数据仓库服务

快速入门

文档版本 16

发布日期 2024-05-07

华为云计算技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

1 免费体验

数据分析实验室是华为云官方实验平台，实验环境为华为云现网环境，整个实验过程完全免费。供您在最短时间内体验GaussDB(DWS)真实环境，快速上手。

开发者可通过表1-1实验手册指导，使用环境中预置的华为云账号，一键创建GaussDB(DWS)实验环境，在云端体验GaussDB(DWS)的数据导入、访问MRS集群、多维度分析、权限管理、JDBC开发、性能调优等场景。

表 1-1 实验列表

场景	实验名称	实验描述	预计时长
导入分析	快速入门	一键式创建GaussDB(DWS)集群，上传csv本地数据到云存储OBS桶，通过创建OBS外表实现数据导入并简单分析。	1h
	零售业经营多维度分析	环境已预置样例数据在OBS桶，通过创建OBS外表导入样例数据，再使用聚合函数、group by、order by、视图进行多维度分析。	1.5h
SQL on Hadoop	导入MRS-Hive数据源	创建数据仓库集群GaussDB(DWS)，并导入MRS的Hive数据，实现跨集群进行大数据融合分析。	1.5h
二次开发	使用Java进行二次开发	使用JDBC驱动连接GaussDB(DWS)进行二次开发，熟悉简单的适配DWS的Java开发用例。	1.5h
数据迁移	从老DWS集群迁移数据到新DWS集群	本实验指导用户创建数据仓库集群GaussDB(DWS)并将老GaussDB(DWS)整库迁移到新的GaussDB(DWS)。同时，针对使用自增序列导致的性能问题场景，指导用户排查原因并提供优化方法。	2~3h
	基于gds实现跨集群数据互联互通	本实验通过部署GDS服务器，使用GDS导入导出的并发能力，实现双DWS集群之间1500万行数据分钟级迁移。	2h

场景	实验名称	实验描述	预计时长
安全管理	权限管理	通过实验创建不同用户，不同Schema，基于权限管理实现数据的隔离和互访，了解用户、角色的关系，了解grant的基本用法，了解基于角色的权限管理(RBAC)。	1.5h
	使用数据脱敏实现卡号等隐私信息屏蔽	本实验通过创建数据仓库服务GaussDB(DWS)并使用DWS的数据脱敏功能，针对不同用户设置部分数据列的屏蔽，实现敏感数据脱敏，确保数据安全。	1h
高级特性	冷热数据管理	指导用户创建数据仓库集群GaussDB(DWS)，并创建冷热分区表实现冷热数据分区管理，不仅可以提高数据分析性能还能降低业务成本。	1h
调优	性能调优	指导用户使用GaussDB(DWS)进行性能调优。通过本实验掌握通过EXPLAIN语句查询执行计划的方法，了解GaussDB(DWS)几种常见的SQL调优手段。	2h
云原生3.0	云原生3.0数仓-存算分离	本实验指导用户创建GaussDB(DWS)新一代Serverless云原生数仓，并体验Serverless存算分离架构下的极致查询。	2h
	云原生3.0数仓-湖仓一体	本实验指导用户创建GaussDB(DWS)新一代Serverless云原生数仓，通过EXTERNAL SCHEMA访问MRS的Hive数据，体验湖仓一体、存算分离等极致查询的高级特性。	2h

2 交通卡口通行车辆分析

本实践将演示交通卡口车辆通行分析，将加载8.9亿条交通卡口车辆通行模拟数据到数据仓库单个数据库表中，并进行车辆精确查询和车辆模糊查询，展示GaussDB(DWS)对于历史详单数据的高性能查询能力。

说明

GaussDB(DWS) 已预先将样例数据上传到OBS桶的“traffic-data”文件夹中，并给所有华为云用户赋予了该OBS桶的只读访问权限。

操作流程

本实践预计时长40分钟，基本流程如下：

1. **准备工作**
2. **步骤一：创建集群**
3. **步骤二：使用Data Studio连接集群**
4. **步骤三：导入交通卡口样例数据**
5. **步骤四：车辆分析**

支持区域

当前已上传OBS数据的区域如**表2-1**所示。

表 2-1 区域和 OBS 桶名

区域	OBS桶名
华北-北京一	dws-demo-cn-north-1
华北-北京二	dws-demo-cn-north-2
华北-北京四	dws-demo-cn-north-4
华北-乌兰察布一	dws-demo-cn-north-9
华东-上海一	dws-demo-cn-east-3
华东-上海二	dws-demo-cn-east-2

区域	OBS桶名
华南-广州	dws-demo-cn-south-1
华南-广州友好	dws-demo-cn-south-4
中国-香港	dws-demo-ap-southeast-1
亚太-新加坡	dws-demo-ap-southeast-3
亚太-曼谷	dws-demo-ap-southeast-2
拉美-圣地亚哥	dws-demo-la-south-2
非洲-约翰内斯堡	dws-demo-af-south-1
拉美-墨西哥城一	dws-demo-na-mexico-1
拉美-墨西哥城二	dws-demo-la-north-2
莫斯科二	dws-demo-ru-northwest-2
拉美-圣保罗一	dws-demo-sa-brazil-1

准备工作

- 已注册账号，且在使用GaussDB(DWS) 前检查账号状态，账号不能处于欠费或冻结状态。
- 获取此账号的“AK/SK”。

步骤一：创建集群

步骤1 登录管理控制台。

步骤2 在“服务列表”中，选择“大数据 > 数据仓库服务 GaussDB(DWS)”。

步骤3 左侧导航栏单击“集群管理”，进入页面后，单击右上角的“创建数据仓库集群”按钮。

步骤4 参见表2-2进行基础配置。

表 2-2 基础配置

参数名称	配置方式
区域	选择“华北-北京四”。 **说明** 本指导以“华北-北京四”为例进行介绍，如果您需要选择其他区域进行操作，请确保所有操作均在同一区域进行。
可用分区	可用区2
产品类型	标准数仓
计算类型	弹性云服务器

参数名称	配置方式
存储类型	SSD云盘
CPU架构	X86
节点规格	dws2.m6.4xlarge.8（16 vCPU | 128GB | 2000GB SSD） **说明** 如规格售罄，可选择其他可用区或规格。
热数据存储	100GB / 节点
节点数量	3

步骤5 信息核对无误，单击“下一步：网络配置”，参见表2-3进行网络配置。

表 2-3 网络配置

参数名称	配置方式
虚拟私有云	vpc-default
子网	subnet-default(192.168.0.0/24)
安全组	自动创建安全组
公网访问	现在购买
宽带	1Mbit/s
弹性负载均衡	暂不使用

步骤6 信息核对无误，单击“下一步：高级配置”，参见表2-4进行网络配置。

表 2-4 高级配置

参数名称	配置方式
集群名称	dws-demo
集群版本	使用推荐版本，例如8.1.3.311
管理员用户	dbadmin
管理员密码	-
确认密码	-
数据库端口	8000
企业项目	default
高级配置	默认配置

步骤7 单击“下一步：确认配置”，确认无误后，单击“立即购买”

步骤8 等待约6分钟，待集群创建成功后，单击集群名称前面的，弹出集群信息，记录下“公网访问地址”。

图 2-1 集群信息

----结束

步骤二：使用 Data Studio 连接集群

步骤1 请确保客户端主机已安装JDK 1.8.0以上版本，并进入“此电脑 > 属性 > 高级系统设置 > 环境变量”设置JAVA_HOME（例如C:\Program Files\Java\jdk1.8.0_191），并在变量path中添加“;%JAVA_HOME%\bin”。

步骤2 在GaussDB(DWS) 控制台的“连接管理”页面，下载Data Studio客户端。

步骤3 解压下载的Data Studio软件包，进入解压目录后，双击Data Studio.exe启动客户端。

步骤4 在Data Studio主菜单中选择“文件 > 新建连接”，并在弹出框中参照**表2-5**所示配置。

表 2-5 Data Studio 软件配置

参数名称	配置方式
数据库类型	GaussDB(DWS)
名称	dws-demo
主机	dws-demov.dws.huaweicloud.com 与**步骤一：创建集群**查询到的“公网访问地址”一致。
端口	8000
数据库	gaussdb
用户名	dbadmin
密码	-
启用SSL	不启用

步骤5 单击“确定”。

----结束

步骤三：导入交通卡口样例数据

使用SQL客户端工具连接到集群后，就可以在SQL客户端工具中，执行以下步骤导入交通卡口车辆通行的样例数据并执行查询。

步骤1 执行以下语句，创建traffic数据库。

```
CREATE DATABASE traffic encoding 'utf8' template template0;
```

步骤2 执行以下步骤切换为连接新建的数据库。

1. 在Data Studio客户端的**“对象浏览器”**窗口，右键单击数据库连接名称，在弹出菜单中单击“刷新”，刷新后就可以看到新建的数据库。
2. 右键单击“traffic”数据库名称，在弹出菜单中单击“打开连接”。
3. 右键单击“traffic”数据库名称，在弹出菜单中单击“打开新的终端”，即可打开连接到指定数据库的SQL命令窗口，后面的步骤，请全部在该命令窗口中执行。

步骤3 执行以下语句，创建用于存储卡口车辆信息的数据库表。

```
CREATE SCHEMA traffic_data;
SET current_schema= traffic_data;
DROP TABLE if exists GCJL;
CREATE TABLE GCJL
(
    kkbh   VARCHAR(20),
    hphm   VARCHAR(20),
    gcsj   DATE ,
    cplx   VARCHAR(8),
    cllx   VARCHAR(8),
    csys   VARCHAR(8)
)
with (orientation = column, COMPRESSION=MIDDLE)
distribute by hash(hphm);
```

步骤4 创建外表。外表用于识别和关联OBS上的源数据。

须知

- 其中，*<obs_bucket_name>*代表OBS桶名，仅支持部分区域，当前支持的区域和对应的OBS桶名请参见**支持区域**。GaussDB(DWS) 集群不支持跨区域访问OBS桶数据。
- 本实践以“华北-北京四”地区为例，可填入dws-demo-cn-north-4，<Access_Key_Id>和<Secret_Access_Key>替换为实际值，在**准备工作**获取。
- 认证用的AK和SK硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。
- 创建外表如果提示“ERROR: schema "*xxx*" does not exist Position”，则说明schema不存在，请先参照上一步创建schema。

```
CREATE SCHEMA tpchobs;
SET current_schema = 'tpchobs';
DROP FOREIGN table if exists GCJL_OBS;
CREATE FOREIGN TABLE GCJL_OBS
(
    like traffic_data.GCJL
)
```

```
SERVER gsmpp_server
OPTIONS (
        encoding 'utf8',
        location 'obs://<obs_bucket_name>/traffic-data/gcxx',
        format 'text',
        delimiter ',',
        access_key '<Access_Key_Id>',
        secret_access_key '<Secret_Access_Key>',
        chunksize '64',
        IGNORE_EXTRA_DATA 'on'
);
```

步骤5 执行以下语句，将数据从外表导入到数据库表中。

```
INSERT INTO traffic_data.GCJL SELECT * FROM tpchobs.GCJL_OBS;
```

导入数据需要一些时间，请耐心等待。

----结束

步骤四：车辆分析

1. **执行Analyze**

   用于收集与数据库中普通表内容相关的统计信息，统计结果存储在系统表PG_STATISTIC中。执行计划生成器会使用这些统计数据，以生成最有效的查询执行计划。

   执行以下语句生成表统计信息：

   ```
   ANALYZE;
   ```

2. **查询数据表中的数据量**

   执行如下语句，可以查看已加载的数据条数。

   ```
   SET current_schema= traffic_data;
   SELECT count(*) FROM traffic_data.gcjl;
   ```

3. **车辆精确查询**

   执行以下语句，指定车牌号码和时间段查询车辆行驶路线。GaussDB(DWS) 在应对点查时秒级响应。

   ```
   SET current_schema= traffic_data;
   SELECT hphm, kkbh, gcsj
   FROM traffic_data.gcjl
   where hphm =  'YD38641'
   and gcsj between '2016-01-06' and '2016-01-07'
   order by gcsj desc;
   ```

4. **车辆模糊查询**

   执行以下语句，指定车牌号码和时间段查询车辆行驶路线，GaussDB(DWS) 在应对模糊查询时秒级响应。

   ```
   SET current_schema= traffic_data;
   SELECT hphm, kkbh, gcsj
   FROM traffic_data.gcjl
   where hphm like  'YA23F%'
   and kkbh in('508', '1125', '2120')
   and gcsj between '2016-01-01' and '2016-01-07'
   order by hphm,gcsj desc;
   ```

3 某公司供应链需求分析

本实践将演示从OBS加载样例数据集到GaussDB(DWS) 集群中并查询数据的流程，从而向您展示GaussDB(DWS) 在数据分析场景中的多表分析与主题分析。

说明

GaussDB(DWS) 已经预先生成了1GB的TPC-H-1x的标准数据集，已将数据集上传到了OBS桶的tpch文件夹中，并且已赋予所有华为云用户该OBS桶的只读访问权限，用户可以方便的进行导入。

操作流程

本实践预计时长60分钟，基本流程如下：

1. **准备工作**
2. **步骤一：导入公司样例数据**
3. **步骤二：多表分析与主题分析**

支持区域

当前已上传OBS数据的区域如**表3-1**所示。

表 3-1 区域和 OBS 桶名

区域	OBS桶名
华北-北京一	dws-demo-cn-north-1
华北-北京二	dws-demo-cn-north-2
华北-北京四	dws-demo-cn-north-4
华北-乌兰察布一	dws-demo-cn-north-9
华东-上海一	dws-demo-cn-east-3
华东-上海二	dws-demo-cn-east-2
华南-广州	dws-demo-cn-south-1
华南-广州友好	dws-demo-cn-south-4

区域	OBS桶名
中国-香港	dws-demo-ap-southeast-1
亚太-新加坡	dws-demo-ap-southeast-3
亚太-曼谷	dws-demo-ap-southeast-2
拉美-圣地亚哥	dws-demo-la-south-2
非洲-约翰内斯堡	dws-demo-af-south-1
拉美-墨西哥城一	dws-demo-na-mexico-1
拉美-墨西哥城二	dws-demo-la-north-2
莫斯科二	dws-demo-ru-northwest-2
拉美-圣保罗一	dws-demo-sa-brazil-1

场景描述

了解GaussDB(DWS)的基本功能和数据导入，对某公司与供应商的订单数据分析，分析维度如下：

1. 分析某地区供应商为公司带来的收入，通过该统计信息可用于决策在给定的区域是否需要建立一个当地分配中心。
2. 分析零件/供货商关系，可以获得能够以指定的贡献条件供应零件的供货商数量，通过该统计信息可用于决策在订单量大，任务紧急时，是否有充足的供货商。
3. 分析小订单收入损失，通过查询得知如果没有小量订单，平均年收入将损失多少。筛选出比平均供货量的20%还低的小批量订单，如果这些订单不再对外供货，由此计算平均一年的损失。

准备工作

- 已注册账号，且在使用GaussDB(DWS) 前检查账号状态，账号不能处于欠费或冻结状态。
- 获取此账号的“AK/SK”。
- 已创建集群，并已使用Data Studio连接集群，参见**交通卡口通行车辆分析**。

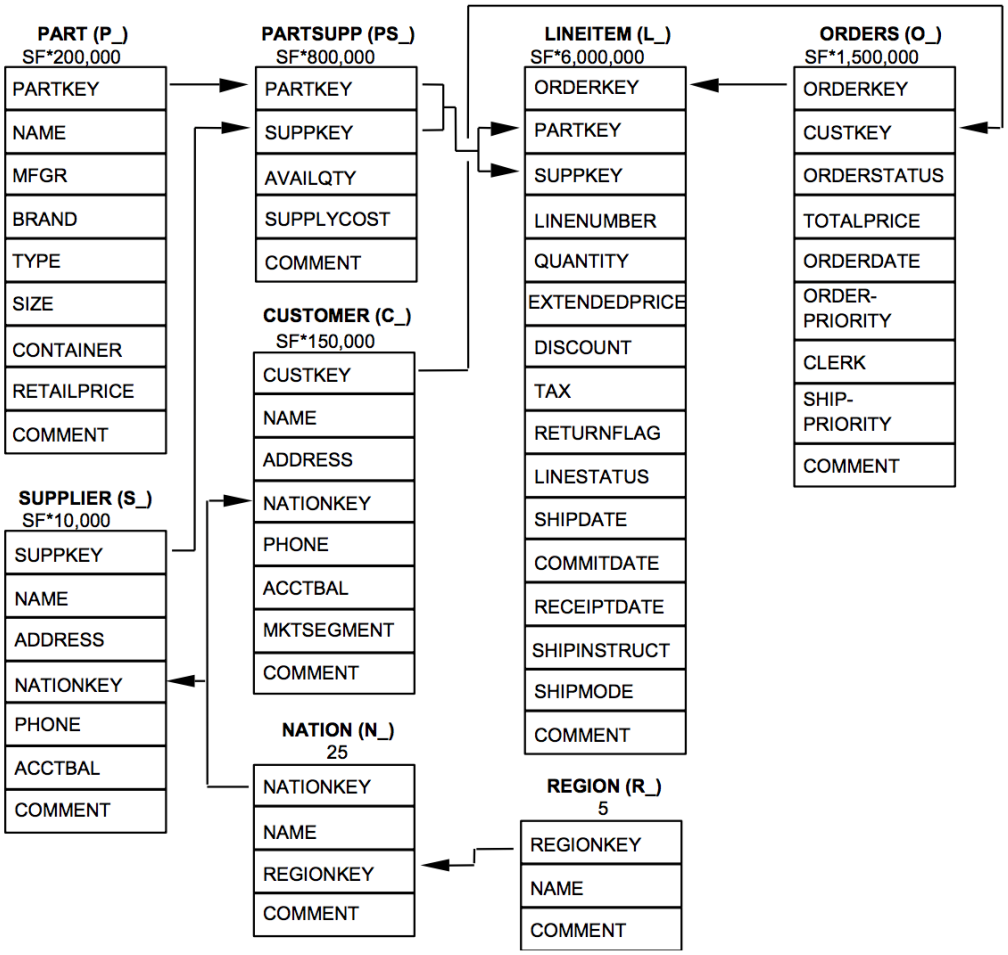
步骤一：导入公司样例数据

使用SQL客户端工具连接到集群后，就可以在SQL客户端工具中，执行以下步骤导入TPC-H样例数据并执行查询。

步骤1 创建数据库表。

TPC-H样例包含8张数据库表，其关联关系如**图3-1**所示。

图 3-1 TPC-H 数据表



复制并执行下列表创建语句，在gaussdb数据库中创建对应的数据表。

```
CREATE SCHEMA tpch;
SET current_schema = tpch;

DROP TABLE if exists region;
CREATE TABLE REGION
(
        R_REGIONKEY  INT NOT NULL ,
        R_NAME       CHAR(25) NOT NULL ,
        R_COMMENT    VARCHAR(152)
)
with (orientation = column, COMPRESSION=MIDDLE)
distribute by replication;

DROP TABLE if exists nation;
CREATE TABLE NATION
(
        N_NATIONKEY  INT NOT NULL,
        N_NAME       CHAR(25) NOT NULL,
        N_REGIONKEY  INT NOT NULL,
        N_COMMENT    VARCHAR(152)
)
with (orientation = column,COMPRESSION=MIDDLE)
distribute by replication;

DROP TABLE if exists supplier;
CREATE TABLE SUPPLIER
```



```
(
    S_SUPPKEY     BIGINT NOT NULL,
    S_NAME        CHAR(25) NOT NULL,
    S_ADDRESS     VARCHAR(40) NOT NULL,
    S_NATIONKEY   INT NOT NULL,
    S_PHONE       CHAR(15) NOT NULL,
    S_ACCTBAL     DECIMAL(15,2) NOT NULL,
    S_COMMENT     VARCHAR(101) NOT NULL
)
with (orientation = column,COMPRESSION=MIDDLE)
distribute by hash(S_SUPPKEY);

DROP TABLE if exists customer;
CREATE TABLE CUSTOMER
(
    C_CUSTKEY     BIGINT NOT NULL,
    C_NAME        VARCHAR(25) NOT NULL,
    C_ADDRESS     VARCHAR(40) NOT NULL,
    C_NATIONKEY   INT NOT NULL,
    C_PHONE       CHAR(15) NOT NULL,
    C_ACCTBAL     DECIMAL(15,2)   NOT NULL,
    C_MKTSEGMENT  CHAR(10) NOT NULL,
    C_COMMENT     VARCHAR(117) NOT NULL
)
with (orientation = column,COMPRESSION=MIDDLE)
distribute by hash(C_CUSTKEY);

DROP TABLE if exists part;
CREATE TABLE PART
(
    P_PARTKEY     BIGINT NOT NULL,
    P_NAME        VARCHAR(55) NOT NULL,
    P_MFGR        CHAR(25) NOT NULL,
    P_BRAND       CHAR(10) NOT NULL,
    P_TYPE        VARCHAR(25) NOT NULL,
    P_SIZE        BIGINT NOT NULL,
    P_CONTAINER   CHAR(10) NOT NULL,
    P_RETAILPRICE DECIMAL(15,2) NOT NULL,
    P_COMMENT     VARCHAR(23) NOT NULL
)
with (orientation = column,COMPRESSION=MIDDLE)
distribute by hash(P_PARTKEY);

DROP TABLE if exists partsupp;
CREATE TABLE PARTSUPP
(
    PS_PARTKEY     BIGINT NOT NULL,
    PS_SUPPKEY     BIGINT NOT NULL,
    PS_AVAILQTY    BIGINT NOT NULL,
    PS_SUPPLYCOST  DECIMAL(15,2)  NOT NULL,
    PS_COMMENT     VARCHAR(199) NOT NULL
)
with (orientation = column,COMPRESSION=MIDDLE)
distribute by hash(PS_PARTKEY);

DROP TABLE if exists orders;
CREATE TABLE ORDERS
(
    O_ORDERKEY       BIGINT NOT NULL,
    O_CUSTKEY        BIGINT NOT NULL,
    O_ORDERSTATUS    CHAR(1) NOT NULL,
    O_TOTALPRICE     DECIMAL(15,2) NOT NULL,
    O_ORDERDATE      DATE NOT NULL ,
    O_ORDERPRIORITY  CHAR(15) NOT NULL,
    O_CLERK          CHAR(15) NOT NULL ,
    O_SHIPPRIORITY   BIGINT NOT NULL,
    O_COMMENT        VARCHAR(79) NOT NULL
)
with (orientation = column,COMPRESSION=MIDDLE)
```

```
distribute by hash(O_ORDERKEY);

DROP TABLE if exists lineitem;
CREATE TABLE LINEITEM
(
    L_ORDERKEY    BIGINT NOT NULL,
    L_PARTKEY     BIGINT NOT NULL,
    L_SUPPKEY     BIGINT NOT NULL,
    L_LINENUMBER  BIGINT NOT NULL,
    L_QUANTITY    DECIMAL(15,2) NOT NULL,
    L_EXTENDEDPRICE  DECIMAL(15,2) NOT NULL,
    L_DISCOUNT    DECIMAL(15,2) NOT NULL,
    L_TAX         DECIMAL(15,2) NOT NULL,
    L_RETURNFLAG  CHAR(1) NOT NULL,
    L_LINESTATUS  CHAR(1) NOT NULL,
    L_SHIPDATE    DATE NOT NULL,
    L_COMMITDATE  DATE NOT NULL ,
    L_RECEIPTDATE DATE NOT NULL,
    L_SHIPINSTRUCT CHAR(25) NOT NULL,
    L_SHIPMODE     CHAR(10) NOT NULL,
    L_COMMENT      VARCHAR(44) NOT NULL
)
with (orientation = column,COMPRESSION=MIDDLE)
distribute by hash(L_ORDERKEY);
```

步骤2 创建外表。外表用于识别和关联OBS上的源数据。

须知

- 其中，*<obs_bucket_name>*代表OBS桶名，仅支持部分区域，当前支持的区域和对应的OBS桶名请参见**支持区域**。GaussDB(DWS) 集群不支持跨区域访问OBS桶数据。
- 本实践以“华北-北京四”地区为例，可填入dws-demo-cn-north-4，<Access_Key_Id>和<Secret_Access_Key>替换为实际值，在**准备工作**获取。
- 认证用的AK和SK硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。
- 创建外表如果提示“ERROR: schema "*xxx*" does not exist Position”，则说明schema不存在，请先参照上一步创建schema。

```
CREATE SCHEMA tpchobs;
SET current_schema='tpchobs';
DROP FOREIGN table if exists region;
CREATE FOREIGN TABLE REGION
(
    like tpch.region
)
SERVER gsmpp_server
OPTIONS (
    encoding 'utf8',
    location 'obs://<obs_bucket_name>/tpch/region.tbl',
    format 'text',
    delimiter '|',
    access_key '<Access_Key_Id>',
    secret_access_key '<Secret_Access_Key>',
    chunksize '64',
    IGNORE_EXTRA_DATA 'on'
);

DROP FOREIGN table if exists nation;
CREATE FOREIGN TABLE NATION
(
    like tpch.nation
```

```
)
SERVER gsmpp_server
OPTIONS (
        encoding 'utf8',
        location 'obs://<obs_bucket_name>/tpch/nation.tbl',
        format 'text',
        delimiter '|',
        access_key '<Access_Key_Id>',
        secret_access_key '<Secret_Access_Key>',
        chunksize '64',
        IGNORE_EXTRA_DATA 'on'
);

DROP FOREIGN table if exists supplier;
CREATE FOREIGN TABLE SUPPLIER
(
      like tpch.supplier
)
SERVER gsmpp_server
OPTIONS (
      encoding 'utf8',
      location 'obs://<obs_bucket_name>/tpch/supplier.tbl',
      format 'text',
      delimiter '|',
      access_key '<Access_Key_Id>',
      secret_access_key '<Secret_Access_Key>',
      chunksize '64',
      IGNORE_EXTRA_DATA 'on'
);

DROP FOREIGN table if exists customer;
CREATE FOREIGN TABLE CUSTOMER
(
      like tpch.customer
)
SERVER gsmpp_server
OPTIONS (
      encoding 'utf8',
      location 'obs://<obs_bucket_name>/tpch/customer.tbl',
      format 'text',
      delimiter '|',
      access_key '<Access_Key_Id>',
      secret_access_key '<Secret_Access_Key>',
      chunksize '64',
      IGNORE_EXTRA_DATA 'on'
);
DROP FOREIGN table if exists part;
CREATE FOREIGN TABLE PART
(
      like tpch.part

)
SERVER gsmpp_server
OPTIONS (
      encoding 'utf8',
      location 'obs://<obs_bucket_name>/tpch/part.tbl',
      format 'text',
      delimiter '|',
      access_key '<Access_Key_Id>',
      secret_access_key '<Secret_Access_Key>',
      chunksize '64',
      IGNORE_EXTRA_DATA 'on'
);
DROP FOREIGN table if exists partsupp;
CREATE FOREIGN TABLE PARTSUPP
(
      like tpch.partsupp
)
SERVER gsmpp_server
```



```
OPTIONS (
        encoding 'utf8',
        location 'obs://<obs_bucket_name>/tpch/partsupp.tbl',
        format 'text',
        delimiter '|',
        access_key '<Access_Key_Id>',
        secret_access_key '<Secret_Access_Key>',
        chunksize '64',
        IGNORE_EXTRA_DATA 'on'
);
DROP FOREIGN table if exists orders;
CREATE FOREIGN TABLE ORDERS
(
        like tpch.orders
)
SERVER gsmpp_server
OPTIONS (
        encoding 'utf8',
        location 'obs://<obs_bucket_name>/tpch/orders.tbl',
        format 'text',
        delimiter '|',
        access_key '<Access_Key_Id>',
        secret_access_key '<Secret_Access_Key>',
        chunksize '64',
        IGNORE_EXTRA_DATA 'on'
);
DROP FOREIGN table if exists lineitem;
CREATE FOREIGN TABLE LINEITEM
(
        like tpch.lineitem
)
SERVER gsmpp_server
OPTIONS (
        encoding 'utf8',
        location 'obs://<obs_bucket_name>/tpch/lineitem.tbl',
        format 'text',
        delimiter '|',
        access_key '<Access_Key_Id>',
        secret_access_key '<Secret_Access_Key>',
        chunksize '64',
        IGNORE_EXTRA_DATA 'on'
);
```

步骤3 复制并执行以下语句，将外表数据导入到对应的数据库表中。

将OBS外表的数据通过insert命令导入GaussDB(DWS) 的数据库表中，数据库内核对应的操作为OBS数据高速并发导入GaussDB(DWS) 。

```
INSERT INTO tpch.lineitem SELECT * FROM tpchobs.lineitem;
INSERT INTO tpch.part SELECT * FROM tpchobs.part;
INSERT INTO tpch.partsupp SELECT * FROM tpchobs.partsupp;
INSERT INTO tpch.customer SELECT * FROM tpchobs.customer;
INSERT INTO tpch.supplier SELECT * FROM tpchobs.supplier;
INSERT INTO tpch.nation SELECT * FROM tpchobs.nation;
INSERT INTO tpch.region SELECT * FROM tpchobs.region;
INSERT INTO tpch.orders SELECT * FROM tpchobs.orders;
```

导入数据需要约10分钟，请耐心等待。

----结束

步骤二：多表分析与主题分析

以下以TPC-H标准查询为例，演示在GaussDB(DWS) 中进行的基本数据查询。

在进行数据查询之前，请先执行“Analyze”命令生成与数据库表相关的统计信息。统计信息存储在系统表PG_STATISTIC中，执行计划生成器会使用这些统计数据，以生成最有效的查询执行计划。

查询示例如下：

- **某地区供货商为公司带来的收入查询（TPCH-Q5）**

  通过执行TPCH-Q5查询语句，可以查询到通过某个地区零件供货商获得的收入（收入按sum(l_extendedprice * (1 - l_discount))计算）统计信息。该统计信息可用于决策在给定的区域是否需要建立一个当地分配中心。

  复制并执行以下TPCH-Q5语句进行查询。该语句的特点是：带有分组、排序、聚集操作并存的多表连接查询操作。

```
SET current_schema='tpch';
SELECT
n_name,
sum(l_extendedprice * (1 - l_discount)) as revenue
FROM
customer,
orders,
lineitem,
supplier,
nation,
region
where
c_custkey = o_custkey
and l_orderkey = o_orderkey
and l_suppkey = s_suppkey
and c_nationkey = s_nationkey
and s_nationkey = n_nationkey
and n_regionkey = r_regionkey
and r_name = 'ASIA'
and o_orderdate >= '1994-01-01'::date
and o_orderdate < '1994-01-01'::date + interval '1 year'
group by
n_name
order by
revenue desc;
```

- **零件/供货商关系查询（TPCH-Q16）**

  通过执行TPCH-Q16查询语句，可以获得能够以指定的贡献条件供应零件的供货商数量。该信息可用于决策在订单量大，任务紧急时，是否有充足的供货商。

  复制并执行以下TPCH-Q16语句进行查询，该语句的特点是：带有分组、排序、聚集、去重、NOT IN子查询操作并存的多表连接操作。

```
SET current_schema='tpch';
SELECT
p_brand,
p_type,
p_size,
count(distinct ps_suppkey) as supplier_cnt
FROM
partsupp,
part
where
p_partkey = ps_partkey
and p_brand <> 'Brand#45'
and p_type not like 'MEDIUM POLISHED%'
and p_size in (49, 14, 23, 45, 19, 3, 36, 9)
and ps_suppkey not in (
        select
        s_suppkey
        from
        supplier
        where
        s_comment like '%Customer%Complaints%'
)
group by
p_brand,
p_type,
```



```
p_size
order by
supplier_cnt desc,
p_brand,
p_type,
p_size
limit 100;
```

- **小订单收入损失查询（TPCH-Q17）**

  通过查询得知如果没有小量订单，平均年收入将损失多少。筛选出比平均供货量的20%还低的小批量订单，如果这些订单不再对外供货，由此计算平均一年的损失。

  复制并执行以下TPCH-Q17语句进行查询，该语句的特点是：带有聚集、聚集子查询操作并存的两表连接操作。

```
SET current_schema='tpch';
SELECT
sum(l_extendedprice) / 7.0 as avg_yearly
FROM
lineitem,
part
where
p_partkey = l_partkey
and p_brand = 'Brand#23'
and p_container = 'MED BOX'
and l_quantity < (
        select 0.2 * avg(l_quantity)
        from lineitem
        where l_partkey = p_partkey
);
```

4 零售业百货公司经营状况分析

零售业百货公司样例简介

本实践将演示以下场景：从OBS加载各个零售商场每日经营的业务数据到数据仓库对应的表中，然后对商铺营业额、客流信息、月度销售排行、月度客流转化率、月度租售比、销售坪效等KPI信息进行汇总和查询。本示例旨在展示在零售业场景中GaussDB(DWS) 数据仓库的多维度查询分析的能力。

说明

GaussDB(DWS) 已预先将样例数据上传到OBS桶的“retail-data”文件夹中，并给所有华为云用户赋予了该OBS桶的只读访问权限。

操作流程

本实践预计时长60分钟，基本流程如下：

1. **准备工作**
2. **步骤一：导入零售业百货公司样例数据**
3. **步骤二：经营状况分析**

支持区域

当前已上传OBS数据的区域如**表4-1**所示。

表 4-1 区域和 OBS 桶名

区域	OBS桶名
华北-北京一	dws-demo-cn-north-1
华北-北京二	dws-demo-cn-north-2
华北-北京四	dws-demo-cn-north-4
华北-乌兰察布一	dws-demo-cn-north-9
华东-上海一	dws-demo-cn-east-3
华东-上海二	dws-demo-cn-east-2

区域	OBS桶名
华南-广州	dws-demo-cn-south-1
华南-广州友好	dws-demo-cn-south-4
中国-香港	dws-demo-ap-southeast-1
亚太-新加坡	dws-demo-ap-southeast-3
亚太-曼谷	dws-demo-ap-southeast-2
拉美-圣地亚哥	dws-demo-la-south-2
非洲-约翰内斯堡	dws-demo-af-south-1
拉美-墨西哥城一	dws-demo-na-mexico-1
拉美-墨西哥城二	dws-demo-la-north-2
莫斯科二	dws-demo-ru-northwest-2
拉美-圣保罗一	dws-demo-sa-brazil-1

准备工作

- 已注册账号，账号不能处于欠费或冻结状态。
- 获取此账号的“AK/SK”。
- 已创建集群，并已使用Data Studio连接集群，参见**步骤一：创建集群**和**步骤二：使用Data Studio连接集群**。

步骤一：导入零售业百货公司样例数据

使用SQL客户端工具连接到集群后，就可以在SQL客户端工具中，执行以下步骤导入零售业百货公司样例数据并执行查询。

步骤1 执行以下语句，创建retail数据库。

```
CREATE DATABASE retail encoding 'utf8' template template0;
```

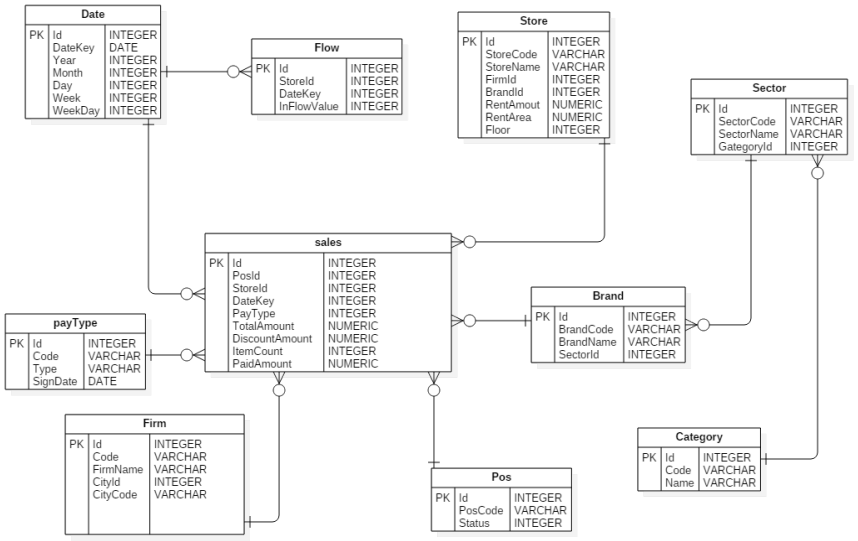
步骤2 执行以下步骤切换为连接新建的数据库。

1. 在Data Studio客户端的**“对象浏览器”**窗口，右键单击数据库连接名称，在弹出菜单中单击“刷新”，刷新后就可以看到新建的数据库。
2. 右键单击“retail”数据库名称，在弹出菜单中单击“打开连接”。
3. 右键单击“retail”数据库名称，在弹出菜单中单击“打开新的终端”，即可打开连接到指定数据库的SQL命令窗口，后面的步骤，请全部在该命令窗口中执行。

步骤3 创建数据库表。

样例数据包含10张数据库表，其关联关系如**图4-1**所示。

图 4-1 百货公司样例数据表



复制并执行以下语句，创建零售业百货公司信息数据库表。

```
CREATE SCHEMA retail_data;
SET current_schema='retail_data';

DROP TABLE IF EXISTS STORE;
CREATE TABLE STORE (
        ID INT,
        STORECODE VARCHAR(10),
        STORENAME VARCHAR(100),
        FIRMID INT,
        FLOOR INT,
        BRANDID INT,
        RENTAMOUNT NUMERIC(18,2),
        RENTAREA NUMERIC(18,2)
)
WITH (ORIENTATION = COLUMN, COMPRESSION=MIDDLE) DISTRIBUTE BY REPLICATION;

DROP TABLE IF EXISTS POS;
CREATE TABLE POS(
        ID INT,
        POSCODE VARCHAR(20),
        STATUS INT,
        MODIFICATIONDATE DATE
)
WITH (ORIENTATION = COLUMN, COMPRESSION=MIDDLE) DISTRIBUTE BY REPLICATION;

DROP TABLE IF EXISTS BRAND;
CREATE TABLE BRAND (
        ID INT,
        BRANDCODE VARCHAR(10),
        BRANDNAME VARCHAR(100),
        SECTORID INT
)
WITH (ORIENTATION = COLUMN, COMPRESSION=MIDDLE) DISTRIBUTE BY REPLICATION;

DROP TABLE IF EXISTS SECTOR;
CREATE TABLE SECTOR(
        ID INT,
        SECTORCODE VARCHAR(10),
        SECTORNAME VARCHAR(20),
        CATEGORYID INT
)
WITH (ORIENTATION = COLUMN, COMPRESSION=MIDDLE) DISTRIBUTE BY REPLICATION;
```

```
DROP TABLE IF EXISTS CATEGORY;
CREATE TABLE CATEGORY(
        ID INT,
        CODE VARCHAR(10),
        NAME VARCHAR(20)
)
WITH (ORIENTATION = COLUMN, COMPRESSION=MIDDLE) DISTRIBUTE BY REPLICATION;

DROP TABLE IF EXISTS FIRM;
CREATE TABLE FIRM(
        ID INT,
        CODE VARCHAR(4),
        NAME VARCHAR(40),
        CITYID INT,
        CITYNAME VARCHAR(10),
        CITYCODE VARCHAR(20)
)
WITH (ORIENTATION = COLUMN, COMPRESSION=MIDDLE) DISTRIBUTE BY REPLICATION;

DROP TABLE IF EXISTS DATE;
CREATE TABLE DATE(
        ID INT,
        DATEKEY DATE,
        YEAR INT,
        MONTH INT,
        DAY INT,
        WEEK INT,
        WEEKDAY INT
)
WITH (ORIENTATION = COLUMN, COMPRESSION=MIDDLE) DISTRIBUTE BY REPLICATION;

DROP TABLE IF EXISTS PAYTYPE;
CREATE TABLE PAYTYPE(
        ID INT,
        CODE VARCHAR(10),
        TYPE VARCHAR(10),
        SIGNDATE DATE
)
WITH (ORIENTATION = COLUMN, COMPRESSION=MIDDLE) DISTRIBUTE BY REPLICATION;

DROP TABLE IF EXISTS SALES;
CREATE TABLE SALES(
         ID INT,
         POSID INT,
         STOREID INT,
         DATEKEY INT,
         PAYTYPE INT,
         TOTALAMOUNT NUMERIC(18,2),
         DISCOUNTAMOUNT NUMERIC(18,2),
         ITEMCOUNT INT,
         PAIDAMOUNT NUMERIC(18,2)
)
WITH (ORIENTATION = COLUMN, COMPRESSION=MIDDLE) DISTRIBUTE BY HASH(ID);

DROP TABLE IF EXISTS FLOW;
CREATE TABLE FLOW (
         ID INT,
         STOREID INT,
         DATEKEY INT,
         INFLOWVALUE INT
)
WITH (ORIENTATION = COLUMN, COMPRESSION=MIDDLE) DISTRIBUTE BY HASH(ID);
```

步骤4 创建外表。外表用于识别和关联OBS上的源数据。

须知

- 其中，*<obs_bucket_name>*代表OBS桶名，仅支持部分区域，当前支持的区域和对应的OBS桶名请参见**支持区域**。GaussDB(DWS) 集群不支持跨区域访问OBS桶数据。
- 本实践以“华北-北京四”地区为例，可填入dws-demo-cn-north-4，<Access_Key_Id>和<Secret_Access_Key>替换为实际值，在**准备工作**获取。
- 认证用的AK和SK硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件或者环境变量中密文存放，使用时解密，确保安全。
- 创建外表如果提示“ERROR: schema "*xxx*" does not exist Position”，则说明schema不存在，请先参照上一步创建schema。

```
CREATE SCHEMA retail_obs_data;
SET current_schema='retail_obs_data';
DROP FOREIGN table if exists SALES_OBS;
CREATE FOREIGN TABLE SALES_OBS
(
    like retail_data.SALES
)
SERVER gsmpp_server
OPTIONS (
    encoding 'utf8',
    location 'obs://<obs_bucket_name>/retail-data/sales',
    format 'csv',
    delimiter ',',
    access_key '<Access_Key_Id>',
    secret_access_key '<Secret_Access_Key>',
    chunksize '64',
    IGNORE_EXTRA_DATA 'on',
    header 'on'
);

DROP FOREIGN table if exists FLOW_OBS;
CREATE FOREIGN TABLE FLOW_OBS
(
    like retail_data.flow
)
SERVER gsmpp_server
OPTIONS (
    encoding 'utf8',
    location 'obs://<obs_bucket_name>/retail-data/flow',
    format 'csv',
    delimiter ',',
    access_key '<Access_Key_Id>',
    secret_access_key '<Secret_Access_Key>',
    chunksize '64',
    IGNORE_EXTRA_DATA 'on',
    header 'on'
);

DROP FOREIGN table if exists BRAND_OBS;
CREATE FOREIGN TABLE BRAND_OBS
(
    like retail_data.brand
)
SERVER gsmpp_server
OPTIONS (
    encoding 'utf8',
    location 'obs://<obs_bucket_name>/retail-data/brand',
    format 'csv',
    delimiter ',',
    access_key '<Access_Key_Id>',
    secret_access_key '<Secret_Access_Key>',
```

```
        chunksize '64',
        IGNORE_EXTRA_DATA 'on',
        header 'on'
);


DROP FOREIGN table if exists CATEGORY_OBS;
CREATE FOREIGN TABLE CATEGORY_OBS
(
      like retail_data.category
)
SERVER gsmpp_server
OPTIONS (
        encoding 'utf8',
        location 'obs://<obs_bucket_name>/retail-data/category',
        format 'csv',
        delimiter ',',
        access_key '<Access_Key_Id>',
        secret_access_key '<Secret_Access_Key>',
        chunksize '64',
        IGNORE_EXTRA_DATA 'on',
        header 'on'
);

DROP FOREIGN table if exists DATE_OBS;
CREATE FOREIGN TABLE DATE_OBS
(
       like retail_data.date
)
SERVER gsmpp_server
OPTIONS (
        encoding 'utf8',
        location 'obs://<obs_bucket_name>/retail-data/date',
        format 'csv',
        delimiter ',',
        access_key '<Access_Key_Id>',
        secret_access_key '<Secret_Access_Key>',
        chunksize '64',
        IGNORE_EXTRA_DATA 'on',
        header 'on'
);

DROP FOREIGN table if exists FIRM_OBS;
CREATE FOREIGN TABLE FIRM_OBS
(
        like retail_data.firm
)
SERVER gsmpp_server
OPTIONS (
        encoding 'utf8',
        location 'obs://<obs_bucket_name>/retail-data/firm',
        format 'csv',
        delimiter ',',
        access_key '<Access_Key_Id>',
        secret_access_key '<Secret_Access_Key>',
        chunksize '64',
        IGNORE_EXTRA_DATA 'on',
        header 'on'
);


DROP FOREIGN table if exists PAYTYPE_OBS;
CREATE FOREIGN TABLE PAYTYPE_OBS
(
        like retail_data.paytype
)
SERVER gsmpp_server
OPTIONS (
        encoding 'utf8',
```



```
        location 'obs://<obs_bucket_name>/retail-data/paytype',
        format 'csv',
        delimiter ',',
        access_key '<Access_Key_Id>',
        secret_access_key '<Secret_Access_Key>',
        chunksize '64',
        IGNORE_EXTRA_DATA 'on',
        header 'on'
);


DROP FOREIGN table if exists POS_OBS;
CREATE FOREIGN TABLE POS_OBS
(
        like retail_data.pos
)
SERVER gsmpp_server
OPTIONS (
        encoding 'utf8',
        location 'obs://<obs_bucket_name>/retail-data/pos',
        format 'csv',
        delimiter ',',
        access_key '<Access_Key_Id>',
        secret_access_key '<Secret_Access_Key>',
        chunksize '64',
        IGNORE_EXTRA_DATA 'on',
        header 'on'
);

DROP FOREIGN table if exists SECTOR_OBS;
CREATE FOREIGN TABLE SECTOR_OBS
(
        like retail_data.sector
)
SERVER gsmpp_server
OPTIONS (
        encoding 'utf8',
        location 'obs://<obs_bucket_name>/retail-data/sector',
        format 'csv',
        delimiter ',',
        access_key '<Access_Key_Id>',
        secret_access_key '<Secret_Access_Key>',
        chunksize '64',
        IGNORE_EXTRA_DATA 'on',
        header 'on'
);


DROP FOREIGN table if exists STORE_OBS;
CREATE FOREIGN TABLE STORE_OBS
(
         like retail_data.store
)
SERVER gsmpp_server
OPTIONS (
         encoding 'utf8',
         location 'obs://<obs_bucket_name>/retail-data/store',
         format 'csv',
         delimiter ',',
         access_key '<Access_Key_Id>',
         secret_access_key '<Secret_Access_Key>',
         chunksize '64',
         IGNORE_EXTRA_DATA 'on',
         header 'on'
);
```

步骤5 复制并执行以下语句，导入外表数据到集群。

```
INSERT INTO retail_data.store SELECT * FROM retail_obs_data.STORE_OBS;
INSERT INTO retail_data.sector SELECT * FROM retail_obs_data.SECTOR_OBS;
```



```
INSERT INTO retail_data.paytype SELECT * FROM retail_obs_data.PAYTYPE_OBS;
INSERT INTO retail_data.firm SELECT * FROM retail_obs_data.FIRM_OBS;
INSERT INTO retail_data.flow SELECT * FROM retail_obs_data.FLOW_OBS;
INSERT INTO retail_data.category SELECT * FROM retail_obs_data.CATEGORY_OBS;
INSERT INTO retail_data.date SELECT * FROM retail_obs_data.DATE_OBS;
INSERT INTO retail_data.pos SELECT * FROM retail_obs_data.POS_OBS;
INSERT INTO retail_data.brand SELECT * FROM retail_obs_data.BRAND_OBS;
INSERT INTO retail_data.sales SELECT * FROM retail_obs_data.SALES_OBS;
```

导入数据需要一些时间，请耐心等待。

步骤6 复制并执行以下语句，创建视图v_sales_flow_details。

```
SET current_schema='retail_data';
CREATE VIEW v_sales_flow_details AS
SELECT
FIRM.ID FIRMID, FIRM.NAME FIRNAME, FIRM. CITYCODE,
CATEGORY.ID CATEGORYID, CATEGORY.NAME CATEGORYNAME,
SECTOR.ID SECTORID, SECTOR.SECTORNAME,
BRAND.ID BRANDID, BRAND.BRANDNAME,
STORE.ID STOREID, STORE.STORENAME, STORE.RENTAMOUNT, STORE.RENTAREA,
DATE.DATEKEY, SALES.TOTALAMOUNT, DISCOUNTAMOUNT, ITEMCOUNT, PAIDAMOUNT, INFLOWVALUE
FROM SALES
INNER JOIN STORE ON SALES.STOREID = STORE.ID
INNER JOIN FIRM ON STORE.FIRMID = FIRM.ID
INNER JOIN BRAND ON STORE.BRANDID = BRAND.ID
INNER JOIN SECTOR ON BRAND.SECTORID = SECTOR.ID
INNER JOIN CATEGORY ON SECTOR.CATEGORYID = CATEGORY.ID
INNER JOIN DATE ON SALES.DATEKEY = DATE.ID
INNER JOIN FLOW ON FLOW.DATEKEY = DATE.ID AND FLOW.STOREID = STORE.ID;
```

----结束

步骤二：经营状况分析

以下以零售百货公司标准查询为例，演示在GaussDB(DWS) 中进行的基本数据查询。

在进行数据查询之前，请先执行“Analyze”命令生成与数据库表相关的统计信息。统计信息存储在系统表PG_STATISTIC中，执行计划生成器会使用这些统计数据，以生成最有效的查询执行计划。

查询示例如下：

- **查询各商铺的月度营业额**

  复制并执行以下语句查询各商铺的月度营业额。

  ```
  SET current_schema='retail_data';
  SELECT DATE_TRUNC('month',datekey)
  AT TIME ZONE 'UTC' AS __timestamp,
  SUM(paidamount)
  AS sum__paidamount
  FROM v_sales_flow_details
  GROUP BY DATE_TRUNC('month',datekey) AT TIME ZONE 'UTC'
  ORDER BY SUM(paidamount) DESC;
  ```

- **查询各门店营收及租售比状况**

  复制并执行以下语句进行营收及租售比状况查询。

  ```
  SET current_schema='retail_data';
  SELECT firname AS firname,
  storename AS storename,
  SUM(paidamount)
  AS sum__paidamount,
  AVG(RENTAMOUNT)/SUM(PAIDAMOUNT)
  AS rentamount_sales_rate
  FROM v_sales_flow_details
  GROUP BY firname, storename
  ORDER BY SUM(paidamount) DESC;
  ```

- **各城市营业汇总分析**

  复制并执行以下语句进行汇总分析查询。

```
SET current_schema='retail_data';
SELECT citycode AS citycode,
SUM(paidamount)
AS sum__paidamount
FROM v_sales_flow_details
GROUP BY citycode
ORDER BY SUM(paidamount) DESC;
```

- **各门店租售比和客流转化率对比分析**

```
SET current_schema='retail_data';
SELECT brandname AS brandname,
firname AS firname,
SUM(PAIDAMOUNT)/AVG(RENTAREA) AS sales_rentarea_rate,
SUM(ITEMCOUNT)/SUM(INFLOWVALUE) AS poscount_flow_rate,
AVG(RENTAMOUNT)/SUM(PAIDAMOUNT) AS rentamount_sales_rate
FROM v_sales_flow_details
GROUP BY brandname,  firname
ORDER BY sales_rentarea_rate DESC;
```

- **品牌业态分析**

```
SET current_schema='retail_data';
SELECT categoryname AS categoryname,
brandname AS brandname,
SUM(paidamount) AS sum__paidamount
FROM v_sales_flow_details
GROUP BY categoryname,
brandname
ORDER BY sum__paidamount DESC;
```

- **查询各品牌每日营业状况**

```
SET current_schema='retail_data';
SELECT brandname AS brandname,
DATE_TRUNC('day', datekey) AT TIME ZONE 'UTC' AS __timestamp,
SUM(paidamount) AS sum__paidamount
FROM v_sales_flow_details
WHERE datekey >= '2016-01-01 00:00:00'
AND datekey <= '2016-01-30 00:00:00'
GROUP BY brandname,
DATE_TRUNC('day', datekey) AT TIME ZONE 'UTC'
ORDER BY sum__paidamount ASC
LIMIT 50000;
```


5 快速创建时序表

场景介绍

时序表继承普通表的行存和列存语法，降低了用户学习成本，易理解和使用；

时序表具备数据生命周期管理的能力，每天各种维度的数据爆炸式增长，需要定期给表增加新的分区，避免新数据无法存储。而对于很久之前的数据，其价值较低且不经常访问，可以定期删除无用的数据。因此时序表需要具备定时增加分区和定时删除分区的能力。

本实践主要讲解如何快速创建适合自己业务的时序表，并对时序表进行分区管理，从而真正发挥时序表的优势。将对应的列指定为合适的类型，能够帮助我们更好的提高导入、查询等场景的性能，让业务场景运行的更加高效。如下图所示，以发电机组数据采样为例：

图 5-1 发电机组数据采样示意图

图 5-2 存储数据表

tag					field				time
发电机	生产厂商	型号	位置	ID	电压	功率	频率	电流相角	timestamp
发电机组1	SX	V310	V1-5-C253S	9527	330	1680	60	20	2022-0315T00:00:00Z
发电机组2	SH	V350	V1-5-C451S	8975	321	1556	50	13	2022-0315T00:00:00Z
发电机组3	XJ	V420	V1-5-C650S	8571	339	1597	58	33	2022-0315T00:00:00Z
发电机组1	SX	V310	V1-5-C253S	9527	350	1730	75	40	2022-0315T00:10:00Z
发电机组2	SH	V350	V1-5-C451S	8975	450	1658	55	25	2022-0315T00:10:00Z
发电机组3	XJ	V420	V1-5-C650S	8571	377	1678	70	39	2022-0315T00:10:00Z
……	……	……	……	……	……	……	……	……	……
发电机组1	SX	V310	V1-5-C253S	9527	1020	3980	240	175	2022-0315T00:80:00Z
发电机组2	SH	V350	V1-5-C451S	8975	1340	4219	225	190	2022-0315T00:80:00Z
发电机组3	XJ	V420	V1-5-C650S	8571	1211	4387	320	155	2022-0315T00:80:00Z

- 对于不随时间变化而变化，描述发电机的属性信息的列（发电机信息、生产厂商、型号、位置、ID）被设置为tag列，在建表时需要将对应的列后面指定为TSTag；
- 对于采样数据的维度（电压、功率、频率、电流相角）这些对应的采样数值随时间的变化而变，将这些维度设置为field列，建表语句数据类型后面指定为TSField；
- 最后一列指定为时间列time，存储field列数据对应的时间信息，建表时将指定为TSTime。

基本流程

本实践预计时长：30分钟，基本流程如下：

1. **创建ECS**。
2. **创建IoT数仓**。
3. **使用gsql命令行客户端连接集群**。
4. **创建时序表**。

创建 ECS

参见**自定义购买弹性云服务器**购买。购买后，参见**登录Linux弹性云服务器**进行登录。

须知

创建ECS过程中，注意选择与后续的IoT数仓在同一个区域、可用区和同一个VPC子网下，ECS的操作系统选择与gsql客户端（本例以CentOS 7.6为例），并选择以密码方式登录。

创建 IoT 数仓

步骤1 登录华为云管理控制台。

步骤2 在“服务列表”中，选择“大数据 > 数据仓库服务”，单击右上角“创建数据仓库集群”。

步骤3 参见表5-1进行参数配置。

表 5-1 软件配置

参数名称	配置方式
区域	选择“华北-北京四”。 **说明** • 本指导以“华北-北京四”为例进行介绍，如果您需要选择其他区域进行操作，请确保所有操作均在同一区域进行。 • 请确保DWS跟ECS在同一个区域、可用区和同一个VPC子网下。
可用分区	可用区2
产品类型	IoT数仓
计算类型	弹性云服务器
存储类型	SSD云盘
CPU架构	X86
节点规格	dwsx2.rt.2xlarge.m6（8 vCPU | 64GB | 100~4,000 GB SSD） **说明** 如规格售罄，可选择其他可用区或规格。
热数据存储	200 GB / 节点
节点数量	3
集群名称	dws-demo01
管理员用户	dbadmin
管理员密码	用户自定义
确认密码	再次输入自定义的管理员密码
数据库端口	8000
虚拟私有云	vpc-default
子网	subnet-default(192.168.0.0/24) **须知** 请确保与ECS在同一个VPC。
安全组	自动创建安全组
公网访问	现在购买
企业项目	default
高级配置	默认配置

步骤4 信息核对无误，单击“立即购买”，单击“提交”。

步骤5 等待约10分钟，待集群创建成功后，单击集群名称进入“基本信息”，在“网络”区域，单击安全组名称，确认安全组规则已添加，以IP为192.168.0.x的客户端网段为例（本例gsql所在ECS的内网IP为192.168.0.90），需要添加192.168.0.0/24，端口为8000的安全组规则。

步骤6 返回到集群“基本信息”界面，记录下“内网IP”。

----结束

使用 gsql 命令行客户端连接集群

步骤1 使用root用户远程登录到需要安装gsql的Linux主机，然后在Linux命令窗口，执行以下命令下载gsql客户端：

```
wget https://obs.cn-north-1.myhuaweicloud.com/dws/download/dws_client_8.1.x_redhat_x64.zip --no-check-certificate
```

步骤2 执行以下命令解压客户端工具。

```
cd <客户端存放路径> unzip dws_client_8.1.x_redhat_x64.zip
```

其中：

- <客户端存放路径>：请替换为实际的客户端存放路径。
- dws_client_*8.1.x*_redhat_x64.zip：这是“RedHat x64”对应的客户端工具包名称，请替换为实际下载的包名。

步骤3 执行以下命令配置客户端。

```
source gsql_env.sh
```

提示以下信息表示客户端已配置成功。

```
All things done.
```

步骤4 执行以下命令，使用gsql客户端连接GaussDB(DWS)集群中的数据库，其中password为用户创建集群时自定义的密码。

```
gsql -d gaussdb -p 8000 -h 192.168.0.86 -U dbadmin -W password -r
```

显示如下信息表示gsql工具已经连接成功：

```
gaussdb=>
```

----结束

创建时序表

1. 以发电机组的场景作为示例，创建一张存储发电机组采样数据的时序表GENERATOR：

```
CREATE TABLE IF NOT EXISTS GENERATOR(
genset text TSTag,
manufacturer text TSTag,
model text TSTag,
location text TSTag,
ID bigint TSTag,
voltage numeric TSField,
power bigint TSField,
```



```
frequency numeric TSField,
angle numeric TSField,
time timestamptz TSTime) with (orientation=TIMESERIES, period='1 hour', ttl='1 month') distribute by
hash(model);
```

2. 查询当前时间：

```
select now();
              now
-------------------------------
2022-05-25 15:28:38.520757+08
(1 row)
```

3. 查询默认的分区与分区边界：

```
SELECT relname, boundaries FROM pg_partition where parentid=(SELECT oid FROM pg_class where
relname='generator') order by boundaries ;
   relname     |         boundaries
 ----------------+----------------------------
default_part_1 | {"2022-05-25 16:00:00+08"}
default_part_2 | {"2022-05-25 17:00:00+08"}
p1653505200    | {"2022-05-25 18:00:00+08"}
p1653541200    | {"2022-05-25 19:00:00+08"}
p1653577200    | {"2022-05-25 20:00:00+08"}
......
```

TSTag列支持类型：text, char, bool, int, big int。

TSTime列支持类型：timestamp with time zone, timestamp without time zone。在兼容Oracle语法的数据库中，也支持date类型。如果涉及到时区相关操作时，请选择带时区的时间类型。

TSField列支持的数据类型同列存表保持一致。

说明

- 写建表语句时，对于tag列的顺序，可以适当进行优化，将唯一性（distinct值）较高的列尽量写在前面，这样对于时序场景的性能有一些提升。
- 创建时序表时需要指定表级参数orientation属性设置为timeseries。
- 时序表不需要手动指定DISTRIBUTE BY和PARTITION BY, 默认按照所有tag列分布，且分区健默认为tstime指定的时间列。
- 对于**create table like**语法，需要自动从源表中继承列名和对应的kv_type类型。因此如果源表是非时序表，新表是时序表，对应的列的kv_type类型无法确定，则无法创建成功。
- 时序表必须指定一个时间属性（**TSTime**），且只能指定一个，且**TSTime**类型的列不能被删除。至少存在一个**TSTag**和**TSField**列，否则建表时会报错。

时序表以TSTIME列为分区键，具有自动分区管理功能。创建具有自动分区管理功能的分区表，帮助用户大大减少运维操作的时间。在上面的建表语句中，在表级参数项中可以看到，时序表指定了自动分区管理两个参数period和ttl。

- **period**：设置自动创建分区的间隔时间，默认值为1 day，取值范围：1 hour ~ 100 years。默认会为时序表创建自增分区任务。自增分区任务动态为我们创建分区，保证当前时刻有足够充裕的分区用于导入数据。
- **ttl**：设置自动淘汰分区的时间，取值范围：1 hour ~ 100 years。默认不创建淘汰分区任务，需要用户自己在建表手动指定，或者建表后通过ALTER TABLE语法设置。淘汰分区的策略是通过计算 nowtime - 分区boundary > ttl，满足该条件的分区将被drop掉。帮助用户定时清理过期的旧数据。

说明

分区边界的设置分为以下几种情况：

- period设置为“小时”，分区起始边界值为下个小时整点，分区的间隔为period的值。
- period设置为“天”，分区起始边界值为第二天零点，分区的间隔为period的值。
- period设置为“月”，分区起始边界值为下个月零点，分区的间隔为period的值。
- period设置为“年”，分区起始边界值为明年零点，分区的间隔为period的值。

创建时序表（手动设置分区边界）

1. 手动指定分区边界的起始值，例如手动设置默认的分区边界时间P1为“2022-05-30 16:32:45”、P2为“2022-05-31 16:56:12”，创建时序表GENERATOR1如下：

```
CREATE TABLE IF NOT EXISTS GENERATOR1(
genset text TSTag,
manufacturer text TSTag,
model text TSTag,
location text TSTag,
ID bigint TSTag,
voltage numeric TSField,
power bigint TSField,
frequency numeric TSField,
angle numeric TSField,
time timestamptz TSTime) with (orientation=TIMESERIES, period='1 day') distribute by hash(model)
partition by range(time)
(
PARTITION P1 VALUES LESS THAN('2022-05-30 16:32:45'),
PARTITION P2 VALUES LESS THAN('2022-05-31 16:56:12')
);
```

2. 查询当前时间：

```
select now();
              now
-------------------------------
2022-05-31 20:36:09.700096+08(1 row)
```

3. 查询分区以及分区边界：

```
SELECT relname, boundaries FROM pg_partition where parentid=(SELECT oid FROM pg_class where
relname='generator1') order by boundaries ;
   relname   |          boundaries
-------------+------------------------------
p1           | {"2022-05-30 16:32:45+08"}
p2           | {"2022-05-31 16:56:12+08"}
p1654073772 | {"2022-06-01 16:56:12+08"}
p1654160172 | {"2022-06-02 16:56:12+08"}
......
```

6 冷热数据管理优秀实践

场景介绍

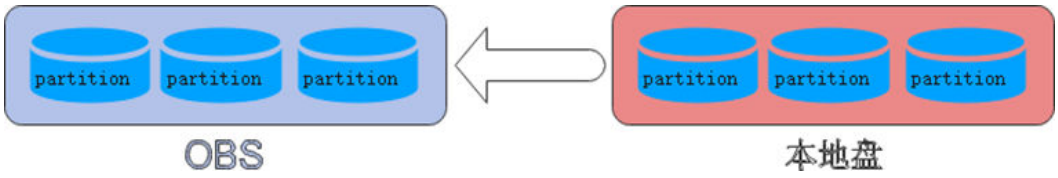
海量大数据场景下，随着业务和数据量的不断增长，数据存储与消耗的资源也日益增长。根据业务系统中用户对不同时期数据的不同使用需求，对膨胀的数据进行“冷热”分级管理，不仅可以提高数据分析性能还能降低业务成本。针对数据使用的一些场景，可以将数据按照时间分为：热数据、冷数据。

冷热数据主要从数据访问频率、更新频率进行划分。

- Hot（热数据）：访问、更新频率较高，对访问的响应时间要求很高的数据。
- Cold（冷数据）：不允许更新或更新访问频率较低，对访问的响应时间要求不高的数据。

用户可以定义冷热管理表，将符合规则的冷数据切换至OBS上进行存储，可以按照分区自动进行冷热数据的判断和迁移。

图 6-1 冷热数据管理



冷热切换的策略名称支持LMT（last modify time）和HPN（hot partition number），LMT指按分区的最后更新时间切换，HPN指保留热分区的个数切换。

- LMT：表示切换[day]时间前修改的热分区数据为冷分区，将该数据迁至OBS表空间中。其中[day]为整型，范围[0, 36500]，单位为天。
- HPN：表示保留HPN个有数据的分区为热分区。在冷热切换时，需要将数据迁移至OBS表空间中。其中HPN为整型，范围为[0,1600]。

约束限制

- 对于同时存在冷热分区的表，查询时会变慢，因为冷数据存储在OBS上，读写速度和时延都比在本地查询要慢。
- 目前冷热表只支持列存2.0版本的分区表，外表不支持冷热分区。

- 仅支持从热数据切换为冷数据，不支持从冷数据切换为热数据。

基本流程

本实践预计时长：30分钟，基本流程如下：

1. **创建集群**。
2. **使用gsql命令行客户端连接集群**。
3. **创建冷热表**。
4. **冷热数据切换**。
5. **查看冷热表数据分布**。

创建集群

步骤1 登录华为云管理控制台。

步骤2 在“服务列表”中，选择“大数据 > 数据仓库服务”，单击右上角“创建数据仓库集群”。

步骤3 参见**表6-1**进行参数配置。

表 6-1 软件配置

参数名称	配置方式
区域	选择“华北-北京四”。 **说明** 本指导以“华北-北京四”为例进行介绍，如果您需要选择其他区域进行操作，请确保所有操作均在同一区域进行。
可用区	可用区2
产品类型	标准数仓
CPU架构	X86
节点规格	dws2.m6.4xlarge.8（16 vCPU | 128GB | 2000GB SSD） **说明** 如规格售罄，可选择其他可用区或规格。
节点数量	3
集群名称	dws-demo
管理员用户	dbadmin
管理员密码	-
确认密码	-
数据库端口	8000
虚拟私有云	vpc-default
子网	subnet-default(192.168.0.0/24)

参数名称	配置方式
安全组	自动创建安全组
公网访问	现在购买
宽带	1Mbit/s
高级配置	默认配置

步骤4 信息核对无误，单击“立即购买”，单击“提交”。

步骤5 等待约6分钟，待集群创建成功后，单击集群名称前面的，弹出集群信息，记录下“公网访问地址”，例如dws-demov.dws.huaweicloud.com。

----结束

使用 gsql 命令行客户端连接集群

步骤1 使用root用户远程登录到需要安装gsql的Linux主机，然后在Linux命令窗口，执行以下命令下载gsql客户端：

```
wget https://obs.cn-north-1.myhuaweicloud.com/dws/download/dws_client_8.1.x_redhat_x64.zip --no-check-certificate
```

步骤2 执行以下命令解压客户端工具。

```
cd <客户端存放路径> unzip dws_client_8.1.x_redhat_x64.zip
```

其中：

- <客户端存放路径>：请替换为实际的客户端存放路径。
- dws_client_*8.1.x*_redhat_x64.zip：这是“RedHat x64”对应的客户端工具包名称，请替换为实际下载的包名。

步骤3 执行以下命令配置客户端。

```
source gsql_env.sh
```

提示以下信息表示客户端已配置成功。

```
All things done.
```

步骤4 执行以下命令，使用gsql客户端连接GaussDB(DWS)集群中的数据库，其中password为用户创建集群时自定义的密码。


```
gsql -d gaussdb -p 8000 -h 192.168.0.86 -U dbadmin -W password -r
```

显示如下信息表示gsql工具已经连接成功：

```
gaussdb=>
```

----结束

创建冷热表

创建列存冷热数据管理表lifecycle_table，指定热数据有效期LMT为100天。

```
CREATE TABLE lifecycle_table(i int, val text) WITH (ORIENTATION = COLUMN, storage_policy = 'LMT:100')
PARTITION BY RANGE (i)
(
PARTITION P1 VALUES LESS THAN(5),
PARTITION P2 VALUES LESS THAN(10),
PARTITION P3 VALUES LESS THAN(15),
PARTITION P8 VALUES LESS THAN(MAXVALUE)
)
ENABLE ROW MOVEMENT;
```

冷热数据切换

切换冷数据至OBS表空间。

- 自动切换：每日0点调度框架自动触发，无需关注切换情况。

  可使用函数pg_obs_cold_refresh_time(table_name, time)自定义自动切换时间。例如，根据业务情况调整自动触发时间为每天早晨6点30分。

  ```
  SELECT * FROM pg_obs_cold_refresh_time('lifecycle_table', '06:30:00');
  pg_obs_cold_refresh_time
  --------------------------
   SUCCESS
  (1 row)
  ```

- 手动切换。

  使用ALTER TABLE语句手动切换单表：

  ```
  ALTER TABLE lifecycle_table refresh storage;
  ALTER TABLE
  ```

  使用函数pg_refresh_storage()批量切换所有冷热表：

  ```
  SELECT pg_catalog.pg_refresh_storage();
   pg_refresh_storage
  --------------------
   (1,0)
  (1 row)
  ```

查看冷热表数据分布

- 查看单表数据分布情况。

  ```
  SELECT * FROM pg_catalog.pg_lifecycle_table_data_distribute('lifecycle_table');
  schemaname |     tablename     |   nodename    | hotpartition | coldpartition | switchablepartition |
  hotdatasize | colddatasize | switchabledatasize
  ------------+-------------------+---------------+--------------+---------------+---------------------+-------------
  +--------------+--------------------
   public     | lifecycle_table | dn_6001_6002 | p1,p2,p3,p8  |               |                     | 96 KB       | 0
  bytes      | 0 bytes
   public     | lifecycle_table | dn_6003_6004 | p1,p2,p3,p8  |               |                     | 96 KB       | 0
  bytes      | 0 bytes
   public     | lifecycle_table | dn_6005_6006 | p1,p2,p3,p8  |               |                     | 96 KB       | 0
  bytes      | 0 bytes
  (3 rows)
  ```

- 查看所有冷热表数据分布情况。

```
SELECT * FROM pg_catalog.pg_lifecycle_node_data_distribute();
schemaname |     tablename     |    nodename    | hotpartition | coldpartition | switchablepartition |
hotdatasize | colddatasize | switchabledatasize
------------+-------------------+----------------+--------------+---------------+---------------------+-------------
+--------------+--------------------
 public     | lifecycle_table | dn_6001_6002 | p1,p2,p3,p8  |               |                     |       98304 |
0 |            0
 public     | lifecycle_table | dn_6003_6004 | p1,p2,p3,p8  |               |                     |       98304 |
0 |            0
 public     | lifecycle_table | dn_6005_6006 | p1,p2,p3,p8  |               |                     |       98304 |
0 |            0
(3 rows)
```

7 分区自动管理优秀实践

场景介绍

对于分区列为时间的分区表，分区自动管理功能可以自动创建新分区和删除过期分区，降低分区表的维护成本，改善查询性能。为了便于查询和维护数据，用户通常使用分区列为时间的分区表来存储时间相关的数据，例如电商的订单信息、 物联网采集的实时数据。这些时间相关的数据导入分区表时，需要保证分区表要有对应时间的分区，由于普通的分区表不会自动创建新的分区和删除过期的分区，所以维护人员需要定期创建新分区和删除过期分区，提高了运维成本。

为解决上述问题，GaussDB(DWS) 引入了分区自动管理特性。可通过设置表级参数period、ttl开启分区自动管理功能，使分区表可以自动创建新分区和删除过期分区，降低分区表的维护成本，改善查询性能。

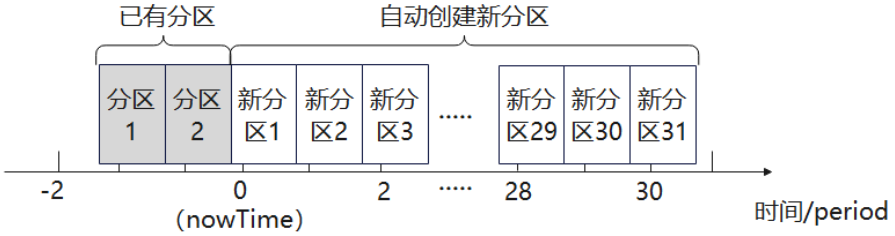
period：设置自动创建分区的间隔时间，默认值为1 day，取值范围：1 hour ~ 100 years。

ttl：设置自动淘汰分区的时间，取值范围：1 hour ~ 100 years。淘汰分区的策略是通过计算 nowtime - 分区boundary > ttl，满足该条件的分区将被清理掉。

- 自动创建新分区

  分区自动管理每隔period的时间就会自动创建分区，每次创建一个或多个时间范围为period的新分区，以推进最大的分区边界时间，保证其大于nowTime +30*period。由于每次创建分区时，都动态地为未来时间创建了预留分区，所以只要有一次自动创建新分区成功，就可以保证在未来30个period的时间之内，都不会出现实时数据因为没有对应分区而导入失败的情况。

  图 7-1 自动创建分区示意图



- 自动删除过期分区

边界时间早于nowTime-ttl的分区被认为是过期分区。分区自动管理每隔period的时间就会遍历检测所有分区，并删除其中的过期分区，如果所有的分区都是过期分区，则保留一个分区，并TRUNCATE该表。

约束限制

在使用分区管理功能时，需要满足如下约束：

- 不支持在小型机、加速集群、单机集群上使用。
- 支持在8.1.3及以上集群版本中使用。
- 仅支持行存范围分区表、列存范围分区表、时序表以及冷热表。
- 分区键唯一且类型仅支持timestamp、timestamptz、date类型。
- 不支持存在maxvalue分区。
- (nowTime - boundaryTime) / period需要小于分区个数上限，其中nowTime为当前时间，boundaryTime为现有分区中最早的分区边界时间。
- period、ttl取值范围为1hour ~ 100years。另外，在兼容Teradata或MySQL的数据库中，分区键类型为date时，period不能小于1day。
- 表级参数ttl不支持单独存在，必须要提前或同时设置period，并且要大于或等于period。
- 集群在线扩容期间，自动增加分区会失败，但是由于每次增分区时，都预留了足够的分区，所以不影响使用。

创建 ECS

参见**自定义购买弹性云服务器**购买。购买后，参见**登录Linux弹性云服务器**进行登录。

须知

创建ECS过程中，注意选择与后续的IoT数仓在同一个区域、可用区和同一个VPC子网下，ECS的操作系统选择与gsql客户端（本例以CentOS 7.6为例），并选择以密码方式登录。

创建集群

步骤1 登录华为云管理控制台。

步骤2 在“服务列表”中，选择“大数据 > 数据仓库服务”，单击右上角“创建数据仓库集群”。

步骤3 参见**表7-1**进行参数配置。

表 7-1 软件配置

参数名称	配置方式
区域	选择“华北-北京四”。 **说明** 本指导以“华北-北京四”为例进行介绍，如果您需要选择其他区域进行操作，请确保所有操作均在同一区域进行。

参数名称	配置方式
可用区	可用区2
产品类型	标准数仓
CPU架构	X86
节点规格	dws2.m6.4xlarge.8（16 vCPU | 128GB | 2000GB SSD） **说明** 如规格售罄，可选择其他可用区或规格。
节点数量	3
集群名称	dws-demo
管理员用户	dbadmin
管理员密码	-
确认密码	-
数据库端口	8000
虚拟私有云	vpc-default
子网	subnet-default(192.168.0.0/24)
安全组	自动创建安全组
公网访问	现在购买
宽带	1Mbit/s
高级配置	默认配置

步骤4 信息核对无误，单击“立即购买”，单击“提交”。

步骤5 等待约6分钟，待集群创建成功后，单击集群名称前面的，弹出集群信息，记录下“公网访问地址”，例如dws-demov.dws.huaweicloud.com。

----结束

使用 gsql 命令行客户端连接集群

步骤1 使用root用户远程登录到需要安装gsql的Linux主机，然后在Linux命令窗口，执行以下命令下载gsql客户端：

```
wget https://obs.cn-north-1.myhuaweicloud.com/dws/download/dws_client_8.1.x_redhat_x64.zip --no-check-certificate
```

步骤2 执行以下命令解压客户端工具。

```
cd <客户端存放路径> unzip dws_client_8.1.x_redhat_x64.zip
```

其中：

- <客户端存放路径>：请替换为实际的客户端存放路径。
- dws_client_*8.1.x*_redhat_x64.zip：这是“RedHat x64”对应的客户端工具包名称，请替换为实际下载的包名。

步骤3 执行以下命令配置客户端。

```
source gsql_env.sh
```

提示以下信息表示客户端已配置成功。

```
All things done.
```

步骤4 执行以下命令，使用gsql客户端连接GaussDB(DWS)集群中的数据库，其中password为用户创建集群时自定义的密码。

```
gsql -d gaussdb -p 8000 -h 192.168.0.86 -U dbadmin -W password -r
```

显示如下信息表示gsql工具已经连接成功：

```
gaussdb=>
```

----结束

分区自动管理

分区管理功能是和表级参数period、ttl绑定的，只要成功设置了表级参数period，即开启了自动创建新分区功能；成功设置了表级参数ttl，即开启了自动删除过期分区功能。第一次自动创建分区或删除分区的时间为设置period或ttl后30秒。

有两种开启分区管理功能的方式，具体如下：

- 建表时指定period、ttl。

  该方式适用于新建分区管理表时使用。新建分区管理表有两种语法，一种是建表时指定分区，另一种是建表时不指定分区。

  建分区管理表时如果指定分区，则语法规则和建普通分区表相同，唯一的区别就是会指定表级参数period、ttl。

  示例：创建分区管理表CPU1，指定分区。

  ```
  CREATE TABLE CPU1(
      id integer,
      IP text,
      time timestamp
  ) with (TTL='7 days',PERIOD='1 day')
  partition by range(time)
  (
      PARTITION P1 VALUES LESS THAN('2023-02-13 16:32:45'),
      PARTITION P2 VALUES LESS THAN('2023-02-15 16:48:12')
  );
  ```

  建分区管理表时可以只指定分区键不指定分区，此时将创建两个默认分区，这两个默认分区的分区时间范围均为period。其中，第一个默认分区的边界时间是大

于当前时间的第一个整时/整天/整周/整月/整年的时间，具体选择哪种整点时间取决于period的最大单位；第二个默认分区的边界时间是第一个分区边界时间加period。假设当前时间是2023-02-17 16:32:45，各种情况的第一个默认分区的分区边界选择如下表：

表 7-2 period 参数说明

period	period最大单位	第一个默认分区的分区边界
1hour	Hour	2023-02-17 17:00:00
1day	Day	2023-02-18 00:00:00
1month	Month	2023-03-01 00:00:00
13months	Year	2024-01-01 00:00:00

创建分区管理表CPU2，不指定分区：

```
CREATE TABLE CPU2(
    id integer,
    IP text,
    time timestamp
) with (TTL='7 days',PERIOD='1 day')
partition by range(time);
```

- 使用ALTER TABLE RESET的方式设置period、ttl。

  该方式适用于给一张满足分区管理约束的普通分区表增加分区管理功能。

  - 创建普通分区表CPU3：

    ```
    CREATE TABLE CPU3(
        id integer,
        IP text,
        time timestamp
    )
    partition by range(time)
    (
        PARTITION P1 VALUES LESS THAN('2023-02-14 16:32:45'),
        PARTITION P2 VALUES LESS THAN('2023-02-15 16:56:12')
    );
    ```

  - 同时开启自动创建和自动删除分区功能：

    ```
    ALTER TABLE CPU3 SET (PERIOD='1 day',TTL='7 days');
    ```

  - 只开启自动创建分区功能：

    ```
    ALTER TABLE CPU3 SET (PERIOD='1 day');
    ```

  - 只开启自动删除分区功能，如果没有提前开启自动创建分区功能，则开启失败：

    ```
    ALTER TABLE CPU3 SET (TTL='7 days');
    ```

  - 通过修改period和ttl修改分区管理功能：

    ```
    ALTER TABLE CPU3 SET (TTL='10 days',PERIOD='2 days');
    ```

- 关闭分区管理功能。

  使用ALTER TABLE RESET语句可以删除表级参数period、ttl，进而关闭相应的分区管理功能。

  说明

  - 不能在存在ttl的情况下，单独删除period。
  - 时序表不支持ALTER TABLE RESET。

- 同时关闭自动创建和自动删除分区功能：
  ```
  ALTER TABLE CPU1 RESET (PERIOD,TTL);
  ```
- 只关闭自动删除分区功能：
  ```
  ALTER TABLE CPU3 RESET (TTL);
  ```
- 只关闭自动创建分区功能，如果该表有ttl参数，则关闭失败：
  ```
  ALTER TABLE CPU3 RESET (PERIOD);
  ```

8 使用 CDM 将 MySQL 数据迁移至 GaussDB(DWS)集群

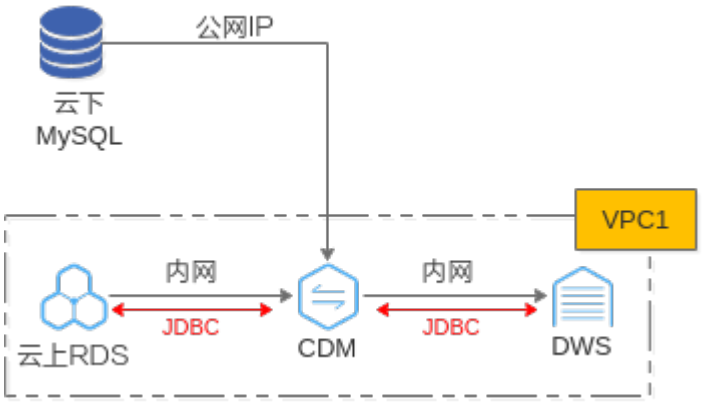
本入门提供通过云数据迁移服务CDM将MySQL数据批量迁移到GaussDB(DWS)集群的指导。

本入门的基本内容如下所示：

1. **迁移前数据检查**
2. **创建GaussDB(DWS)集群**
3. **创建CDM集群**
4. **创建连接**
5. **新建作业和迁移**
6. **迁移后数据一致性验证**

场景描述

图 8-1 迁移场景



主要包括云上和云下的MySQL数据迁移，支持整库迁移或者单表迁移，本文以**云下MySQL的整库迁移**为例。

- 云下MySQL数据迁移：

CDM通过公网IP访问MySQL数据库，CDM与GaussDB(DWS)在同一个VPC下，CDM分别与MySQL和DWS建立JDBC连接。

- 云上RDS-MySQL数据迁移：

  RDS、CDM和GaussDB(DWS)均在同一个VPC下，CDM分别与MySQL和DWS建立JDBC连接。如果云上RDS与DWS不在一个VPC，则CDM通过弹性公网IP访问RDS。

迁移前数据检查

步骤1 连接MySQL实例，查看MySQL数据库情况。

mysql -h *<host>***-P** *<port>***-u** *<userName>***-p--ssl-ca=***<caDIR>*

参数	说明
<host>	MySQL数据库连接地址。
<port>	数据库端口，默认3306。
<userName>	MySQL管理员账号，默认为root。
<caDIR>	CA证书路径，该文件需放在执行该命令的路径下。

出现如下提示时，输入数据库账号对应的密码：

```
Enter password:
```

步骤2 分析需要迁移的数据库名及编码、待迁移的表名、表属性。

例如，查询出待迁移的MySQL目标库为test01、test02以及数据库编码。其中test01库里包括 orders、persons、persons_b三张表和一张视图persons_beijing，test02库里包括一张表persons_c。

1. 查询数据库名。

   ```
   show databases;
   ```

   图 8-2 查询数据库

   ```
   mysql> show databases;
   +--------------------+
   | Database           |
   +--------------------+
   | information_schema |
   | mysql              |
   | performance_schema |
   | sys                |
   | test01             |
   | test02             |
   +--------------------+
   6 rows in set (0.00 sec)
   ```

2. 查询数据库编码。

   ```
   use <databasename>;
   status;
   ```

图 8-3 查询数据库编码（1）

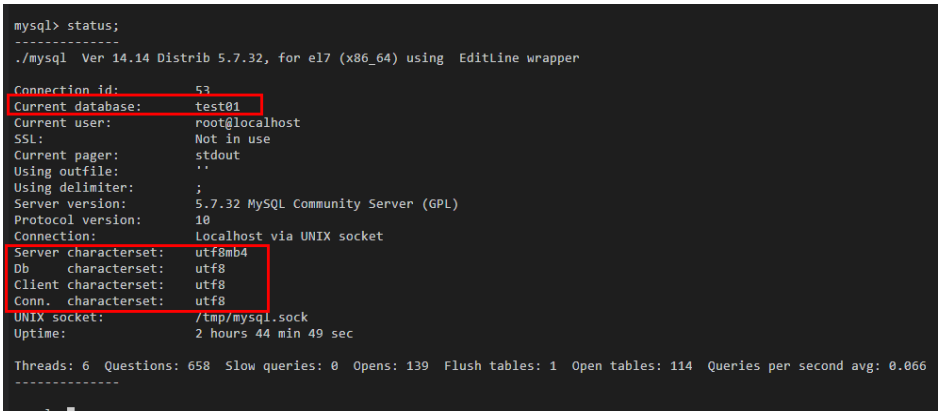
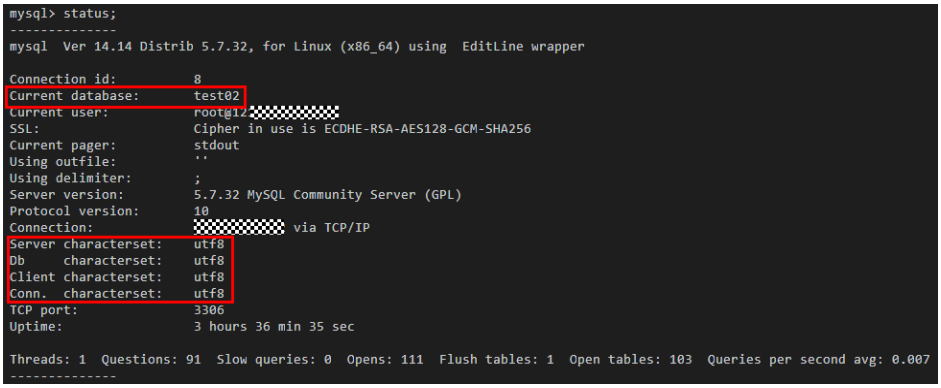


图 8-4 查询数据库编码（2）



3. 查询库表。

```
use <databasename>;
show full tables;
```

须知

- 由于GaussDB(DWS)数据库对表名大小写不敏感，如果原MySQL数据库中存在大小写混用的表名或者纯大写的表名，例如Table01、TABLE01，需要先修改表名为纯小写后才支持迁移，否则会导致迁移后，GaussDB(DWS)无法识别该表。
- 建议将MySQL也设置成大小写不敏感模式，修改方法：修改/etc/my.cnf参数lower_case_table_names=1，并重启MySQL服务。

图 8-5 查询库表

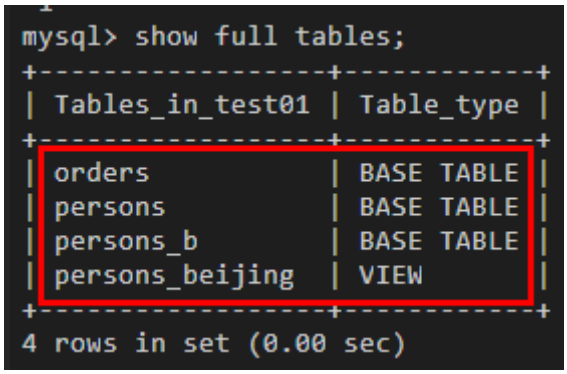
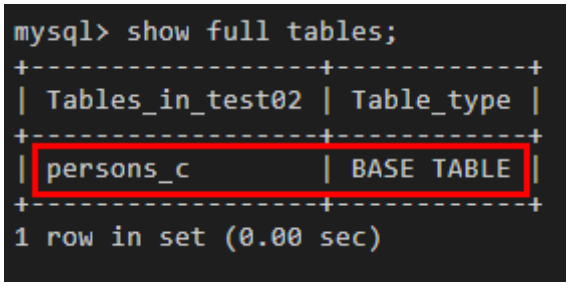


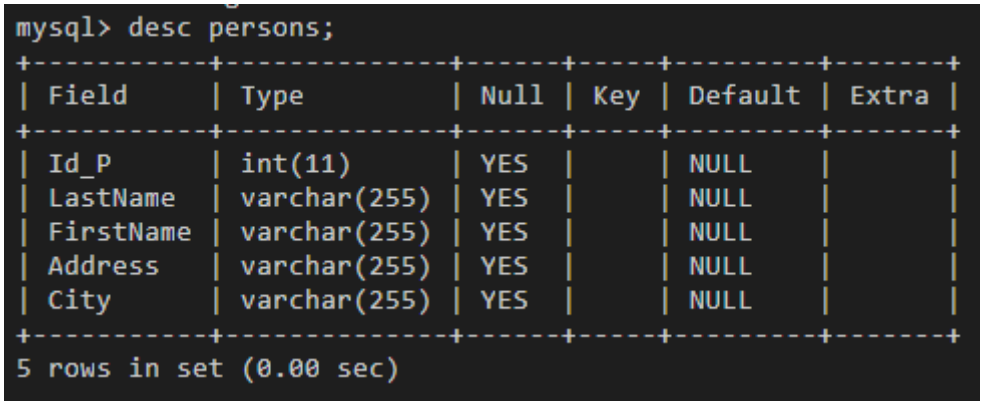
图 8-6 查询库表



4. 查看各个表的属性，以备迁移后对比。

```
use <databasename>;
desc <table name>;
```

图 8-7 查看表属性



----结束

创建 GaussDB(DWS)集群

步骤1 参见**创建集群**进行创建，区域可选择“华北-北京四”。

说明

确保GaussDB(DWS)集群与CDM集群在同一区域，同一个VPC下。

步骤2 参见**使用gsql客户端连接集群**连接到集群。

步骤3 创建**迁移前数据检查**的目标数据库test01和test02，确保与原MySQL的数据库同名，数据库编码一致。

```
create database test01 with encoding 'UTF-8' dbcompatibility 'mysql' template template0;
create database test02 with encoding 'UTF-8' dbcompatibility 'mysql' template template0;
```

----结束

创建 CDM 集群

步骤1 登录华为云控制台。

步骤2 选择“迁移 > 云数据迁移 CDM”进入CDM管理控制台。

步骤3 单击“购买云数据迁移服务”，按以下参数填写。

表 8-1 CDM 集群参数

参数名	取值
当前区域	华北-北京四（**与DWS选择在同一区域**）
可用区	可用区1（如果资源售罄则选其他可用区）
名称	CDM-demo
实例类型	cdm.large（如售罄请选择其他规格）
虚拟私有云	demo-vpc（**与DWS选择在同一个VPC**）
子网	subnet-f377(10.1.0.0/24)（示例）
安全组	-
企业项目	default

步骤4 单击“立即购买”，核对参数无误，单击“提交”。

步骤5 回到CDM管理控制台的“集群管理”页面，等待约5分钟，集群创建成功后，单击集群操作列的“绑定弹性IP”。

步骤6 勾选可用的弹性IP，单击“确认”。如果没有弹性IP，需要跳转到弹性IP界面，购买弹性IP。

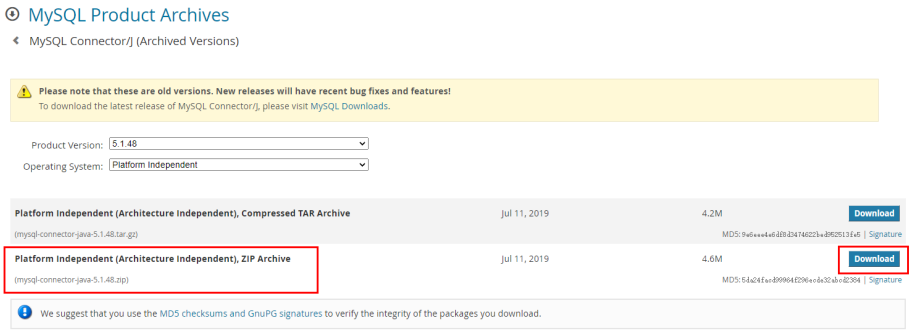
----结束

创建连接

步骤1 初次创建MySQL连接，需要上传驱动。

1. 访问**MySQL驱动**，选择“5.1.48”版本下载。

图 8-8 下载驱动



2. 下载到本地，解压后，获取mysql-connector-java-*xxx*.jar。
3. 回到CDM管理控制台的“集群管理”页面，单击集群操作列的“作业管理”，选择“连接管理 > 驱动管理”。
4. 单击“MySQL”右侧的“上传”，选择mysql-connector-java-*xxx*.jar，单击“上传文件”。

步骤2 创建MySQL连接。

1. 在CDM管理控制台的“集群管理”页面，单击集群操作列的“作业管理”，选择“连接管理 > 新建连接”。
2. 连接器类型勾选“MySQL”，单击“下一步”。（如果是云上RDS，则勾选“云数据库 MySQL”。）
3. 按**表8-2**填写连接信息，填写后单击“测试”，测试成功后，单击“保存”。

说明

如测试不通过，请确认CDM是否以公网IP方式连接MySQL数据库，如果是公网IP方式，请参见**步骤5**绑定公网IP。

表 8-2 MySQL 连接信息

参数项	取值
名称	MySQL
数据库服务器	192.168.1.100（示例，请填写云下MySQL实际的公网IP，要确保MySQL服务器已放开白名单访问）
端口	3306
数据库名称	test01
用户名	root
密码	root用户密码
使用本地API	否
使用Agent	否

步骤3 创建DWS连接。

1. 在CDM管理控制台的“集群管理”页面，单击集群操作列的“作业管理”，选择“连接管理 > 新建连接”。

2. 连接器类型勾选“数据仓库服务 (DWS)”，单击“下一步”。
3. 按表8-3填写连接信息，填写后单击“测试”，测试成功后，单击“保存”。

表 8-3 DWS 连接信息

参数项	取值
名称	DWS-test01
数据库服务器	单击“选择”，从集群列表中选择要连接的DWS集群。 **说明** 系统会自动刷出同一个区域、同一个VPC下的DWS集群，如果没有，则需要手动填写网络已连通的DWS的访问IP。
端口	8000
数据库名称	test01（参见步骤3确保GaussDB(DWS)已手动创建了对应的数据库）
用户名	dbadmin
密码	dbadmin用户密码
使用Agent	否

4. 重复3.1~3.3，创建DWS-test02连接。

----结束

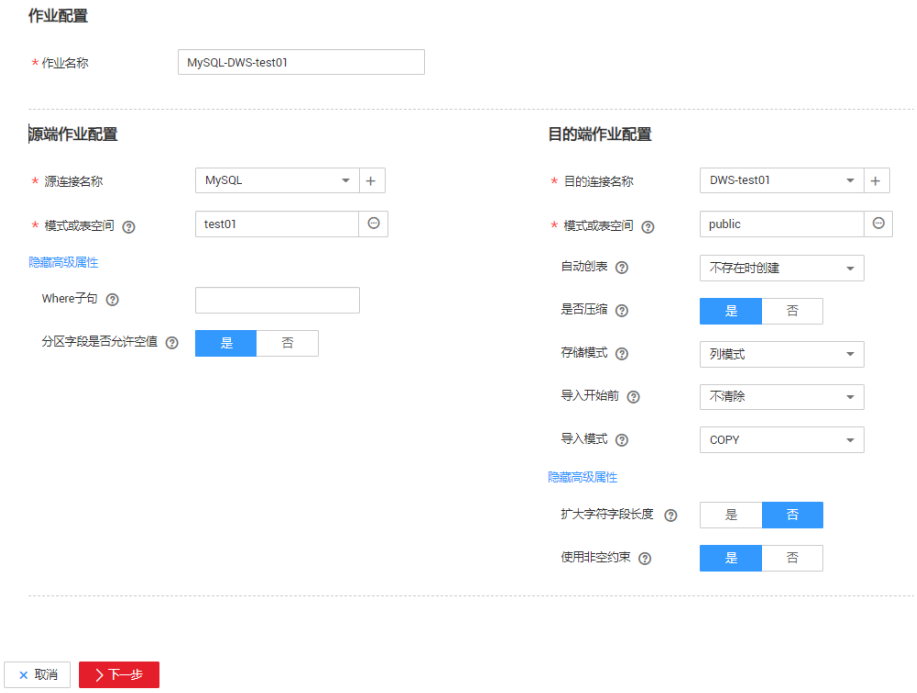
新建作业和迁移

步骤1 在CDM管理控制台的“集群管理”页面，单击集群操作列的“作业管理”，选择“整库迁移 > 新建作业”。

步骤2 填写如下参数后，单击“下一步”。

- 作业名称：MySQL-DWS-test01
- 源端作业配置：
  - 源连接名称：MySQL
- 目的端作业配置：
  - 目的连接名称：DWS-test01
  - 自动创表：不存在时创建
  - 是否压缩：是
  - 存储模式：列模式
  - 其他选项：保持默认

图 8-9 作业配置



步骤3 勾选所有表，单击  ，单击“下一步”。

步骤4 参数保持默认即可，单击“保存并运行”。

步骤5 查看作业运行情况，状态为“Succeeded”，表示迁移成功。

图 8-10 查看作业运行情况

步骤6 重复执行**步骤1**~**步骤5**，迁移数据库test02的所有表。

须知

在新建作业时，目标源的DWS库，需选择对应到test02。

----结束

迁移后数据一致性验证

步骤1 使用gsql连接DWS的test01集群。

gsql -d *test01* **-h** *数据库主机IP* **-p 8000 -U dbadmin -W** *数据库用户密码* **-r;**

步骤2 查询test01库的表。

```
select * from  pg_tables where schemaname= 'public';
```

图 8-11 查询 test01 库的表

步骤3 查询每个表的数据是否齐全，字段是否完整。

```
select count(*) from table name;
\d+ table name;
```

图 8-12 查询表字段

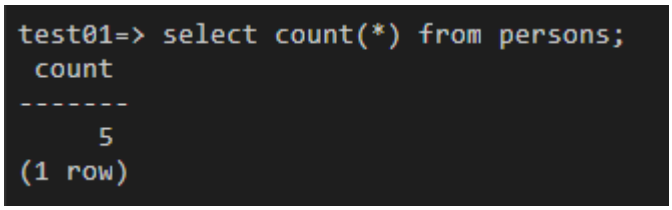
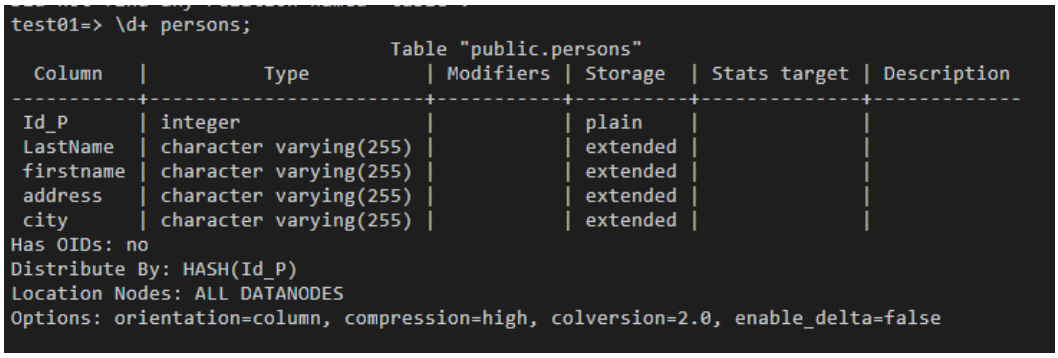


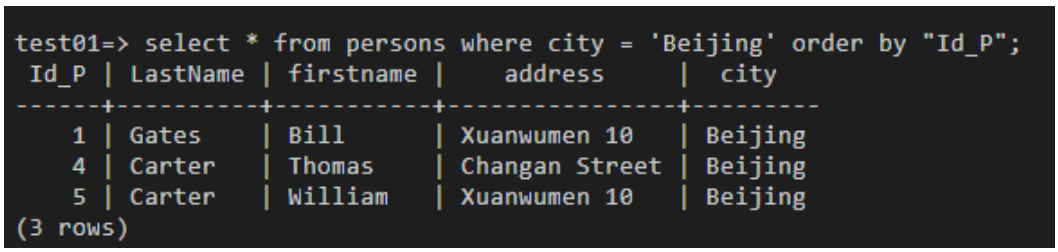
图 8-13 查询表数据



步骤4 抽样检查，验证表数据是否正确。

```
select * from persons where city = ‘Beijing' order by id_p;
```

图 8-14 验证表数据



步骤5 重复执行**步骤2**~**步骤4**查看其它库和表数据是否正确。

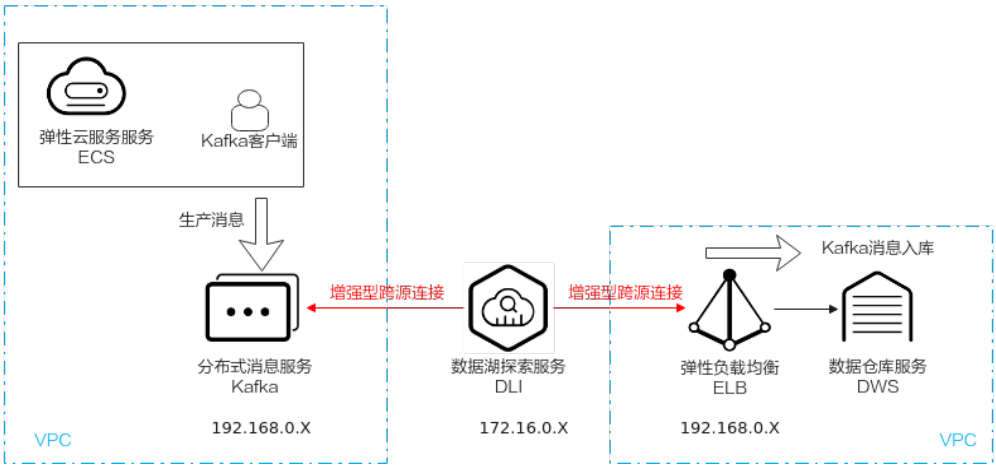
----结束

9 通过 DLI Flink 作业将 Kafka 数据实时写入 DWS

本实践演示通过**数据湖探索服务 DLI** Flink作业将**分布式消息服务 Kafka**的消费数据实时同步至GaussDB(DWS)数据仓库，实现Kafka实时入库到GaussDB(DWS)的过程。演示过程包括实时写入和更新已有数据的场景。

- 了解DLI请参见**数据湖产品介绍** 。
- 了解Kafka请参见**分布式消息服务Kafka产品介绍**。

图 9-1 Kafka 实时入库 DWS



本实践预计时长90分钟，实践用到的云服务包括**虚拟私有云 VPC及子网、弹性负载均衡 ELB、弹性云服务器 ECS、对象存储服务 OBS、分布式消息服务 Kafka、数据湖探索 DLI**和**数据仓库服务 GaussDB(DWS)**，基本流程如下：

1. **准备工作**
2. **步骤一：创建Kafka实例**
3. **步骤二：创建绑定ELB的DWS集群和目标表**
4. **步骤三：创建DLI队列**
5. **步骤四：分别创建Kafka和DWS的增强型跨源连接**
6. **步骤五：准备DWS对接Flink工具dws-connector-flink**

7. **步骤六：创建并编辑DLI Flink作业**
8. **步骤七：通过Kafka客户端生产和修改消息**

场景描述

假设数据源Kafka的样例数据是一个用户信息表，如**表9-1**所示， 包含 id，name，age三个字段。其中id是唯一且固定的字段，多个业务系统会公用，业务上一般不需要修改，仅修改姓名name，年龄age。

首先，通过Kafka生产以下三组数据，通过DLI Flink作业完成数据同步到数据仓库服务GaussDB(DWS)。接着，需要修改id为2和3的用户为新的jim和tom，再通过DLI Flink作业完成数据的更新并同步到GaussDB(DWS)。

表 9-1 样例数据

id	name	age
1	lily	16
2	lucy > jim	17
3	lilei > tom	15

约束限制

- 确保VPC、ECS、OBS、Kafka、DLI和DWS服务在同一个区域内，例如华北-北京四。
- 确保Kafka、DLI、DWS网络互通。本实践将Kafka和DWS创建在同一个区域和虚拟私有云下，同时在Kafka和DWS的安全组中放通了DLI的队列所在网段，确保网络互通。
- 为确保DLI到DWS的连接链路稳定，请创建完DWS集群后为集群绑定ELB服务。

准备工作

- 已注册华为账号并开通华为云，具体请参见**注册华为账号并开通华为云**，且在使用GaussDB(DWS) 前检查账号状态，账号不能处于欠费或冻结状态。
- 已创建虚拟私有云和子网，参见**创建虚拟私有云和子网**。

步骤一：创建 Kafka 实例

步骤1 登录华为云控制台，服务列表选择“应用中间件 > 分布式消息服务Kafka版”，进入Kafka管理控制台。

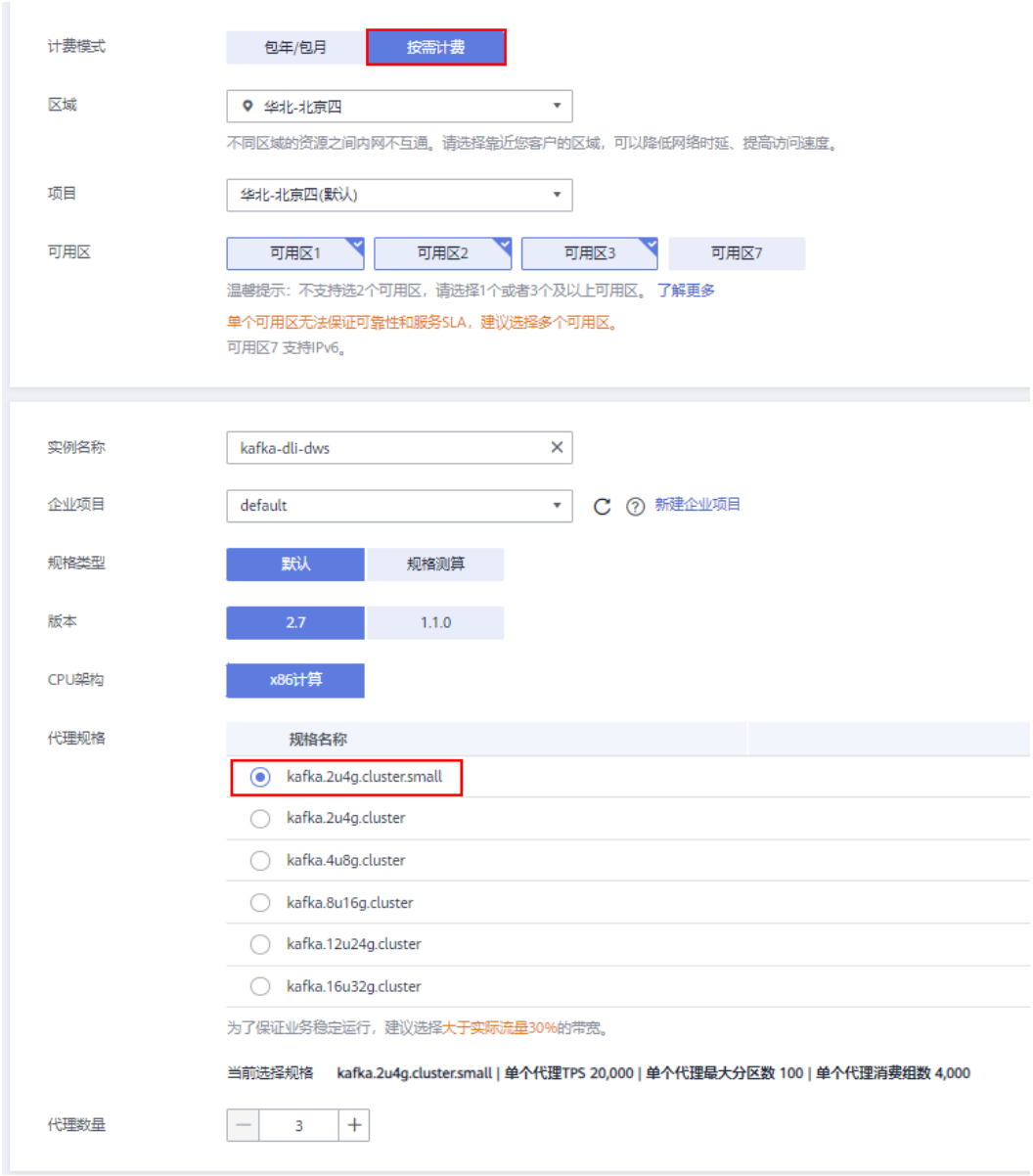
步骤2 左侧导航栏选择“Kafka专项版”，单击右上角的“购买kafka实例”。

步骤3 填写如下参数，其他参数项如表中未说明，默认即可：

表 9-2 kafka 实例参数

参数项	参数值
计费模式	按需计费
区域	华北-北京四
项目	默认
可用区	可用区1（如遇售罄，选择其他可用区）
实例名称	kafka-dli-dws
企业项目	default
规格类型	默认
版本	2.7
CPU架构	x86计算
代理规格	kafka.2u4g.cluster.small（实例仅为参考，选择最小规格即可）
代理数量	3
虚拟私有云	选择已创建的虚拟私有云，如果没有，则需要创建。
安全组	选择已创建的安全组，如果没有，则需要创建。
其他参数	保持默认

图 9-2 创建 Kafka 实例



步骤4 单击“立即创建”。等待创建成功。

步骤5 创建成功后，在Kafka实例列表中，单击创建好的Kafka实例名称，进入基本信息页面。

步骤6 左侧选择“Topic管理”，单击“创建Topic”。

Topic名称设置为“topic-demo”，其他可保持默认。

图 9-3 创建 Topic

步骤7 单击“确定”，在Topic列表中可以看到topic-demo已创建成功。

步骤8 左侧导航栏选择“消费组管理”，单击“创建消费组”。

步骤9 消费组名称输入“kafka01”，单击“确定”。

----结束

步骤二：创建绑定 ELB 的 DWS 集群和目标表

步骤1 **创建独享型弹性负载均衡服务ELB**，网络类型选择IPv4私网即可，区域、VPC选择与Kafka实例保持一致，本实践为“华北-北京四”。

步骤2 **创建集群**，为GaussDB(DWS)绑定弹性负载均衡 ELB，同时为确保网络连通，GaussDB(DWS)集群的区域、VPC选择与Kafka实例保持一致，本实践为“华北-北京四”，虚拟私有云与上面创建Kafka的虚拟私有云保持一致。

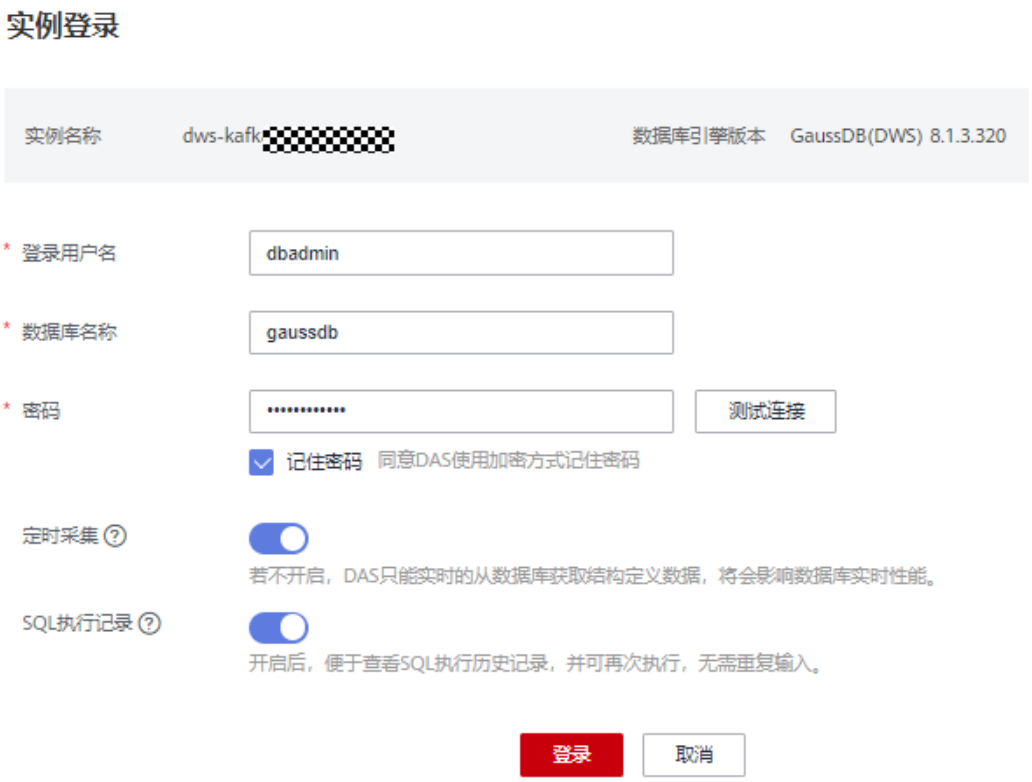
步骤3 在GaussDB(DWS)控制台的集群管理页面，单击指定集群所在行操作列的“登录”按钮。

说明

本实践以8.1.3.x版本为例，8.1.2及以前版本不支持此登录方式，可以**使用Data Studio连接集群**。

步骤4 登录用户名为dbadmin，数据库名称为gaussdb，密码为创建GaussDB(DWS)集群时设置的dbadmin用户密码，勾选“记住密码”，打开“定时采集”，“SQL执行记录”，单击“登录”。

图 9-4 登录 DWS



步骤5 单击“gaussdb”库名，再单击右上角的“SQL窗口”，进入SQL编辑器。

步骤6 复制如下SQL语句，在SQL窗口中，单击“执行SQL”，创建目标表user_dws。

```
CREATE TABLE user_dws (
id int,
name varchar(50),
age int,
PRIMARY KEY (id)
);
```

----结束

步骤三：创建 DLI 队列

步骤1 登录华为云控制台，服务列表选择“大数据 > 数据湖探索DLI”，进入DLI管理控制台。

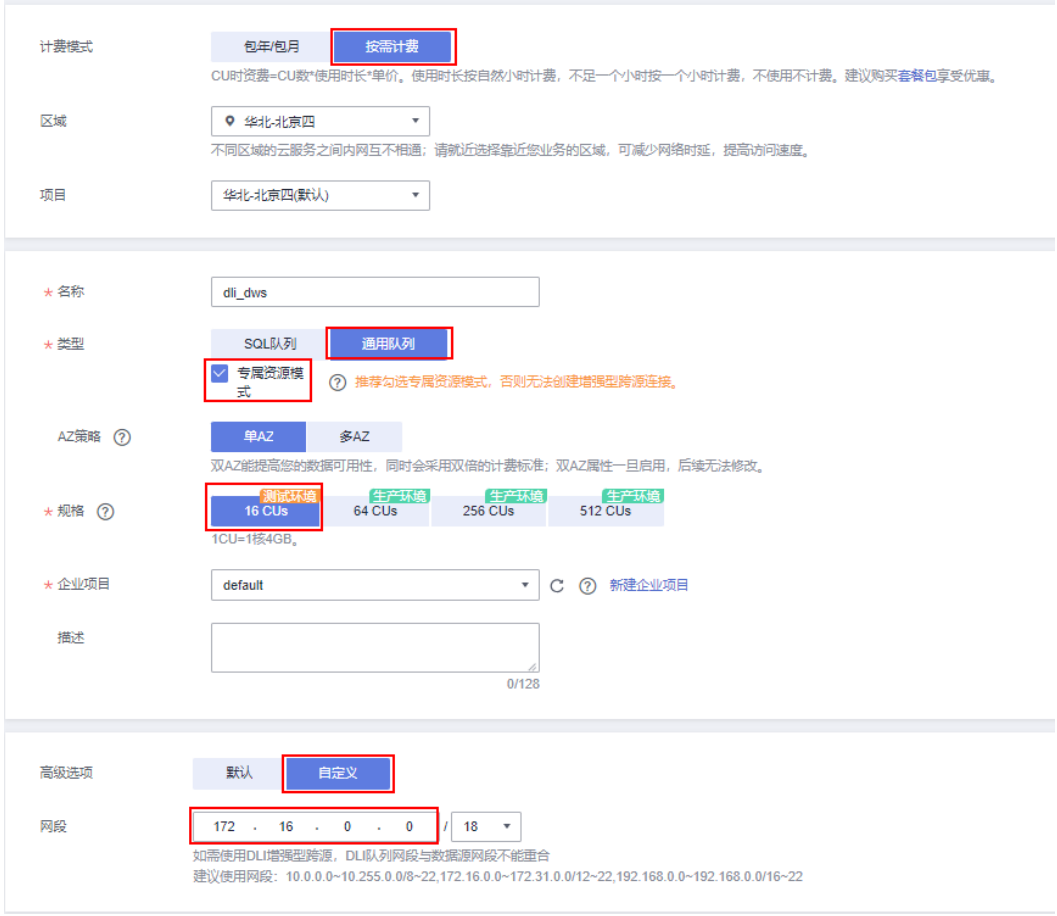
步骤2 左侧导航栏选择“资源管理 > 队列管理”，进入队列管理页面。

步骤3 单击右上角“购买队列”，填写如下参数，其他参数项如表中未说明，默认即可。

表 9-3 DLI 队列

参数项	参数值
计费模式	按需计费
区域	华北-北京四
项目	默认
名称	dli_dws
类型	通用队列，勾选“专属资源模式”。
AZ策略	单AZ
规格	16 CUs
企业项目	default
高级选项	自定义
网段	172.16.0.0/18，需选择与Kafka和DWS不在同一个网段。例如，如果Kafka和DWS在192.168.x.x网段，则DLI则选择172.16.x.x。

图 9-5 创建 DLI 队列



步骤4 单击“立即购买”。

----结束

步骤四：分别创建 Kafka 和 DWS 的增强型跨源连接

步骤1 放通Kafka的安全组，允许DLI队列所在的网段可以访问Kafka。

1. 回到Kafka控制台，单击Kafka实例名称进入基本信息。查看“连接信息”的“内网连接地址”，并记录下此地址，以备后续步骤使用。

图 9-6 kafka 内网连接地址

2. 单击网络的安全组名称。

图 9-7 kafka 安全组

3. 选择“入方向规则 > 添加规则”，如下图，添加DLI队列的网段地址，本实践为172.16.0.0/18，实际请与**步骤三：创建DLI队列**的时候填入的网段保持一致。

图 9-8 kafka 安全组添加规则

4. 单击“确定”。

步骤2 回到DLI管理控制台，单击左侧的“跨源管理”，选择“增强型跨源”，单击“创建”。

步骤3 填写如下参数，其他参数项如表中未说明，默认即可。

表 9-4 DLI 到 Kafka 的连接

参数项	参数值
连接名称	dli_kafka
弹性资源池	选择上面创建的DLI队列名称dli_dws。
虚拟私有云	选择Kafka所在的虚拟私有云。
子网	选择Kafka所在的子网。
其他参数	保持默认。

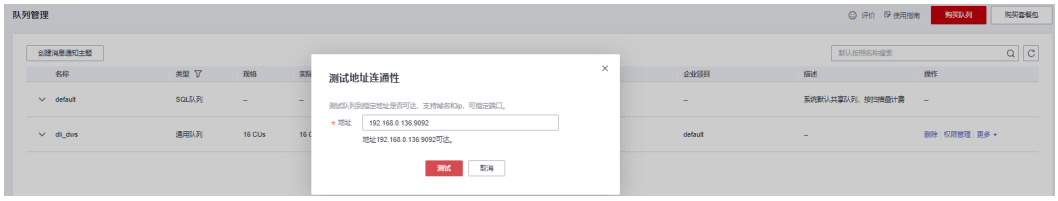
图 9-9 创建连接

步骤4 单击“确定”。等待Kafka连接创建成功。

步骤5 左侧导航栏选择“资源管理 > 队列管理”，选择dli_dws所在行操作列的“更多 > 测试地址连通性”。

步骤6 在地址栏中，输入**步骤1.1**获取的Kafka实例的内网IP和端口（Kafka的地址有三个，输入一个即可）。

图 9-10 测试 kafka 连通性

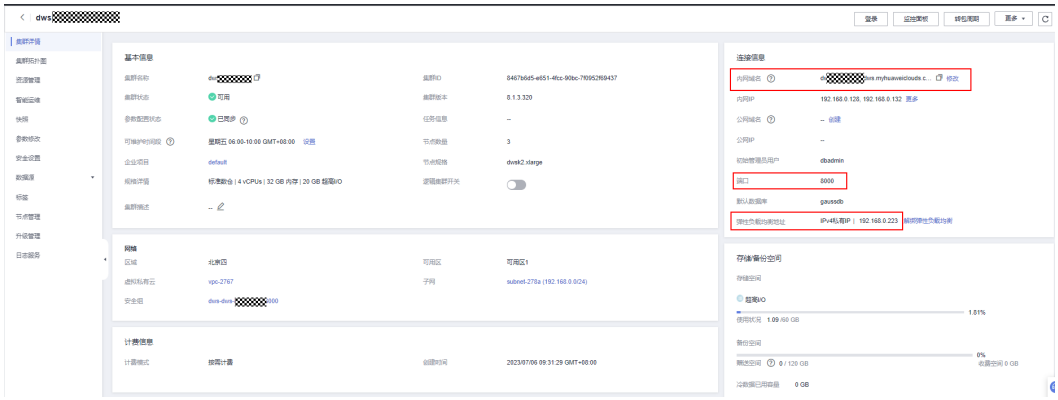


步骤7 单击“测试”，验证DLI连通Kafka成功。

步骤8 进入到DWS管理控制台，左侧导航栏单击“集群管理”，单击集群名称进入DWS集群详情。

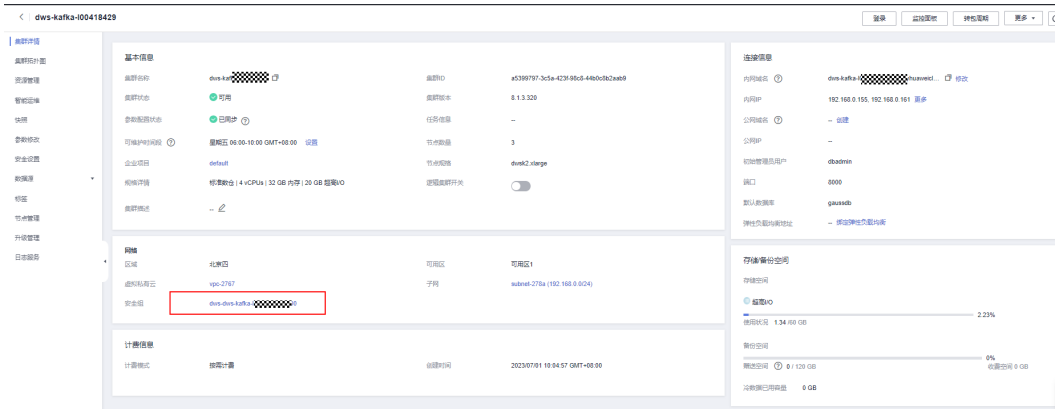
步骤9 如下图，记录下DWS集群的内网域名、端口和弹性负载均衡地址，以备后面步骤需要。

图 9-11 内网域名和 ELB 地址



步骤10 单击安全组名称。

图 9-12 DWS 安全组



步骤11 选择“入方向规则 > 添加规则”，如下图，添加DLI队列的网段地址，本实践为172.16.0.0/18，实际请与**步骤三：创建DLI队列**的时候填入的网段保持一致。

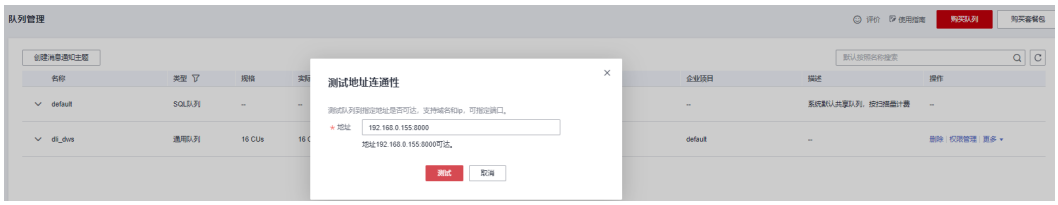
图 9-13 DWS 安全组添加规则

步骤12 单击“确定”。

步骤13 再切换到DLI控制台，左侧选择“资源管理 > 队列管理”，选择dli_dws所在行操作列的“更多 > 测试地址连通性”。

步骤14 在地址栏中，输入**步骤9**获取的GaussDB(DWS)集群的弹性负载均衡IP和端口。

图 9-14 测试 GaussDB(DWS)连通



步骤15 单击“测试”，验证DLI连通GaussDB(DWS)成功。

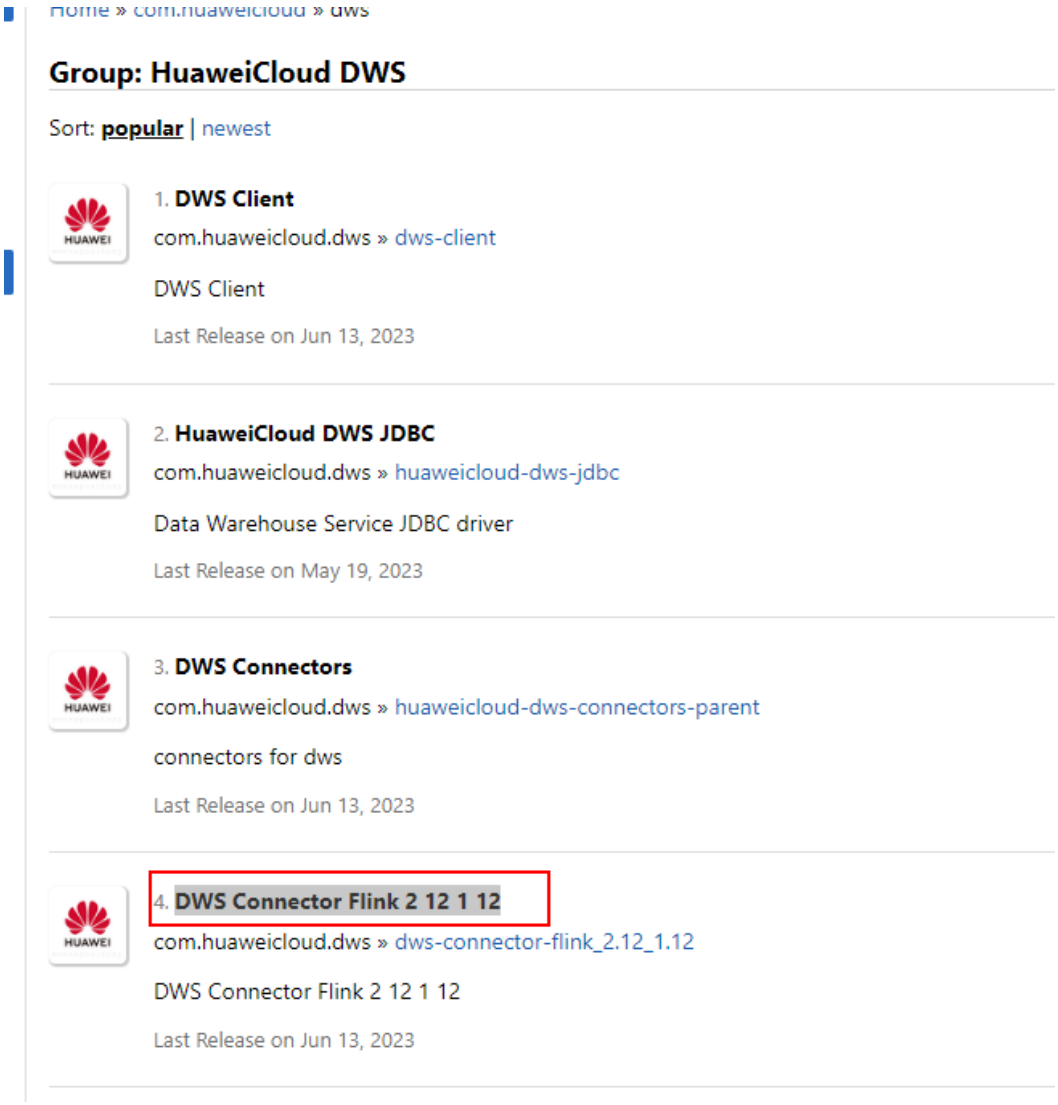
----结束

步骤五：准备 DWS 对接 Flink 工具 dws-connector-flink

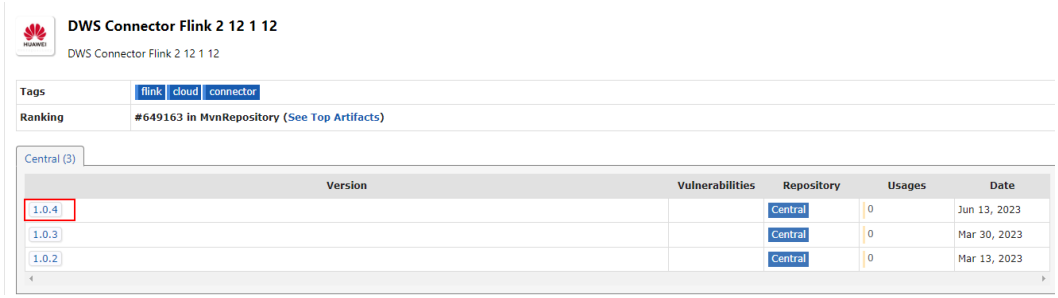
dws-connector-flink是一款基于DWS JDBC接口实现对接Flink的一个工具。在配置DLI作业阶段，将该工具及依赖放入Flink类加载目录，提升Flink作业入库DWS的能力。

步骤1 浏览器访问**https://mvnrepository.com/artifact/com.huaweicloud.dws**。

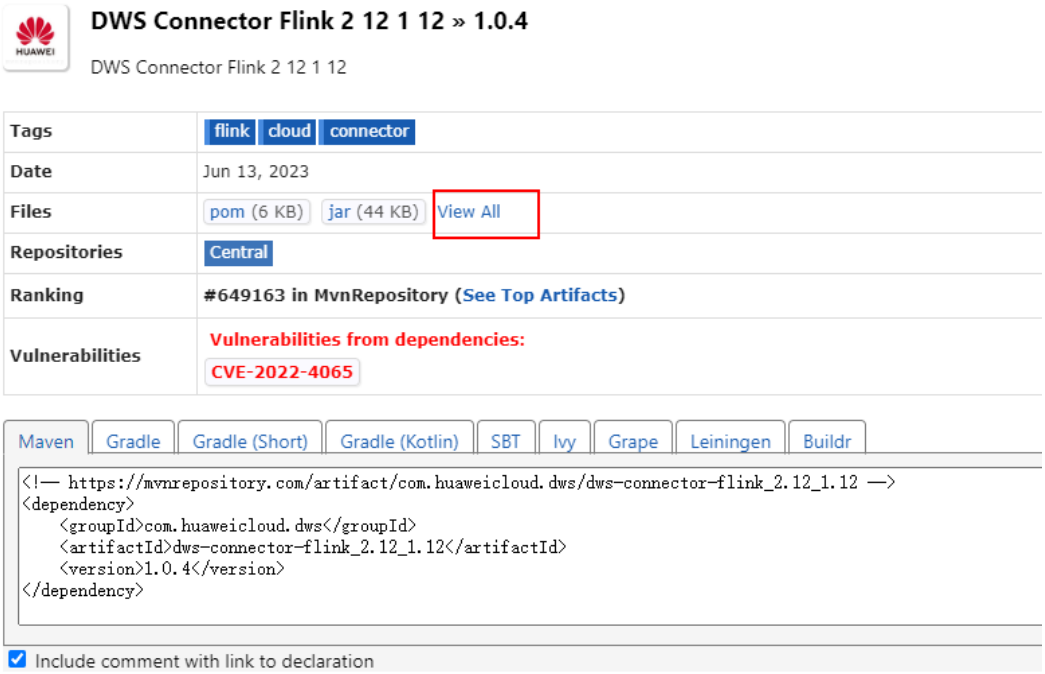
步骤2 在软件列表中选择最新版本的DWS Connectors Flink，本实践选择**DWS Connector Flink 2 12 1 12**。



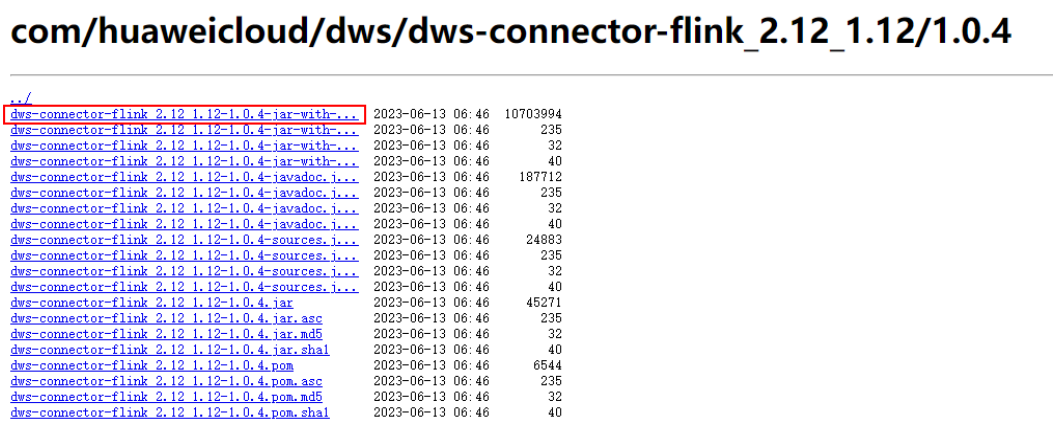
步骤3 单击“1.0.4”分支，实际请以官网发布的新分支为准。



步骤4 单击“View ALL”。



步骤5 单击dws-connector-flink_2.12_1.12-1.0.4-jar-with-dependencies.jar，下载到本地。



步骤6 创建OBS桶，本实践桶名设置为obs-flink-dws，并将此文件上传到OBS桶下，注意桶也保持与DLI在一个区域下，本实践为 华北-北京四。

图 9-15 上传 jar 包到 OBS 桶

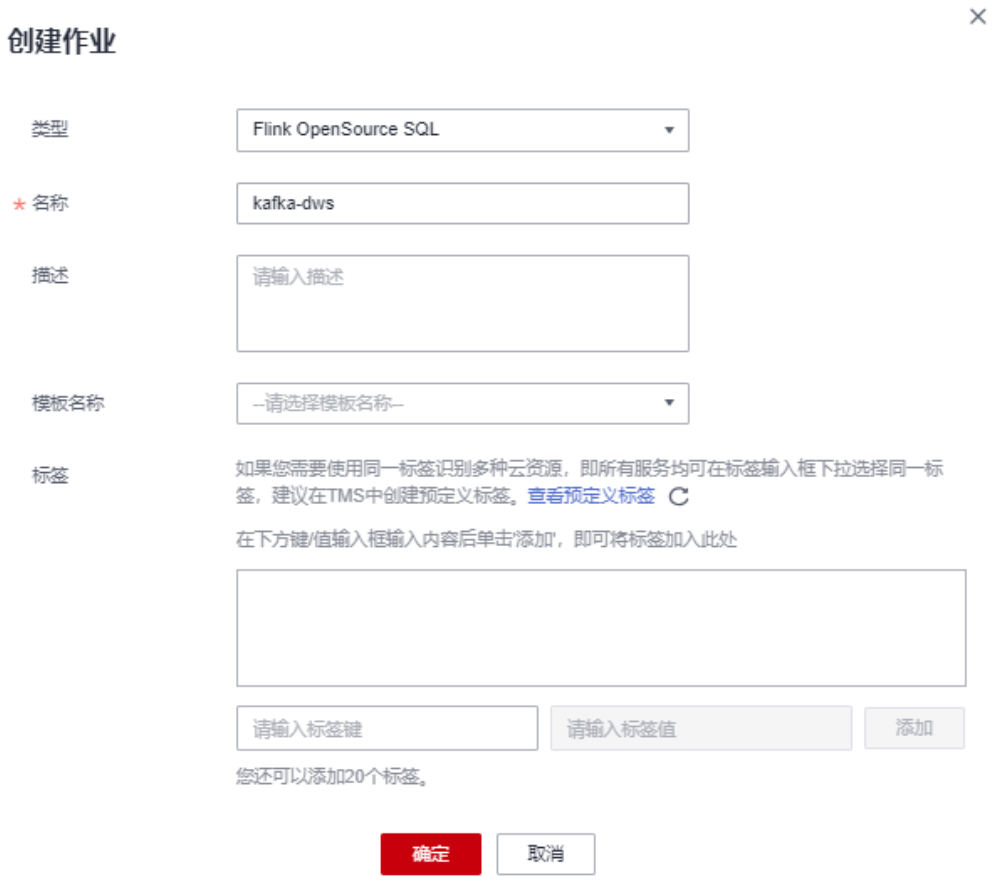
----结束

步骤六：创建并编辑 DLI Flink 作业

步骤1 回到DLI管理控制台，左侧选择“作业管理 > Flink作业”，单击右上角“创建作业”。

步骤2 类型选择“Flink OpenSource SQL”，名称填写kafka-dws。

图 9-16 创建作业



步骤3 单击“确定”。系统自动进入到作业的编辑页面。

步骤4 在页面右侧填写如下参数，其他参数项如表中未说明，默认即可。

表 9-5 flink 作业参数

参数项	参数值
所属队列	dli_dws
Flink版本	1.12

参数项	参数值
UDF Jar	选择**步骤五：准备DWS对接Flink工具dws-connector-flink**的OBS桶中的jar文件。 
OBS桶	选择**步骤五：准备DWS对接Flink工具dws-connector-flink**的桶。
开启Checkpoint	勾选
其他参数	保持默认

图 9-17 编辑作业

步骤5 将以下符合Flink要求的SQL代码复制到左侧的SQL代码窗。

其中“Kafka实例内网IP地址和端口”参见**步骤1.1**获取，“DWS内网域名”由**步骤9**获取。

```
CREATE TABLE user_kafka (
  id string,
  name string,
  age int
) WITH (
```

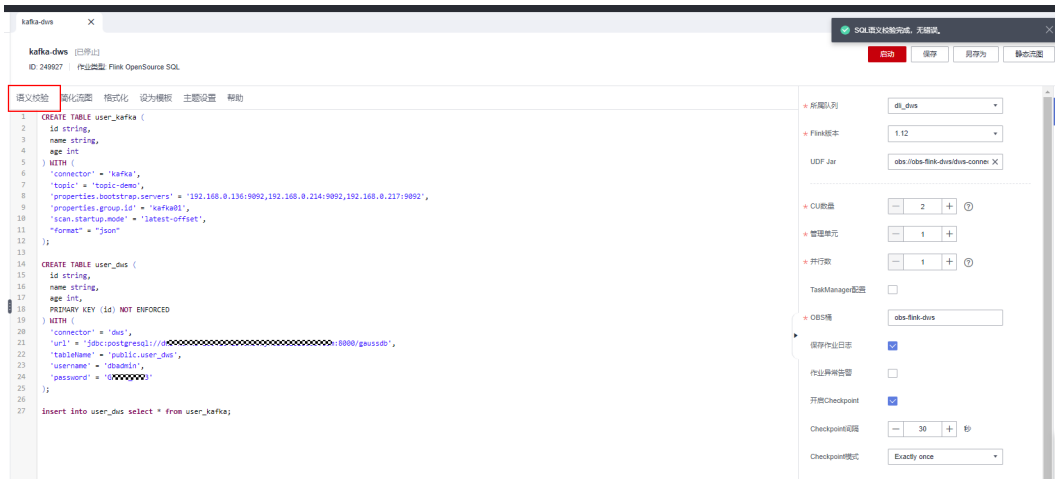


```
  'connector' = 'kafka',
  'topic' = 'topic-demo',
  'properties.bootstrap.servers' = 'Kafka实例内网IP地址和端口',
  'properties.group.id' = 'kafka01',
  'scan.startup.mode' = 'latest-offset',
  "format" = "json"
);

CREATE TABLE user_dws (
  id string,
  name string,
  age int,
  PRIMARY KEY (id) NOT ENFORCED
) WITH (
  'connector' = 'dws',
  'url' = 'jdbc:postgresql://DWS内网域名:8000/gaussdb',
  'tableName' = 'public.user_dws',
  'username' = 'dbadmin',
  'password' = '数据库用户dbdamin密码'
);

insert into user_dws select * from user_kafka;
```

步骤6 单击“语义校验”，等待校验成功。

如校验失败，则检查SQL的输入是否存在语法错误。

图 9-18 作业的 SQL 语句



步骤7 单击“保存”。

步骤8 回到DLI控制台首页，左侧选择“作业管理 > Flink作业”。

步骤9 单击作业名称kafka-dws右侧的“启动”，单击“立即启动”。

等待约1分钟，再刷新页面，状态在“运行中”表示作业成功运行。

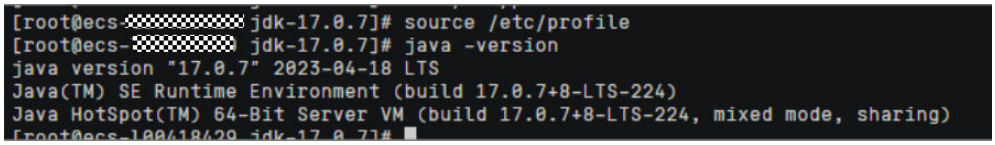
图 9-19 作业运行状态

----结束

步骤七：通过 Kafka 客户端生产和修改消息

步骤1 参见ECS文档创建一台ECS，具体创建步骤此处不再赘述。创建时，确保ECS的区域、虚拟私有云保持与Kafka一致。

步骤2 安装JDK。

1. 登录ECS，进入到/usr/local，下载JDK包。
   ```
   cd /usr/local
   wget https://download.oracle.com/java/17/latest/jdk-17_linux-x64_bin.tar.gz
   ```
2. 解压下载好的JDK包。
   ```
   tar -zxvf jdk-17_linux-x64_bin.tar.gz
   ```
3. 执行以下命令进入/etc/profile文件。
   ```
   vim /etc/profile
   ```
4. 按i进入编辑模式，将以下内容增加到/etc/profile文件的末尾。
   ```
   export JAVA_HOME=/usr/local/jdk-17.0.7  #jdk安装目录
   export JRE_HOME=${JAVA_HOME}/jre
   export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib:${JAVA_HOME}/test:${JAVA_HOME}/lib/gsjdbc4.jar:${JAVA_HOME}/lib/dt.jar:${JAVA_HOME}/lib/tools.jar:$CLASSPATH
   export JAVA_PATH=${JAVA_HOME}/bin:${JRE_HOME}/bin
   export PATH=$PATH:${JAVA_PATH}
   ```
   
5. 按ESC，输入:wq!按回车，保存退出。
6. 执行命令，使环境变量生效。
   ```
   source /etc/profile
   ```
7. 执行以下命令，提示如下信息表示jdk安装成功。
   ```
   java -version
   ```
   

步骤3 安装Kafka客户端。

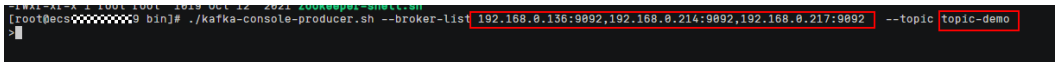
1. 进入/opt目录，执行以下命令获取Kafka客户端软件包。
   ```
   cd /opt
   wget https://archive.apache.org/dist/kafka/2.7.2/kafka_2.12-2.7.2.tgz
   ```
2. 解压下载好的软件包。
   ```
   tar -zxf kafka_2.12-2.7.2.tgz
   ```
3. 进入到Kafka客户端目录。
   ```
   cd /opt/kafka_2.12-2.7.2/bin
   ```

步骤4 执行以下命令连接Kafka。其中 {连接地址}为Kafka的内网连接地址，参见**步骤1.1**获取，topic为**步骤6**创建的Kafka的topic名称。

```
./kafka-console-producer.sh --broker-list {连接地址} --topic {Topic名称}
```

本实践示例如下：

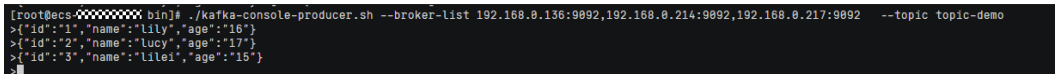
```
./kafka-console-producer.sh --broker-list
192.168.0.136:9092,192.168.0.214:9092,192.168.0.217:9092 --topic topic-demo
```



如上图出现>符号，无其他报错，表示连接成功。

步骤5 在已连接kafka的客户端窗口下，根据**场景描述**规划的数据，复制以下内容（注意一次复制一行），按回车发送，进行生产消息。

```
{"id":"1","name":"lily","age":"16"}
{"id":"2","name":"lucy","age":"17"}
{"id":"3","name":"lilei","age":"15"}
```



步骤6 回到DWS控制台，左侧选择“集群管理”，单击DWS集群右侧“登录”，进入SQL页面。

步骤7 执行以下SQL语句，发现数据实时入库成功。

```
SELECT * FROM user_dws ORDER BY id;
```

	id	name	age
1	1	lily	16
2	2	lucy	17
3	3	lilei	15

步骤8 继续回到ECS中连接Kafka的客户端窗口，复制以下内容（注意一次复制一行），按回车发送，进行生产消息。

```
{"id":"2","name":"jim","age":"17"}
{"id":"3","name":"tom","age":"15"}
```

步骤9 回到DWS已打开的SQL窗口，执行以下SQL语句，发现id为2和3的姓名已修改为jim和tom。

符合场景描述预期，本实践结束。

```
SELECT * FROM user_dws ORDER BY id;
```

	id	name	age
1	1	lily	16
2	2	jim	17
3	3	tom	15

----结束

10 SQL 基本操作

本节主要介绍GaussDB(DWS)数据库的一些SQL基本操作。

创建、查看和删除数据库

- 使用CREATE DATABASE语句创建数据库。
  ```
  CREATE DATABASE test_db ENCODING 'UTF8' template = template0;
  ```
- 使用\l元命令查看数据库系统的数据库列表。
  ```
  \l
  ```
- 通过系统表PG_DATABASE查询数据库列表。
  ```
  SELECT datname FROM pg_database;
  ```
- 使用DROP DATABASE语句删除数据库。
  ```
  DROP DATABASE test_db;
  ```

创建、查看、修改和删除表

- 使用CREATE TABLE语句创建表。
  ```
  CREATE TABLE customer_t1(id INT, name CHAR(40),age TINYINT);
  ```
- 使用PG_GET_TABLEDEF()函数查看建表语句。
  ```
  SELECT * FROM PG_GET_TABLEDEF('customer_t1');
  ```
- 使用ALTER TABLE语句修改表。

  增加列：
  ```
  ALTER TABLE customer_t1 ADD (address VARCHAR(100));
  ```
  删除列：
  ```
  ALTER TABLE customer_t1 DROP COLUMN address;
  ```
  修改字段类型：
  ```
  ALTER TABLE customer_t1 MODIFY age INTEGER NOT NULL;
  ```
- 使用DROP TABLE语句删除表。
  ```
  DROP TABLE customer_t1;
  ```

创建、查看和删除索引

- 使用CREATE INDEX或ALTER TABLE语句创建普通索引。
  ```
  CREATE INDEX c_id_index on customer_t1(id);
  ALTER TABLE customer_t1 ADD INDEX c_id_index (id);
  ```
- 使用PG_INDEXES系统表查看表内所有索引。
  ```
  SELECT * FROM pg_indexes WHERE tablename = 'customer_t1';
  ```


- 使用ALTER TABLE或DROP INDEX语句删除索引。
```
DROP INDEX c_id_index;
ALTER TABLE customer_t1 DROP INDEX c_id_index;
```

增删改查表数据

- 使用INSERT INTO语句插入表数据。
```
INSERT INTO customer_t1 VALUES(1001,'user1',22);
```
- 使用SELECT语句查询表数据。
```
SELECT * FROM customer_t1;
```
- 使用UPDATE更新表数据。
```
UPDATE customer_t1 SET id = 1009 WHERE id = '1001';
```
- 使用DELETE删除表数据。
```
DELETE FROM customer_t1 WHERE id = '1009';
```

11 入门实践

当用户完成集群创建后，可以根据自身的业务需求使用GaussDB(DWS)提供的一系列常用实践。

表 11-1 常用最佳实践

实践		描述
数据导入导出	从OBS导入数据到集群	本教程旨在通过演示将样例数据上传OBS，及将OBS的数据导入进GaussDB(DWS)上的目标表中，让您快速掌握如何从OBS导入数据到GaussDB(DWS)集群的完整过程。 GaussDB(DWS)支持通过外表将OBS上TXT、CSV、ORC、PARQUET、CARBONDATA以及JSON格式的数据导入到集群进行查询。
	使用GDS从远端服务器导入数据	本教程旨在演示使用GDS（General Data Service）工具将远端服务器上的数据导入GaussDB(DWS)中的办法，帮助您学习如何通过GDS进行数据导入的方法。 GaussDB(DWS)支持通过GDS外表将TXT、CSV和FIXED格式的数据导入到集群进行查询。

实践		描述
	导入远端DWS数据源	大数据融合分析场景下，支持同一区域内的多套GaussDB(DWS)集群之间的数据互通互访，本实践将演示通过Foreign Table方式从远端DWS导入数据到本端DWS。 本实践演示过程为：以gsql作为数据库客户端，gsql安装在ECS，通过gsql连接DWS，再通过外表方式导入远端DWS的数据。
	导出ORC数据到MRS	GaussDB(DWS)数据库支持通过HDFS外表导出ORC格式数据至MRS，通过外表设置的导出模式、导出数据格式等信息来指定导出的数据文件，利用多DN并行的方式，将数据从GaussDB(DWS)数据库导出到外部，存放在HDFS文件系统上，从而提高整体导出性能。
数据迁移	Oracle迁移到GaussDB(DWS)实践	本教程演示将Oracle业务相关的表数据迁移到GaussDB(DWS)的数据库的基本过程。
	MySQL表数据实时同步到GaussDB(DWS)实践	本实践演示通过华为云数据复制服务DRS完成MySQL数据实时同步到GaussDB(DWS)的基本过程。
	通过DLI Flink作业将Kafka数据实时写入DWS	本实践演示通过数据湖探索服务DLI Flink作业将分布式消息服务Kafka的消费数据实时同步至DWS数据仓库，实现Kafka实时入库到DWS的过程。 本实践预计时长90分钟，实践用到的云服务包括虚拟私有云VPC及子网、弹性负载均衡ELB、弹性云服务器ECS、对象存储服务OBS、分布式消息服务Kafka、数据湖探索DLI和数据仓库服务DWS

实践		描述
调优表	调优表实践	在本实践中，您将学习如何优化表的设计。您首先不指定存储方式，分布键、分布方式和压缩方式创建表，然后为这些表加载测试数据并测试系统性能。接下来，您将应用优秀实践以使用新的存储方式、分布键、分布方式和压缩方式重新创建这些表，并再次为这些表加载测试数据和测试系统性能，以便比较不同的设计对表的加载性能、存储空间和查询性能的影响。 估计时间：60 分钟。
高级特性	冷热数据管理优秀实践	海量大数据场景下，随着业务和数据量的不断增长，数据存储与消耗的资源也日益增长。根据业务系统中用户对不同时期数据的不同使用需求，对膨胀的数据进行“冷热”分级管理，不仅可以提高数据分析性能还能降低业务成本。针对数据使用的一些场景，可以将数据按照时间分为：热数据、冷数据。

实践		描述
	分区自动管理优秀实践	对于分区列为时间的分区表，分区自动管理功能可以自动创建新分区和删除过期分区，降低分区表的维护成本，改善查询性能。为了便于查询和维护数据，用户通常使用分区列为时间的分区表来存储时间相关的数据，例如电商的订单信息、 物联网采集的实时数据。这些时间相关的数据导入分区表时，需要保证分区表要有对应时间的分区，由于普通的分区表不会自动创建新的分区和删除过期的分区，所以维护人员需要定期创建新分区和删除过期分区，提高了运维成本。 为解决上述问题，GaussDB(DWS) 引入了分区自动管理特性。可通过设置表级参数period、ttl开启分区自动管理功能，使分区表可以自动创建新分区和删除过期分区，降低分区表的维护成本，改善查询性能。
数据库管理	资源管理优秀实践	本实践将演示GaussDB(DWS)的资源管理功能，帮助企业客户解决数据分析过程中，多用户查询作业遇到的性能瓶颈，最终实现多用户执行SQL作业互不影响，节省资源消耗。
	SQL查询优秀实践	根据数据库的SQL执行机制以及大量的实践总结发现：通过一定的规则调整SQL语句，在保证结果正确的基础上，能够提高SQL执行效率。
	数据倾斜查询优秀实践	本实践包含以下存储倾斜案例： • 导入过程存储倾斜即时检测 • 快速定位查询存储倾斜的表
	用户管理优秀实践	GaussDB(DWS)集群中，常用的用户分别是系统管理员和普通用户。本实践简述了系统管理员和普通用户的权限，如何创建以及如何查询用户相关信息。

实践		描述
	查看表和数据库的信息	本实践演示了基本数据库查询案例： ● 查询表信息 ● 查询表大小 ● 查询数据库 ● 查询数据库大小
模拟数据分析	交通卡口通行车辆分析	本实践将演示交通卡口车辆通行分析，将加载8.9亿条交通卡口车辆通行模拟数据到数据仓库单个数据库表中，并进行车辆精确查询和车辆模糊查询，展示GaussDB(DWS) 对于历史详单数据的高性能查询能力。
	供应链需求分析(TPC-H数据集)	本实践将演示从OBS加载样例数据集到GaussDB(DWS) 集群中并查询数据的流程，从而向您展示GaussDB(DWS) 在数据分析场景中的多表分析与主题分析。
	零售业百货公司经营状况分析	本实践将演示以下场景：从OBS加载各个零售商场每日经营的业务数据到数据仓库对应的表中，然后对商铺营业额、客流信息、月度销售排行、月度客流转化率、月度租售比、销售坪效等KPI信息进行汇总和查询。本示例旨在展示在零售业场景中GaussDB(DWS) 数据仓库的多维度查询分析的能力。
数据安全	实现数据列的加解密	数据加密作为有效防止未授权访问和防护数据泄露的技术，在各种信息系统中广泛使用。作为信息系统的核心，GaussDB(DWS)数仓也提供数据加密功能，包括透明加密和使用SQL函数加密。本章节主要讨论SQL函数加密。