

TDSQL-H LibraDB 开发指南





【版权声明】

©2013-2024 腾讯云版权所有

本文档(含所有文字、数据、图片等内容)完整的著作权归腾讯云计算(北京)有限责任公司单独所有,未经腾讯云事先明确书面许可,任何主体不得以任何形式 复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯,腾讯云将依法采取措施追究法律责任。

【商标声明】

🔗 腾讯云

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。未经腾讯云及有关 权利人书面许可,任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为,否则将构成对腾讯云及有关权利人商标权的侵犯,腾讯云将依 法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况,部分产品、服务的内容可能不时有所调整。 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。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务,及相应的技术售后服务,任何问题请联系 4009100100或95716。



文档目录

开发指南 使用 DBeaver 连接 TDSQL-H LibraDB 使用 JDBC 连接 TDSQL-H LibraDB SQL参考 物化视图 修改键列数据类型 常见错误码



开发指南 使用 DBeaver 连接 TDSQL-H LibraDB

最近更新时间: 2024-10-17 20:41:22

TDSQL-H LibraDB 支持您通过数据库管理工具 DBeaver 访问实例,进行数据管理操作。本文为您介绍如何通过 DBeaver 的 ClickHouse (Legacy) 驱 动连接 TDSQL-H LibraDB 实例并管理数据。

背景信息

DBeaver 是一款免费的多平台数据库工具,适用于开发人员、数据库管理员、分析师和所有需要使用数据库的人员。更多 DBeaver 信息,请参见 DBeaver 。

前提条件

- 已创建TDSQL-HLibraDB实例。
- 已获取访问 TDSQL-H LibraDB 实例的账号和密码。
- 获取 DBeaver 连接 TDSQL-H LibraDB 实例时的主机地址。访问 TDSQL-H LibraDB 实例方式不同,获取的主机地址不同,请根据实际选择。
 - ・ 若通过搭建代理使用外网访问 TDSQL-H LibraDB 实例,请获取搭建代理时使用的 CVM 云服务器的公网 IP,并确保外网连接使用的 CVM 云服务器 安全组已添加 TCP:8123、TCP:9000 协议端口。本文中的操作以此方式为例。

 搭建代理的具体操作,请参见 连接实例分析引擎 中的外网连接相关内容。
 - 若使用镜像为 Windows 的 CVM 云服务器,安装 DBeaver 并连接 TDSQL-H LibraDB 实例,请获取 TDSQL-H LibraDB 实例的内网 IP。

 镜像为 Windows 的 CVM 云服务器与 TDSQL-H LibraDB 实例必须为相同的 VPC,并确保 CVM 云服务器安全组已添加 TCP:8123、

 TCP:9000 协议端口。
- 已下载并安装 DBeaver 。更多安装 DBeaver 操作,请参见 Install 。
 推荐使用 DBeaver 版本 DBeaver Community 22.1.2,其他版本可能存在兼容性问题。本文中的 DBeaver 示例版本为 DBeaver Community 22.1.2。

步骤1: 连接 DBeaver 和 TDSQL-H LibraDB 实例

- 1. 启动 DBeaver, 进入 DBeaver 界面。
- 2. 在顶部菜单栏单击 📫 图标后,在选择您的数据库对话框选择 SQL > ClickHouse (Legacy),单击下一步。



DBeaver 22.1.2								-	
件(F) 编辑(E) 导航(N) 搜索	ছ(A) SQL 编辑器	暑 数据库(D) 窗□	l(W) 帮助(H)						
🛊 🔻 🕸 🎋 🕷 🞵 SQL 🔻 🗄	🖻 i 📿 🚟 👻 i 🔍	•						Q	
💆 😨 连接到数据库				— [X				
△ 洗择您的数据库									
创建新的数据库连接。	在下面的列表中	找到您的数据库	驱动程序。						
	输入数据库/驱动	动名称的部分内容	进行过滤 夂	排序方式: 〇 Tit	ile Score				
◎ 全部	4				^				
◎热门				- III •					
📮 SQL	Azure SQL	Babelfish via	Cache	ClickHouse					
NoSQL	Server	IDS (beta)							
□ 分析型		Â							
□嵌入式	ClickHouse	CaskroashDR	CrateDR	CUPPID					
□ Hadoop / 大数据	(Legacy)	COCKIOACIIDB	CrateDB	COBRID					
□ 全文搜索	Me	IBM	IBM						
■ 图数据库	2.5	DB2	DB2						
	Data Virtuality	DB2 iSeries/AS	DB2 z/OS	Denodo 8	~				
测试连接(T)	~上	一步(B) 下一	步(N)>	完成(F)	取消				
L									
							CST zh CN		

3. 在通用 JDBC 连接设置对话框中,单击编辑驱动设置。

-	×	
通用 JDBC 连接设置 ClickHouse (Legacy) 连接设置	IIII	
主要 驱动属性 SSH Proxy 一般		
主机: localhost 数据库/模式:	端口: 8123	
认证 (Database Native) 用户名: 密码:	ŧ.	
① 可以在连接参数中使用变量	连接详情(名称、类型)	
	编辑驱动设置	
驱动名称: ClickHouse (Legacy)		
驱动名称: ClickHouse (Legacy)		

4. 在编辑驱动 'ClickHouse (Legacy)' 对话框的库页签,单击下载/更新,在下载驱动文件对话框中单击下载,待下载完成后,在编辑驱动 'ClickHouse (Legacy)' 对话框单击确定。



) DBeaver 22.1.2 ケ(に) (会語(に) 日前(N) 地奈(A) SOL (会語現 教提佐(D) 窗口(AA) 邦助(4)	- 0 X
	Q [證]
●	
驱动类: ・ 取动类: ・ 重置为默认状态 ・ 取动名称: ClickHouse (Legacy) 編編駆动设置	
週以连按(1) <(∇) トージ(∇)> 元以(F) 取用	
nastions Control #14	
mecuons - General 注反	

- 5. 在通用 JDBC 连接设置对话框的主要页签中,填写以下信息,并单击测试连接。
- 主机:填写前提条件已获取的 CVM 云服务器公网 IP。
- 用户名、密码:填写前提条件已获取的访问 TDSQL-H LibraDB 实例的账号和密码。
- 其他参数:保持默认。

DBeaver 22.1.2	- 0
(F) 编辑(E) 导航(N) 搜索(A) SQL 编辑器 数据库(D) 窗口(W) 帮助(H)	
▼ ♥ ₺ ¾ 🗓 SQL ▼ 📾 🕰 🏝 ▼ Q, ▼	Q I
② 连接到数据库	X
通用 JDBC 连接设置	
ClickHouse (Legacy) 连接设置	III.
主要 驱动属性 SSH Proxy	
一 般	
JDBC URL: jdbc:clickhouse:// 184:8123	
主机: 184 端口: 8123	
数据库/模式:	
认证 (Database Native)	
用户名: root	
密码: ●●●●●●●●●●	
① 可以在连接参数中使用变量 连接详情(名称、类型)	2) 20 m
驱动名称:ClickHouse (LegaCy) 编辑驱动	<u>设直</u>
	_
测试连接(T) <上一步(B) 下一步(N)> 完成(F) 取消	
测试连接(T) <上一步(B) 下一步(N)> 完成(F) 取消	



6. 连接成功后依次单击确定、完成,成功连接 TDSQL-H LibraDB 实例。

(☆) DBeaver 22.1.2 文件(F) 编辑(E) 导航(N) 搜索(A) SQL 编辑器 数据库(D) 窗口(W) 帮 体 ▼ ♥ ₺ ☆ ①SQL ▼ :: 台 :: 〇 프 ▼ :: Q ▼	助(H)	- L ×
Se 数 😰 连接到数据库	- 🗆 X	- E
输入 通用 JDBC 连接设置 ClickHouse (Legacy) 连接设置	IIII	
主要 驱动属性 SSH Proxy 一般 JDBC URL: 主机: 数据库/模式: 认证 (Databas 用户名: root 密码: ●●●●	□ × □: 8123 ○ 0.2.6 〕详细信息(D) >>	
① 可以在连接参数中使用变量	连接详情(名称、类型)	
驱动名称: ClickHouse (Legacy)	编辑驱动设置	
测试连接(T) <上一步(B) 下一步(N)>	完成(F) 取消	
Connections - General 连接		CST zh_CN

步骤2: 使用 DBeaver 查询及分析数据

连接 DBeaver 和 TDSQL-H LibraDB 实例后,您可以在 DBeaver 左侧查看到新建的 TDSQL-H LibraDB 实例连接,并通过 SQL 管理 TDSQL-H LibraDB 实例。

可通过 DBeaver 执行的数据库操作示例如下。更多操作,请参见 DBeaver 帮助。

• 查看所有数据库及表

在创建的 TDSQL-H LibraDB 实例连接下查看所有库表信息。

DBeaver 22.1.2	-		\times
文件(F) 编辑(E) 导航(N) 搜索(A) SQL 编辑器 数据库(D) 窗口((W) 帮助(H)		
🗱 🔻 🕸 🏀 💘 🎵 SQL 🔻 🔂 🕰 🖛 🔍 🗶		Q	: 😭 🧃
急 数据库导航 × 🕴 🗧 🗗			- E
输入表格名称的一部分			
✓ 184 - 184:8123			
> tencentdb			
✓ 目 db」			
> customer local 1.2M			
> Ineorder_local			
> part_local 3.6M			
> supplier_local 77K			
> 2			
> 🗈 default			
> 📑 system			
	CST zh CN		



• 查看表结构

在创建的 TDSQL-H LibraDB 实例连接下,在目标表处单击右键,选择查看表,即可查看表结构信息。



• 查看表数据

在创建的 TDSQL-H LibraDB 实例连接下,在目标表处单击右键,选择查看表后,在数据页签即可查看表数据信息。



😰 DBeaver 22.1.2 - customer_local								- [
文件(F) 编辑(E) 导航(N) 搜索(A) SQL 编辑器 数据库	€(D) 窗口(W) 帮助(H)						
👫 🔻 🏺 🇞 💘 🗊 SQL 🔻 💽 提交 🗋 回滚 📑 👻 🧯	Auto	⊙ - i	.184 🔻	🖪 default 🔻 🙆 🖶 🔻	Q -				९ 🔡 🕼
🧟 数据库导航 🛛 🛛 🙀 🖛 📭 🗖 🖛	* 8 🗆 🗆	== custo	omer_local $ imes$						- E
输入表格名称的一部分	▼	☴ 属性	🖪 数据 🐣 ER 图				184 🗈 db	∃表 ▼ ธcus	tomer loca
✓ 184184:8123	4	Curte	mor local 5.7 mar		用 (店田 Ctrl+Space)		b y d	T. E. 🕲 👻	$\leftarrow \tau \Rightarrow \tau$
> tencentdb				TOL REIGER	R (Igh currspace)		▲ ■ 粉店杏手哭 >		
> 🛾 db		略	123 C_CUSTKEY 11	RUC C_NAME	Ruc C_ADDRESS	RINC C_CITY			
✓ II db		<u>∎</u> 1	1	Customer#000000001	j5JsirBM9P	MOROCCO			政
	1 2 M	2	2	Customer#00000002	487LW1dovn6Q4dMVym	JORDAN 1			[333
> Ineorder local	162M	<u></u> ∦ 3	3	Customer#00000003	fkRGN8n	ARGENTINA	N.		
> == part_local	3.6M	€ <u>4</u>	4	Customer#00000004	4u58h f	EGYPT 4			i i
> supplier_local	77K	5	5	Customer#00000005	hwBtxkoBF qSW4Krl	CANADA 5	5		
> 🙋 视图		6	6	Customer#00000006	g1s,pzDenUEBW3O,2 px	SAUDI ARA	2		
■ 存储过程		7	7	Customer#00000007	80kMVLQ1dK6Mbu6WG9	CHINA 0			
		8	8	Customer#00000008	j,pZ,Qp,qtFEo0r0c 92qo	PERU 6			
> iii default		9	9	Customer#00000009	vglql8H6zoyuLMFN	INDIA 6			
y in System		10	10	Customer#000000010	Vf mQ6Ug9Ucf5OKGYq fs	ETHIOPIA 9			
		11	11	Customer#000000011	cG48rYjF3Aw7xs	UNITED KI3			
		12	12	Customer#000000012	Sb4gxKs7	JORDAN 5			
		13	13	Customer#000000013	Ez3ax0D5HnUbeU	CANADA 8	3		
		14	14	Customer#000000014	h3GFMzeFf	ARGENTINA	4(
		15	15	Customer#00000015	3y4KK4CcfNwNCTP0u0p1	UNITED KIO			
		16	16	Customer#000000016	P2IQMff18er	IRAN 5			
		17	17	Customer#000000017	Js JrVHNAyCYMANzPGzvo	BRAZIL 6			
		18	18	Customer#00000018	YyukcsqlxlyuXs7	FRANCE 0			
		19	19	Customer#000000019	yO0XPkiuSWk0vN Ffc	CHINA 3			
		戚 20	20	Customer#00000020	i bGScA	RUSSIA 0			
		ይ 21	21	Customer#000000021	42E5BARt	INDIA 4	~		
						>			
		Ø(¥存 凶 取消 」 脚本			0+ : 行数: 1			
		d i	200 行已获取 - 190n	ns (1ms 获取), 7月 25, 14	:52:23				

步骤3: 使用 DBeaver 执行 SQL

连接 DBeaver 和 TDSQL-H LibraDB 实例后,您可以在 DBeaver 左侧查看到新建的 TDSQL-H LibraDB 实例连接,并可通过 SQL 编辑器执行 SQL。

可通过 SQL 编辑器执行 SQL 语句的操作示例如下。更多执行 SQL 操作,请参见 SQL 编辑器。

1. 在创建的 TDSQL-H LibraDB 实例连接下,在界面顶部选择 SQL 编辑器 > 新建 SQL 编辑器。





2. 在 SQL 编辑器中输入 SQL 语句,选中需要执行的语句,选择执行 > 执行 SQL 语句。

也可直接在 SQL 编辑器窗口左侧的主工具栏单击 > 执行SQL语句。



执行结果如下:

OBeaver 22.1.2 - < 184> Script 文件(F) 編輯(E) 导航(N) 搜索(A) SQL 編辑器 数据库(D) 窗口(W) 帮助(H)		- 0		×
· 株 ▼ ● 乾 X U SQL ▼ : 記提交 L3回滚 ぷ ▼ 曽 Auto ① ▼ : IIII	184 -		Q	E 🖉
●数据库导航 × ●	¹ <u>□</u> *<	184> Script ×		. 8
输入基格名称的一部分	• • • • • • • • • • • • • • • • • • •	<pre>\$ELECT * FROM customer_local;</pre>		~
> ■ customer local 1.2 > ■ customer local 162 > ■ part_local 3.6 > ■ supplier_local 7 > ■ 存描过程 > ■ 安雄类型 > ■ default > ① default > ① default	M M K R R	¢		>
	📗 cus	omer local 1 ×		
	of SEL	CT * FROM customer local 🕃 給入一个 SQL 表达式来过滤结果 (使用 Ctrl+Space) 🕨 💌 🗷 🖼 🕫 🕶	\leftarrow $+$	÷ •
	400	TRAC CUSTKEY VI === C NAME VI === C ADDRESS VI === C CITY VI === C NATION V	R = 1	
	Ĕ,			
	1 2	2 Customer#000000002 487/W1dovn604dMVvm IORDAN 1 JORDAN MIDE		san
	₩ 3	3 Customer#000000003 fkRGN8n ARGENTINA7 ARGENTINA AMEE		
		4 Customer#000000004 4u58h f EGYPT 4 EGYPT MIDE		
	5	5 Customer#000000005 hwBtxkoBF aSW4Krl CANADA 5 CANADA AMEF		
	6	6 Customer#000000006 g1s.pzDenUEBW30.2 pxu SAUDI ARA2 SAUDI ARABIA MIDE		80
	7	7 Customer#00000007 80kMVLQ1dK6Mbu6WG9 CHINA 0 CHINA ASIA		00
	8	8 Customer#000000008 i.pZ.Op.atFEo0r0c 92ap PERU 6 PERU AMEF		
	9	9 Customer#00000009 valal8H6zovuLMFN INDIA 6 INDIA ASIA		
	10	10 Customer#000000010 Vf mQ6Uq9Ucf5OKGYq fs ETHIOPIA 9 ETHIOPIA AFRIC		
	11	11 Customer#000000011 cG48rYjF3Aw7xs UNITED KI3 UNITED KINGDOM EURC		
	12	12 Customer#00000012 Sb4qxKs7 JORDAN 5 JORDAN MIDL		
	13	13 Customer#000000013 Ez3ax0D5HnUbeU CANADA 8 CANADA AMEF		
	14	14 Customer#000000014 h3GFMzeFf ARGENTINA0 ARGENTINA AMEF		
	15	15 Customer#000000015 3y4KK4CcfNwNCTP0u0p1Rk6 UNITED KI0 UNITED KINGDOM EURC		
	16	16 Customer#00000016 P2IQMff18er IRAN 5 IRAN MIDE		
	17	17 Customer#00000017 Js JrVHNAyCYMANzPGzvonS BRAZIL 6 BRAZIL AMEF		
	嘭 18	18 Customer#000000018 YyukcsqlxlyuXs7 FRANCE 0 FRANCE EURC		
	10	19 Customer#00000019 VOOYDEVISANDOWN EFF CHINA 3 CHINA ASIA		
		空存回 開始 長期本 「東京 東京 まいど ノントル」 1912 ・ ま : (* 2000) (* 2000 - 1 (本教) 1 (* 2000 行日共取 - 1977mr 7日 25 (* 200-26		-
	©			



使用 JDBC 连接 TDSQL-H LibraDB

最近更新时间: 2024-10-17 20:41:22

TDSQL-H LibraDB JDBC 驱动是 TDSQL-H LibraDB 提供的 Java 数据库连接接口。本文为您介绍如何使用 JDBC 连接 TDSQL-H LibraDB,并提 供示例代码。

背景信息

本文以使用 IntelliJ IDEA 工具为例,介绍如何使用 JDBC 连接 TDSQL-H LibraDB。

前提条件

- 已 创建 TDSQL-H LibraDB 实例。
- 已获取访问 TDSQL-H LibraDB 实例的账号和密码。
- 获取 JDBC 连接 TDSQL-H LibraDB 实例时的主机地址。访问 TDSQL-H LibraDB 实例方式不同,获取的主机地址不同,请根据实际选择。
 - 若通过搭建代理使用外网访问 TDSQL-H LibraDB 实例,请获取搭建代理时使用的 CVM 云服务器的公网 IP,并确保外网连接使用的 CVM 安全组已 添加 TCP:8123、TCP:9000 协议端口。本文中的操作以此方式为例。
 - 搭建代理的具体操作,请参见 连接实例分析引擎 中的外网连接相关内容。
 - 若使用镜像为 Windows 的 CVM,安装 DBeaver 并连接 TDSQL-H LibraDB 实例,请获取 TDSQL-H LibraDB 实例的内网 IP。

 镜像为 Windows 的 CVM 与 TDSQL-H LibraDB 实例必须为相同的 VPC,并确保 CVM 安全组已添加 TCP:8123、TCP:9000 协议端口。
- 已下载并安装 IntelliJ IDEA。

推荐使用 IntelliJ IDEA 版本 IntelliJ IDEA 2021.3.1,其他版本可能存在兼容性问题。本文中的 IntelliJ IDEA 示例版本为 IntelliJ IDEA 2021.3.1。

JDBC 下载

使用 IntelliJ IDEA 工具,创建 Maven 项目在对象模型 POM(Project Object Model)引入 ClickHouse 驱动依赖包。

```
<dependency>
  <groupId>ru.yandex.clickhouse</groupId>
   <artifactId>clickhouse-jdbc</artifactId>
   <version>0.3.2</version>
</dependency>
```

连接 TDSQL-H LibraDB

1. 加载 TDSQL-H LibraDB JDBC 驱动。

lass.forName("ru.yandex.clickhouse.ClickHouseDriver")

2. 通过 DriverManager 创建 Connection。

Connection connection = DriverManager.getConnection(connectionStr, username, password);

- connectionStr: 格式为 jdbc:clickhouse://" + url + ":8123 。其中, url 为前提条件获取的 JDBC 连接 TDSQL-H LibraDB 实例时的主机 地址。
- username: 访问 TDSQL-H LibraDB 实例的账号。
- password:访问 TDSQL-H LibraDB 实例账号对应的密码。
- 3. 执行查询。

```
Statement stmt = connection.createStatement();
ResultSet rs = stmt.executeQuery("SELECT foo FROM bar");
while (rs.next()) {
   // process the results
}
```



```
stmt.close();
```

connection.close();

示例代码





SQL参考 物化视图

最近更新时间: 2024-10-17 20:41:22

物化视图(Materialized View)本质是一张特殊的数据表,主要用于预先计算并保存耗时较多的操作的结果,以便在查询时直接复用,从而避免这些耗时的操 作,最终达到加速查询的目的。本文为您介绍物化视图相关命令、语法格式并提供使用示例,及物化视图在 CDC 同步场景中的数据处理过程。

创建物化视图

语法

CREATE MATERIALIZED VIEW [IF NOT EXISTS] [db.]table_name [TO[db.]name] [ON CLUSTER default_cluster] ENGINE = engine_name() ORDER BY expr [POPULATE] AS SELECT ...

参数说明

- IF NOT EXISTS: 可选,表示如果已经存在一个同名的表,则会忽略后续的创建过程。
- db.:: 可选,指定数据库的名称,默认为当前选择的数据库。
- table_name : 必填,物化视图表名称。
- TO[db.]name:可选,将物化视图的数据写入到新表中。

() 说明:

如果需要将物化视图的数据写入新表,不能使用 POPULATE 关键字。

- ON CLUSTER default_cluster : 可选,增加 ON CLUSTER default_cluster 可在 default_cluster 集群的所有服务器上创建物化视图。
- ENGINE = engine_name() : 必填,表引擎类型。
- ORDER BY : 必填,指定排序键。
- POPULATE: 可选, POPULATE 关键字。如果创建物化视图时指定了 POPULATE 关键字,则在创建时将 SELECT 子句所指定的源表数据插入到物化视 图中。不指定 POPULATE 关键字时,物化视图只会包含在物化视图创建后新写入源表的数据。

🕛 说明:

请谨慎选择指定 POPULATE 关键字。如果创建物化视图过程中源表有插入新数据,则新插入的数据不会插入到物化视图中,导致源表与物化视图数 据不一致。

• SELECT: 必填, SELECT 子句。当数据写入物化视图中 SELECT 子句所指定的源表时,插入的数据会通过 SELECT 子句查询进行转换并将最终结 果插入到物化视图中。

() 说明:

- SELECT 语句不能包含 PREWHERE 、 SAMPLE 、 FINAL 。
- 当物化视图的表涉及多表 JOIN 时,只有向最左边的表插入数据时才会更新。
 当物化视图 SELECT 使用类似 FROM A JOIN B 的多表 JOIN 时,向 B 表插入数据时不会触发更新,只有向 A 表插入数据时才会触发物化 视图更新,此时物化视图更新才会去 JOIN B 表中的数据。

使用示例

1. 连接 LibraSQL 分析引擎后在 TDSQL-H LibraDB 中创建源表。

```
CREATE TABLE view.basic_test ON CLUSTER default_cluster (
`id` UInt64,
```





2. 在源表中插入数据。



3. 创建物化视图,并指定 POPULATE 关键字。



4. 查询物化视图。

<pre>SELECT * FROM view.basic_modify_view;</pre>
返回结果
FROM view.basic_modify_view
2 rows in set. Elapsed: 0.005 sec.

可查询到源表 basic_test 符合指定 POPULATE 关键字的数据已插入到物化视图 basic_modify_view 中。

① 说明: 如果没有指定 POPULATE 关键字,查询结果将为空。因为不指定 POPULATE 关键字时,物化视图只会包含在物化视图创建后新写入源表的数据。

5. 在源表继续插入数据。

```
INSERT INTO view.basic_test VALUES
(5, 'click', 15, 0),
(6, 'click', 25, 0),
(7, 'modify', 16,0),
(8,'modify', 29, 0);
```

6. 再次查询物化视图。



# 返回结果 SELECT *	
FROM view.basic_modi	fy_view
idtypecnt_	version
	0
	version-
7 modify 16	0
	0
	·I

可查询到源表 basic_test 符合指定 POPULATE 关键字的数据已插入到物化视图 basic_modify_view 中。

删除物化视图

语法

DROP TABLE [IF EXISTS] [db.]table_name [ON CLUSTER default_cluster]

参数说明

- IF EXISTS : 可选。如果没有指定 IF EXISTS 且物化视图不存在会返回报错。
- db.::可选,指定数据库的名称,默认为当前选择的数据库。
- table_name : 必填,物化视图表名称。
- ON CLUSTER default_cluster: 可选,如果创建物化视图时增加 ON CLUSTER default_cluster,删除时也需要增加。

使用示例

DROP TABLE view.basic_test_view ON CLUSTER default_cluster;

CDC 场景物化视图使用说明

原生 ClickHouse 实时 update 和 delete 性能较弱,TDSQL-H LibraDB 对 CDC 同步的数据表,使用 insert 代替 update 和 delete,额外为每个表 新增字段 _sign 和 _version ,无需 FINAL 即可保证目的端的数据与源端完全一致。详细说明请参见 对 update 和 delete 优化 。 在 CDC 同步场景中,TDSQL-H LibraDB 并未针对物化视图表进行处理。针对数据的插入、更新和删除操作,TDSQL-H LibraDB 会使用 _sign 和 _version 这两个字段的不同组合展示数据处理过程。

以下提供示例介绍在 CDC 同步场景中,源端数据更新时,TDSQL-H LibraDB 侧物化视图如何处理数据。

使用示例

1. 在待创建 CDC 任务的源端创建源表。

```
CREATE TABLE view.basic_test (
`id` int NOT NULL,
`type` varchar(50) DEFAULT '',
```

-) ENGINE=InnoDB DEFAULT CHARSET=utf8;
- 2. 在源表中插入数据。



INSERT INTO view.basic_test VALUES
(1, "click", 100),
(2, "insert", 101),
(3, "click", 50),
(4, "insert", 40);

3. 创建 CDC 任务,具体操作请参见 添加 CDC 。

其中,CDC 任务配置如下:

- 同步类型为初始结构 + 初始全量数据 + 增量数据。
- 同步对象为指定对象,源端库表为 view.basic_test , 同步到目的端名称保持不变,仍为 view.basic_test 。
- 其它保持默认。
- 4. CDC 任务创建成功后,连接 LibraSQL 分析引擎后在 TDSQL-H LibraDB 中创建物化视图(包括本地表和分布式表)。

```
# dige>webs, 其中 b0bbb924-d555-431f-9228-51c9e4767413 为 usid, 请根据实际替换, 且 SELSCT 子句必须增加 _sign、
_version子段。
CREATE MATERIALIZED VIEW view.basic_test_view_local ON CLUSTER default_cluster
ENGINE = ReplicatedReplacingWergeTree('/clickhouse/tables/b0bdb924-d555-431f-8228-
51c9e4767413/(shard)', '(replica)', _version)
ORDER BY id POPULATE AS
SELECT
id,
type,
ent,
__sign,
__version
FROM view.basic_test_local;
//
//
// dige>fats
CREATE TABLE view.basic_test_view ON CLUSTER default_cluster
(
  'id' Int32,
  'type' Nullable(String),
  'ent' Nullable(String),
  'ent' Nullable(Int32),
  '_asign' Int8,
  '_version' UInt64
)
ENGINE = Distributed('default_cluster', 'view', 'basic_test_view_local',
cityHash64(toString(tuple(id))));
```

5. 查询物化视图。

	iew.basi	c_test_	view	
—id—			- sign v	version-
	click	100		
		101		
	click	50		
		40		
		LI.		



可查询到 CDC 任务源端库表 view.basic_test 数据已插入到物化视图 view.basic_test_view 中。

- 6. 在 CDC 任务源端继续插入数据、更新数据、删除数据,观察 CDC 任务目的端物化视图的分布式表数据变化。
- 插入数据

在 CDC 任务源端插入数据。

```
INSERT INTO view.basic_test VALUES
(5, "new_insert", 150);
```

在 CDC 任务目的端物化视图的分布式表查询数据。

SELECT * FROM view.basic_test_view;
返回结果 BELECT * FROM view. basic_test_view
-idtype
idtypecntsignversion 1 click 100 1 1 2 insert 101 1 1 3 click 50 1 1 4 insert 40 1 1
5 rows in set. Elapsed: 0.010 sec.
的端数据变化: 源端插入一条数据,目的端同步插入一条数据 _sign = 1 、 _version = 65412 ,即 insert 一条数据, _sign = 1 、

_version **递增。**

• 删除数据

在 CDC 任务源端删除数据。

LETE FROM view.basic_test WHERE id=5;

在 CDC 任务目的端物化视图的分布式表查询数据。

SELEC:	T * FROM view	.basic_	_test_vie	ew;
# 返回				
	view.basic_te	st_viev		
id—		-cnt-	—_sign—	version-
5 	new_insert 		1	
_—id—		-cnt-	—_sign—	version-
	new_insert		-1	
r—id—	⊤-typecn	tsi	lgnve	ersion—
1	click 10			
2				
3	click 5			





目的端数据变化: 源端删除一条数据,目的端同步数据将 delete 转换为 insert 模式,插入一条数据 _sign = -1 、_version = 66316 ,即 insert 一条数据,值为原值, _sign=-1 、_version 递增。

更新数据

在 CDC 任务源端更新数据。

UPDATE view.basic_test SET cnt=300 WHERE id=1;

在 CDC 任务目的端物化视图的分布式表查询数据。

目的端数据变化: 源端更新一条数据,目的端同步数据将 update 转换为 delete+insert 模式,具体说明如下:

```
      6.1 插入一条数据 _sign = -1 、 _version = 67021 ,即 delete 一条数据,值为原值, sign=-1 、 _version 递增。

      6.2 插入一条数据 _sign = 1 、 _version = 67021 ,即 insert 一条数据,值为新值, sign=1 、 _version 和 delete 数据的 _version 值

      相同。
```

物化视图数据处理

源端插入数据、删除数据、更新数据,目的端物化视图对数据处理过程总结如下:

插入数据 insert 一条数据, _sign = 1 、_version 递增。
删除数据 insert 一条数据,值为原值, _sign=-1 、_version 递增。
更新数据

```
○ delete 一条数据,值为原值, sign=-1 、 _version 递增。
```



○ insert 一条数据,值为新值, sign=1 、 _version 和 delete 数据的 _version 值相同。

参考文档

创建物化视图的更多信息,请参见 Create Materialized View。

修改键列数据类型

腾田元

最近更新时间: 2024-10-17 20:41:22

TDSQL-H LibraDB 支持修改主键、排序键和分区键的数据类型。本文为您介绍如何修改数据类型、修改数据类型支持的范围和修改数据类型的功能限制。

修改键列数据类型支持的范围

主键、排序键可修改的数据类型支持 Int 类型、UInt 类型、Float 类型、Int 类型与 UInt 类型交叉四种场景,分区键可修改的数据类型支持 Int 类型、UInt 类 型、Int 类型与 UInt 类型交叉三种场景。

修改数据类型仅支持从小类型修改为大类型,具体支持范围如下:

- Int 类型: Int8、Int16、Int32、Int64。
- UInt 类型: UInt8、UInt16、UInt32、UInt64。
- Float 类型: Float32、Float64。
- Int 类型与 UInt 类型交叉:
 - UInt8、Int16、Int32、Int64。
 - UInt16、Int32、Int64。
 - UInt32、Int64

使用限制

某些场景下,会使用主键或排序键作为 ShardingKey,并且通过 CityHash64 的函数进行数据写入时负载均衡。针对该类场景,同样的数据在修改数据类型 前后数据类型不同,计算出来的 CityHash64 的值也不同,导致相同的数据可能存储在不同节点上,造成数据异常。

针对该场景,如果使用 TDSQL-H LibraDB 的 CDC 能力将源端数据同步至 TDSQL-H LibraDB,修改源端数据类型,可保证相同数据存储在同一节点 上,因为 CDC 同步任务会将 ShardingKey 各种数据类型均转换为 String 类型。

针对该场景,如果创建表和修改数据类型均在 TDSQL-H LibraDB 侧操作,创建表时推荐使用以下语句定义 ShardingKey。

🕛 说明:

- 创建表时,一般不需要指定主键,只要指定排序键即可。当没有指定主键时,排序键则同时作为主键。
- 当同时指定主键和排序键时,主键必须是排序键的前缀。
- 分区键不支持指定数据类型为 Float 的列。

修改键列数据类型

语法

ALTER TABLE [db.]table_name [ON CLUSTER default_cluster] modify column column_name type;

参数说明

- db.::可选,指定数据库的名称,默认为当前选择的数据库。
- table_name : 必填, 表名。
- ON CLUSTER default_cluster : 可选, 增加 ON CLUSTER default_cluster 可在 default_cluster 集群的所有服务器上创建库表。



- column_name : 必填,待修改的主键列名。
- type:必填,待修改的数据类型。

使用示例

用户 A 表中主键为 id, 是一个递增序列, 原始主键类型为 UInt32 类型, 但是随着数据的增多, UInt32 已经存在不下当前数据, 此时 需要修改主键类型扩大数据存储空间, 用户可将 A 表的主键 id 从 UInt32 类型修改成 UInt64。 ALTER TABLE testdb.A ON CLUSTER default cluster modify column id UInt64;

常见错误码

最近更新时间: 2024-10-17 20:41:22

本文为您介绍 TDSQL-H LibraDB 的常见的错误码及处理方法。

错误码	错误码说明	错误码示例	分析与处理方法
16	查不到表字段	code: 16, message: No such column subject_type in table express_order_merge_snapshot	 问题分析: 该报错主要原因为对应的 CDC 任务创建时间较早尚不支持 DDL 同步。 处理方法: 请通过 在线支持 联系我们处理。
44	修改列名或数据 类型问题	Code: 44. DB::Exception: Received from xx.xx.xx.xxx. DB::Exception: Trying to ALTER key column name.	 问题分析: LibraSQL分析引擎不支持修改表的主键或分区键字段的数据类型。 处理方法: 不修改主键或者分区键字段的数据类型。
122	Zookeeper 中 的表列结构与本 地表结构不同	Code: 122, e.displayText() = DB::Exception:Table columns structure in Zookeeper is different from local table structure	• 问题分析: 创建表前删除同名表时未添加 on cluster,仅删除某一个 实例的本地表,其他实例的本地表还有残留,导致元数据信息不 一致。 • 处理方法: 添加 on cluster 再次删除同名表后再次创建表。
159	DDL 语句执行 超时	Code: 159. DB::Exception: Received from localhost:9000. DB::Exception: Watching task xxx is executing longer than distributed_ddl_task_timeout (=180) seconds. There are 1 unfinished hosts (0 of them are currently active), they are going to execute the query in background.	 问题分析: DDL 语句在某个集群节点中执行失败。 处理方法: 将 DDL 执行的超时时间参数
210	连接被拒绝	Code: 210 DB:NetException: Connection refused	• 问题分析: 集群负载过重,导致无可用连接。 • 处理方法: 请稍后重试。
225	Zookeeper 会 话已过期	code: 225, message: Zookeeper session has been expired.	 问题分析: XID 溢出导致 LibraSQL 强制让 Zookeeper 会话过期。 处理方法: 请稍后重试。
241	超出内存限制	Code: 241. DB::Exception: Received from localhost:xxx. DB::Exception: Memory limit (for query) exceeded	 问题分析: SQL 使用的内存过大。 处理方法: 进行 SQL 调优或者升级实例配置。
242	表为只读模式	Code 242. DB::Exception: Received from localhost:xxx. DB::Exception: Table is in readonly mode (Zookeeper path: /xxx)	 问题分析: Zookeeper 压力过大,导致表处于只读模式。 处理方法: 请稍后重试。
253	副本已存在	code: 253, Replica /xxx/tables/2/t1/replicas/xx.xx.xx.xx	● 问题分析:



		already exists.	DDL 清理任务是异步的。当执行 Drop 语句时, Zookeeper上的元数据没有清除,导致再创建相同的表使用相 同的 Zookeeper 路径报错。 • 处理方法: 调整 database_atomic_wait_for_drop_and_detach_sync hronously 配置参数,将异步删除任务修改成同步删除任务。
359	表或分区未删除	Code: 359, Table or Partition in xxx was not dropped	 问题分析: LibraSQL 对删除数据提供保护。默认最大删除50 GB的数据。 处理方法: 登录 TDSQL-H LibraDB 控制台,将分析引擎参数 max_table_size_to_drop 参数运行值设置为0,具体操作 请参见 设置参数。
420	不允许更新键列	Code: 420. DB::Exception: Received from xx.xx.xx.xx:DB::Exception:Cannot UPDATE key column xxx	• 问题分析: 不支持更新主键或分区键。 • 处理方法: 不更新主键或分区键。
517	副本上的元数据 未与 Zookeeper 中 的通用元数据保 持同步	Code: 517, e.displayText() = DB::Exception: Metadata on replica is not up to date with common metadata in Zookeeper. Cannot alter	 问题分析: LibraSQL DDL 处理机制是异步的。各个副本之间的元数据没有一致的情况下,进行 DDL 会报错。 处理方法: 将系统参数修改为 replication_alter_partitions_sync = 2 。