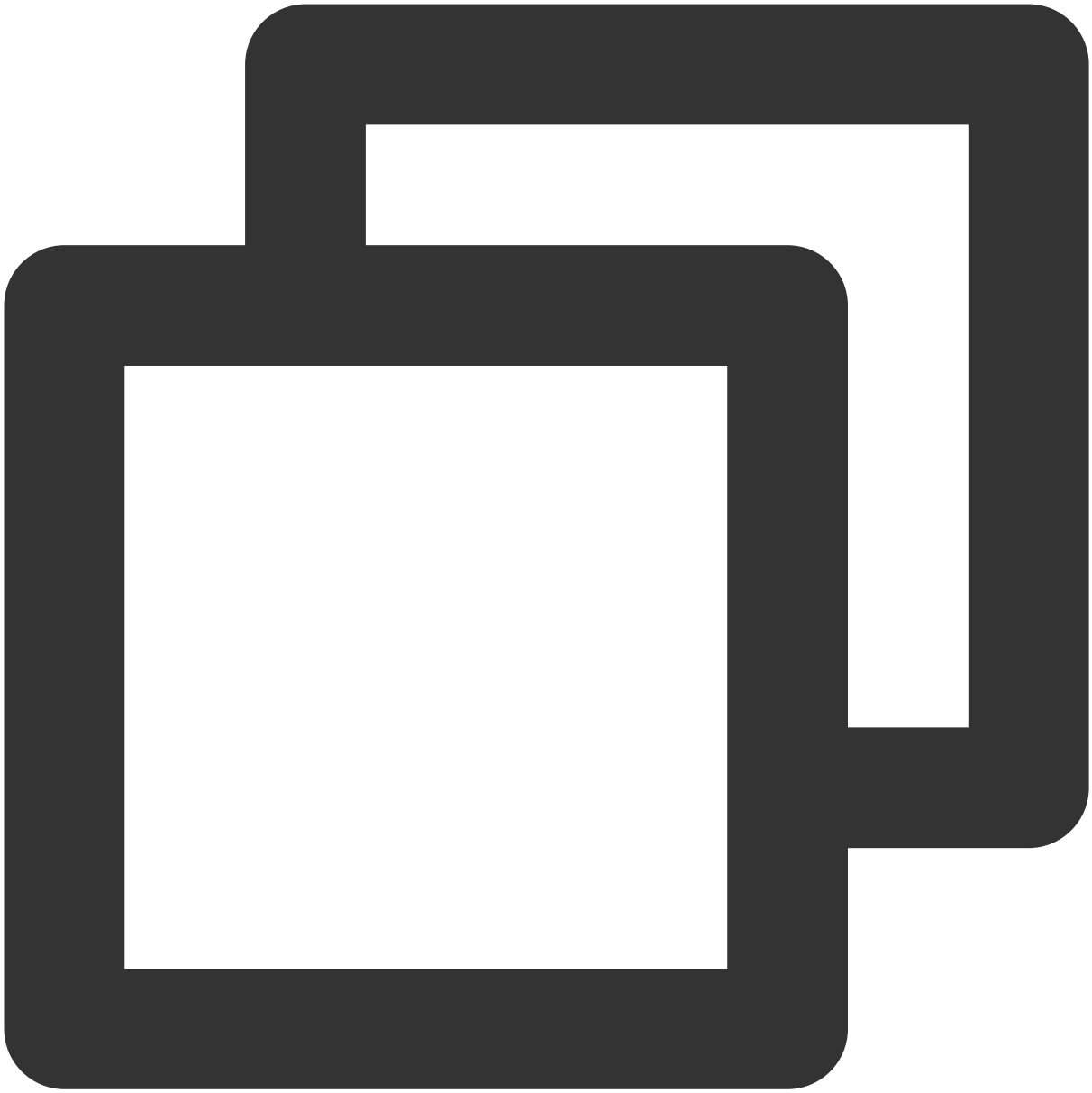
表进行 Insert into/overwrite 方式写入时，上游的数据分散，如上游数据也是分区表，且分区内数据量少。

表进行 merge into 写入 MOR 表，但是没有进行小文件合并。

表进行 upsert 写入，但是没有进行小文件合并。

Step2 修复方法

参考如下 SQL，手动执行小文件合并。



```
CALL `DataLakeCatalog`.`system`.`rewrite_data_files`(
  `table` => 'test_db.test_tb',
  `options` => map(
    'delete-file-threshold',
    '10',
    'max-concurrent-file-group-rewrites',  --根据实际资源情况而定，并发度越高，采用的资源越
    '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 元文件大小检查不通过处理方法

Step1 原因汇总

MetaData 元文件过大，通常是数据文件数量过多引起的，大部分是如下几个原因导致：

表长期进行 append 写入，且每次写入的文件分散量大。

表属性为 MOR 表，长期进行 merge into 写入，但没有开启小文件合并。

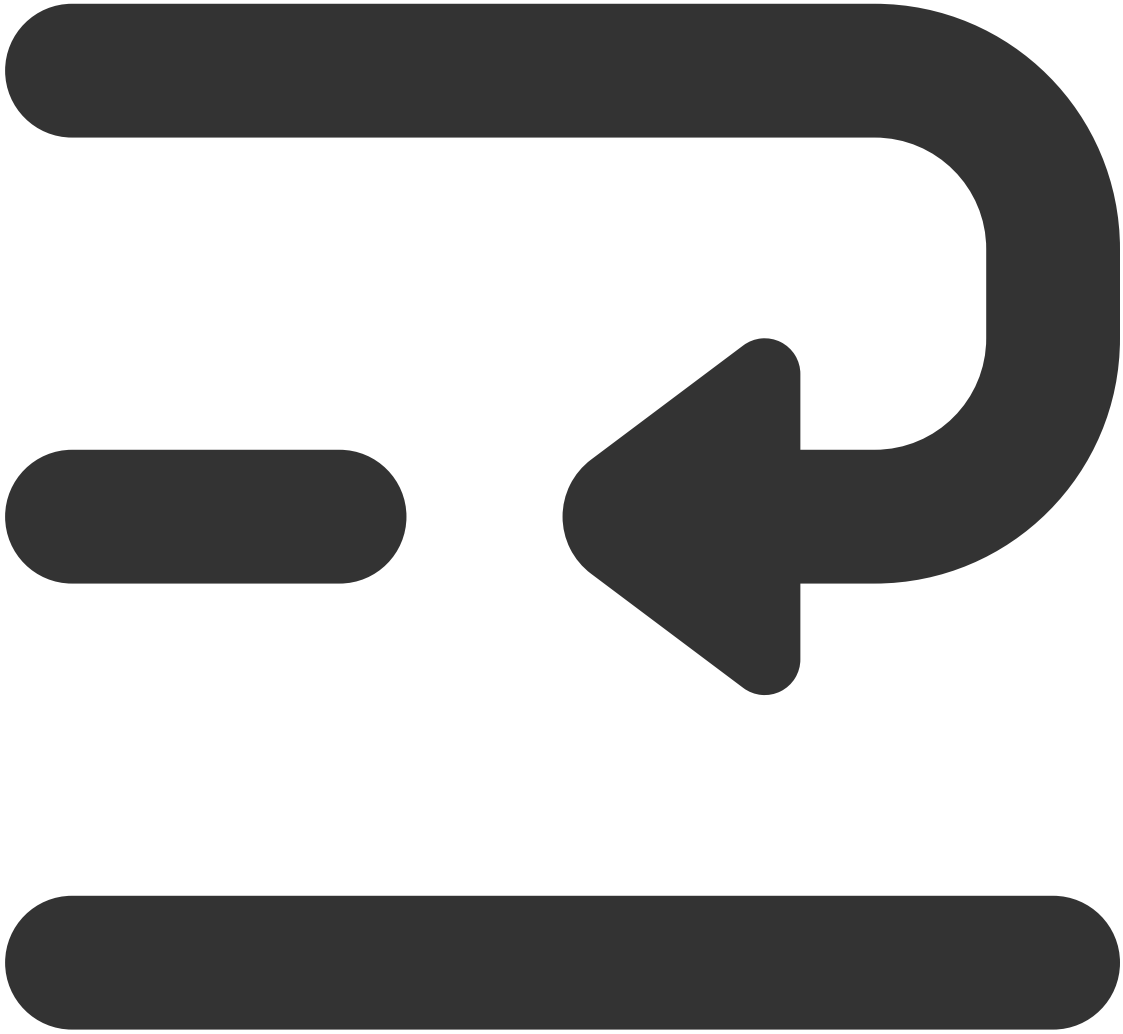
表长期没有进行快照过期，表维护大量的历史快照数据文件。

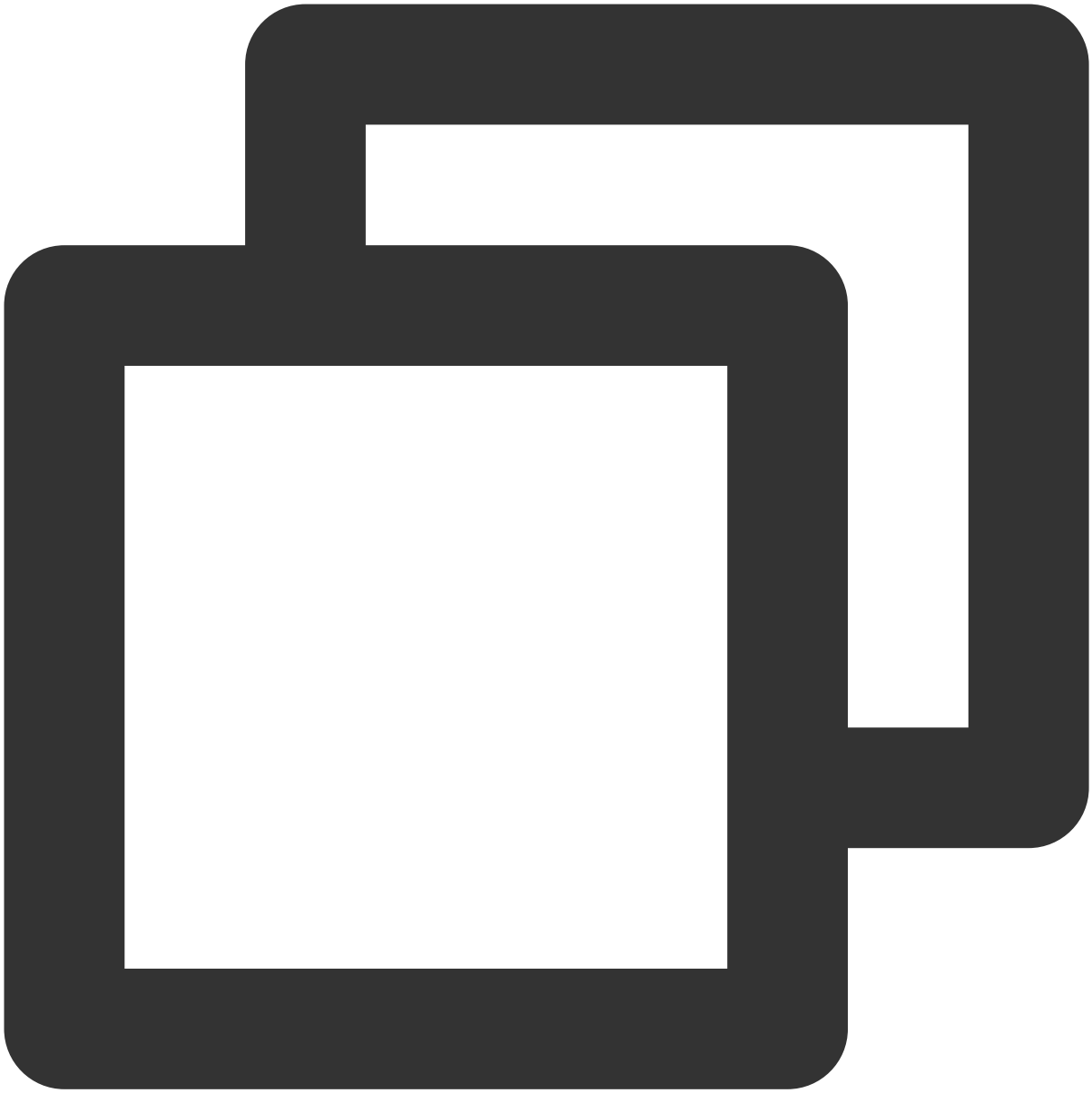
表分区较大，且每个分区的文件都较小数量多。

Step2 修复方法

参考手动执行小文件合并。

参考如下 SQL，手动执行过期快照 SQL，将历史快照清理。





```
--采用spark 作业引擎过期快照参考
CALL `DataLakeCatalog`.`system`.`expire_snapshots`(
  `table` => 'test_db.test_tb',
  `retain_last` => 5,
  `stream_results` => true,
  `older_than` => TIMESTAMP '2024-01-10 13:02:40.407', --过期时间根据表实际快照为准
  `max_concurrent_deletes` => 50
);

--采用spark sql引擎过期快照参考
CALL `DataLakeCatalog`.`system`.`expire_snapshots`(
```

```
    `table` => 'test_db.test_tb',
    `retain_last` => 5,
    `stream_results` => true,
    `delete_by_executor` => true,
    `older_than` => TIMESTAMP '2024-01-10 13:02:40.407', --过期时间根据表实际快照为准
    `max_concurrent_deletes` => 4
  )
```

结合业务场景，将写入的文件进行一定程度聚合，避免写入的文件较分散。

如数据为 insert into/insert overwrite 写入，可通过如下两种方式自动添加一个 repartition 操作。

1. 下面两个参数都为 true 时生效。仅该情况下，可以使用上文的参数控制 repartition 后自动适配分区数或分区大小。

spark.sql.adaptive.enabled：该参数需为 true，集群创建默认为 true。

spark.sql.adaptive.insert.repartition：该参数需为 true，集群创建默认为 false。

2. 指定下面参数生效。这种情况下 repartition 后的分区数即 spark.sql.adaptive.insert.repartition.forceNum 指定的值。

spark.sql.adaptive.insert.repartition.forceNum：该参数指定了具体需要分区的值，集群创建时默认为空。

快照数量检查检查不通过处理方法

Step1 原因汇总

长期没有进行快照过期。

upsert 写入 checkpoint 周期不合理，产生大量的快照。

Step2 修复方法

参考快照过期 SQL，进行快照过期。

调整 flink 写入 checkpoint 周期，DLC 原生表 upsert 写入 checkpoint 周期建议设置在3 - 5min。

跨源分析 EMR Hive 数据

最近更新时间：2024-07-17 15:16:50

数据湖计算 DLC 支持配置 EMR Hive 的数据源进行跨源联合分析。

说明：

标准引擎-presto暂不支持该功能，可使用 SuperSQL 类型的引擎及标准引擎-spark进行分析。

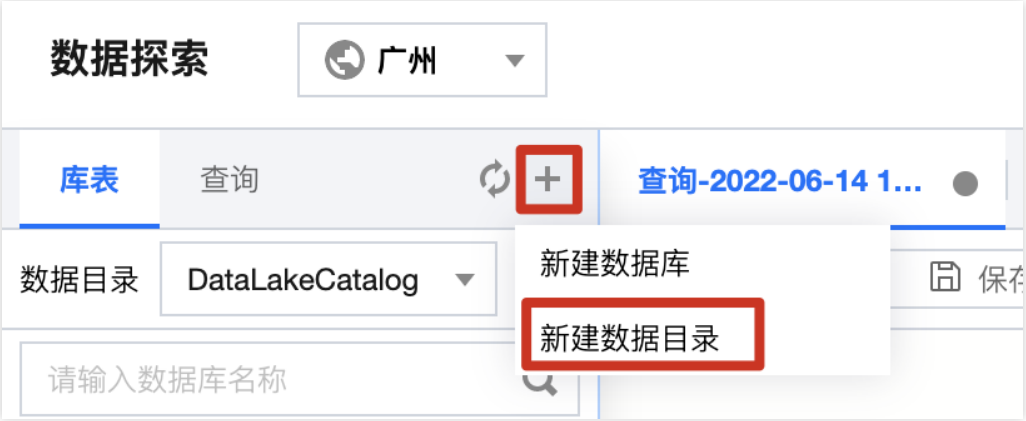
使用前准备

获取 EMR Hive 地址。

使用具备创建数据目录权限的账号，详细权限请参见 DLC 权限概述。

创建 EMR Hive 数据源

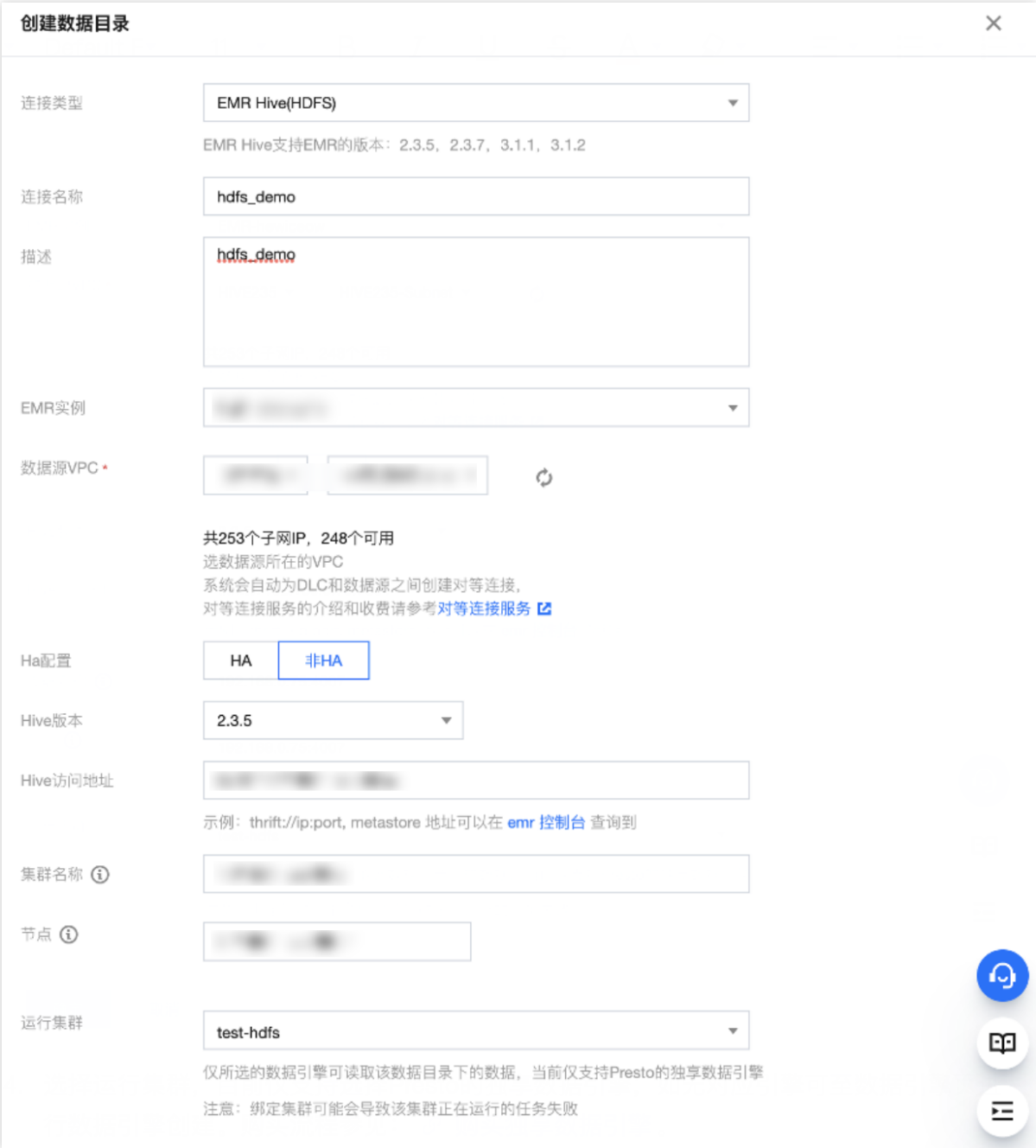
1. 登录 数据湖计算 DLC 控制台，选择服务地域。
2. 通过左侧导航栏进入**数据探索**，单击库表栏的**+按钮，选择新建数据目录**。



3. 选择连接类型为 EMR Hive（HDFS），选择 EMR 的对应实例，VPC 信息将在实例选择后默认填充。**EMR Hive 支持 EMR 的版本：2.3.5，2.3.7，3.1.1，3.1.2**。

注意：

需具备 EMR Hive 实例的相关权限才可进行选择。



4. 选择运行集群，目前仅支持选择Presto的独享数据引擎，如无对应引擎可至数据引擎页进行数据引擎创建。购买流程请参见 购买独享数据引擎。

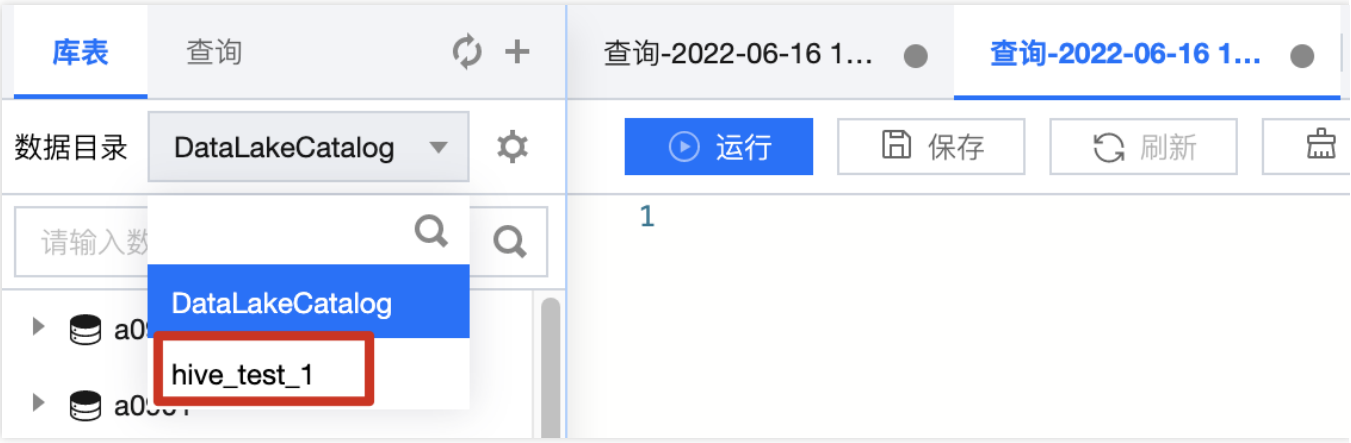
注意：

所选数据引擎网段不可与 EMR 实例网段相同，否则将导致网络冲突，无法进行数据查询分析。

5. 单击**确认**按钮即可完成数据目录创建。

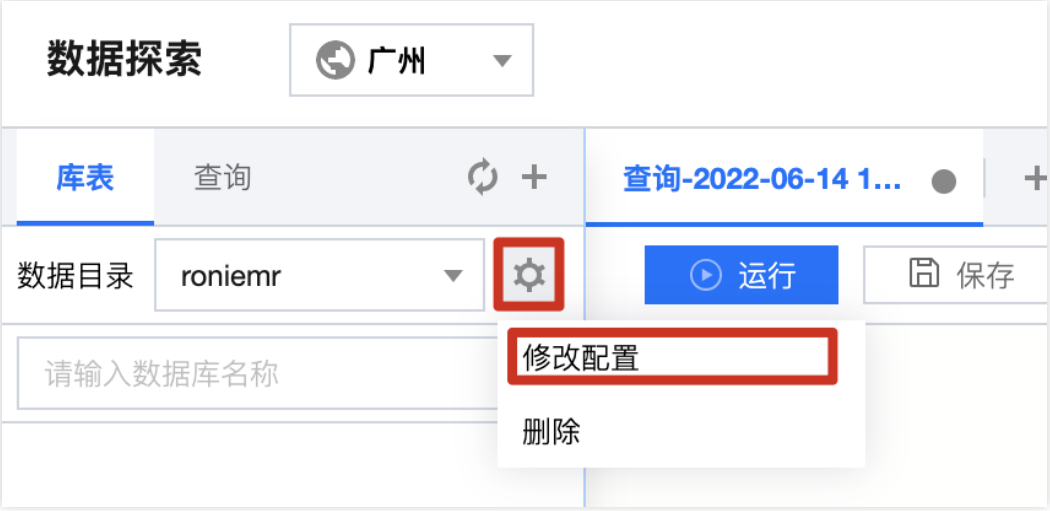
查询 EMR Hive 数据

完成数据目录创建之后，即可在**数据探索页**的数据目录菜单进行数据目录切换。



此时您可通过 SQL 语句对该数据目录进行查询分析。

选择创建数据目录时绑定的数据引擎即可单击**运行**按钮，获得查询结果。

注意：

仅绑定的数据引擎可查询该数据目录，其他数据引擎将无法进行查询。如需变更绑定的引擎，可单击数据目录判的设置按钮就行编辑修改。