ALIBABA CLOUD

Alibaba Cloud

云原生数据仓库 AnalyticDB PostgreSQL 版

数据接入

文档版本：20201015

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
`Courier字体`	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.通过实时计算Flink版读取云原生数据仓库AnalyticDB PostgreSQL

本文介绍如何通过阿里云实时计算Flink版实时读取云原生数据仓库AnalyticDB PostgreSQL（以下简称ADB PG版，原分析型数据库PostgreSQL版）数据，包括版本限制、语法示例、创建和运行Flink作业、WITH参数、CACHE参数、类型映射和参数支持等。

版本限制

- 创建3.6.0及以上版本实时计算集群
- 创建6.0版本ADB PG集群（实时计算集群和ADB PG版实例需要位于同一VPC下，且ADB PG版实例的白名单规则允许Flink集群网段访问）。

语法示例

```
CREATE TABLE dim_adbpg(
        id int,
        username varchar,
        INDEX(id)
) with(
        type='custom',
        tableFactoryClass='com.alibaba.blink.customersink.ADBPGCustomSourceFactory',
        url='jdbc:postgresql://内网连接串/databasename',
        tableName='tablename',
        userName='username',
        password='password',
        joinMaxRows='100',
        maxRetryTimes='1',
        connectionMaxActive='5',
        retryWaitTime='100',
        targetSchema='public',
        caseSensitive='0',
        cache='LRU',
        cacheSize='1000',
        cacheTTLMs='10000',
        cacheReloadTimeBlackList='2017-10-24 14:00 -> 2017-10-24 15:00',
        partitionedJoin='true'
);

-- join时需要指定在代码中加入维表标识 FOR SYSTEM_TIME AS OF PROCTIME()
INSERT INTO print_sink
SELECT R.c1, R.a2, R.a3, R.a4, R.a5, R.a6, R.a6, R.a8, R.a9, R.a10, R.a11, R.a13, T.username
FROM s_member_cart_view AS R
left join
dim_adbpg FOR SYSTEM_TIME AS OF PROCTIME() AS T
on R.c1 = T.id;
```

WITH参数

参数名	参数含义	备注

参数名	参数含义	备注
url	ADBPG连接地址	必填，需要填写需要填写格式为jdbc:postgresql://<ADBPG内网连接串>/databaseName 的内网连接地址。
tableName	ADBPG源表名	必填，填写维表对应的ADBPG数据仓库中的表名。
userName	ADBPG用户名	必填。
password	ADBPG密码	必填。
joinMaxRows	左表一条记录连接右表的最大记录数	非必填，表示在一对多连接时，左表一条记录连接右表的最大记录数（默认值为1024）。在一对多连接的记录数过多时，可能会极大的影响流任务的性能，因此您需要增大Cache的内存（cacheSize限制的是左表key的个数）。
maxRetryTimes	单次SQL失败后重试次数	非必填，实际执行时，可能会因为各种因素造成执行失败，比如网络或者IO不稳定，超时等原因，ADBPG维表支持SQL执行失败后自动重试，用maxRetryTimes参数可以设定重试次数。默认值为3。
connectionMaxActive	连接池最大连接数	非必填，ADBPG维表中内置连接池，设置合理的连接池最大连接数可以兼顾效率和安全性，默认值为5。
retryWaitTime	重试休眠时间	非必填，每次SQL失败重试之间的sleep间隔，单位ms，默认值100

参数名	参数含义	备注
targetSchema	查询的ADBPG schema	非必填，默认值public
caseSensitive	是否大小写敏感	非必填，默认值0，即不敏感；填1可以设置为敏感；

CACHE参数

参数名	参数含义	备注
cache	缓存策略	目前ADB PG版支持以下三种缓存策略： • None（默认值）：无缓存。 • LRU：缓存维表里的部分数据。源表来一条数据，系统会先查找Cache，如果没有找到，则去物理维表中查询。 • ALL：缓存维表里的所有数据。在Job运行前，系统会将维表中所有数据加载到Cache中，之后所有的维表查询都会通过Cache进行。如果在Cache中无法找到数据，则KEY不存在，并在Cache过期后重新加载一遍全量Cache。
cacheSize	设置LRU缓存的最大行数	非必填，默认为10000行。
cacheTTLMs	缓存更新时间间隔。系统会根据您设置的缓存更新时间间隔，重新加载一次维表中的最新数据，保证源表能JOIN到维表的最新数据。	非必填，单位为毫秒。默认不设置此参数，表示不重新加载维表中的新数据。
cacheReloadTimeBlackList	更新时间黑名单。在缓存策略选择为ALL时，启用更新时间黑名单，防止在此时间内做Cache更新（例如双11场景）。	非必填，默认空，格式为 '2017-10-24 14:00 -> 2017-10-24 15:00, 2017-11-10 23:30 -> 2017-11-11 08:00' 。其中分割符使用情况如下： • 用逗号（,）来分隔多个黑名单。 • 用箭头（->）来分割黑名单的起始结束时间。

参数名	参数含义	备注
partitionedJoin	是否开启partitionedJoin。在开启partitionedJoin优化时，主表会在关联维表前，先按照Join KEY进行Shuffle，这样做有以下优点： • 在缓存策略为LRU时，可以提高缓存命中率。 • 在缓存策略为ALL时，节省内存资源，因为每个并发只缓存自己并发所需要的数据。	非必填，默认情况下为false，表示不开启partitionedJoin。

类型映射

实时计算字段类型	ADB PG版字段类型
BOOLEAN	BOOLEAN
TINYINT	SMALLINT
SMALLINT	SMALLINT
INT	INT
BIGINT	BIGINT
DOUBLE	DOUBLE PRECISION
VARCHAR	TEXT
DATETIME	TIMESTAMP
DATE	DATE
FLOAT	REAL
DECIMAL	DOUBLE PRECISION

实时计算字段类型	ADB PG版字段类型
TIME	TIME
TIMESTAMP	TIMESTAMP

创建和运行Flink作业

在实时计算控制台上，点击项目管理-项目列表，进入项目名进入自己创建的项目：

点击开发-新建作业，创建数据写入的Flink SQL作业：

目前采用Flink自定义维表的方式支持读取ADB PG版目标表数据，使用自定义维表功能上线前需要在资源引用界面上传及引用jar包，编写完作业后点击资源引用->新建资源->上传jar包->更多->引用：

通过以下链接下载jar包：

https://adbpg-public.oss-cn-beijing.aliyuncs.com/blink-customerdim-adbpg-0909.jar

完成作业开发后，依次点击保存、上线，即可上线该任务。

继续点击运维，启动对应项目即可启动任务。

代码示例

这里给出读取ADB PG版数据打印到Flink日志中的Flink SQL示例：

```
--SQL
--********************************************************************--
--Author: zihua
--CreateTime: 2019-09-07 10:34:34
--********************************************************************--

CREATE TABLE s_member_cart
(
    a1 int,
    a2 tinyint  ,
    a3 smallint ,
    a4 int,
    a5 boolean,
```

```
    a6 FLOAT    ,
    a7 DECIMAL  ,
    a8 double,
    a9 date   ,
    a10 time    ,
    a11 timestamp ,
    a12 tinyint
) WITH (
    type='random'
);

CREATE VIEW s_member_cart_view AS
SELECT MOD(a1, 10) c1, a2, a3, a4, a5, a6, a6, a8, a9, a10, a11, case when a12 >0 then 'test1' else 'test5'
end as b12,'{ "customer": "中文56", "items": {"product": "Beer","qty": 6}}' a13
FROM s_member_cart;

--adbpg dim index
CREATE TABLE dim_adbpg(
    id int,
    username varchar,
    INDEX(id)
) with(
    type='custom',
    tableFactoryClass='com.alibaba.blink.customersink.ADBPGCustomSourceFactory',
    url='jdbc:postgresql://内网连接串/databasename',
    tableName='tablename',
    userName='username',
    password='password',
    joinMaxRows='100',
    maxRetryTimes='1',
    connectionMaxActive='5',
    retryWaitTime='100',
    targetSchema='public',
    caseSensitive='0',
    cache='LRU',
    cacheSize='1000',
    cacheTTLMs='10000',
    cacheReloadTimeBlackList='2017-10-24 14:00 -> 2017-10-24 15:00',
    partitionedJoin='true'
);
```

```
-- ads sink.
CREATE TABLE print_sink (
    B1 int,
    B2 tinyint  ,
    B3 smallint  ,
    B4 int,
    B5 boolean,
    B6 FLOAT    ,
    B7 FLOAT   ,
    B8 double,
    B9 date   ,
    B10 time    ,
    B11 timestamp ,
    B12 varchar,
    B15 varchar,
    PRIMARY KEY(B1)
) with (
    type='print'
);


INSERT INTO print_sink
SELECT R.c1, R.a2, R.a3, R.a4, R.a5, R.a6, R.a6, R.a8, R.a9, R.a10, R.a11, R.a13, T.username
FROM s_member_cart_view AS R
left join
dim_adbpg FOR SYSTEM_TIME AS OF PROCTIME() AS T
on R.c1 = T.id;
```

2.数据迁移及同步方案综述

AnalyticDB for PostgreSQL 提供了多种数据迁移方案，可满足不同的数据同步或迁移的业务需求，使您可以在不影响业务的情况下，平滑地与各种数据库类型实例之间进行迁移或数据同步，包括 RDS MySQL，PolarDB for MySQL， RDS PostgreSQL、RDS PPAS、MaxCompute、Greenplum Database、以及自建MySQL 、 PostgreSQL 或 Amazon Redshift。除下面所列阿里云方案外，也支持第三方数据同步产品，例如DSG。

AnalyticDB for PostgreSQL支持的各种数据迁移应用场景及相关操作如下：

操作	类型	场景
OSS外表高速导入或导出OSS数据	数据迁移	通过OSS外部表将数据在AnalyticDB for PostgreSQL和OSS之间进行导入或者导出。
Dataworks数据集成	数据同步/数据迁移	通过数据集成（Data Integration）服务，实现分钟级延迟的数据同步，支持AnalyticDB for PostgreSQL作为同步的目标端或者源端，同其它多种异构数据源，进行数据的导入或者导出。
COPY命令导入或导出本地数据	数据迁移	通过 `\COPY` 命令，将本地的文本文件数据导入到AnalyticDB for PostgreSQL中。
rds_dbsync迁移/同步MySQL数据到AnalyticDB for PostgreSQL	数据同步/数据迁移	通过开源项目rds_dbsync的mysql2pgsql工具将本地MySQL中的表同步到AnalyticDB for PostgreSQL中。
rds_dbsync迁移/同步PostgreSQL数据到AnalyticDB for PostgreSQL	数据同步/数据迁移	通过开源项目rds_dbsync的pgsql2pgsql工具将AnalyticDB for PostgreSQL/Greenplum Database/PostgreSQL/PPAS中的表同步到AnalyticDB for PostgreSQL中。

1. 数据迁移：是指将各种数据库实例或者本地存储的数据迁移到AnalyticDB for PostgreSQL。
2. 数据同步：是指将其他数据库中的数据实时同步到AnalyticDB for PostgreSQL。

3.通过实时计算 Flink 写入数据

Blink 3.6.0版本开始支持通过Blink connector将数据写入云原生数据仓库PostgreSQL版（以下简称ADB PG版），本文将为您介绍使用的必要条件、操作流程、字段映射和参数支持。

前提条件

- 实时计算集群和ADB PG版实例位于同一VPC下，且ADB PG实例的白名单规则允许Blink集群网段访问。
- 实时计算集群为3.6.0及以上版本，可按以下步骤创建。
  i. 开通阿里云实时计算服务和项目，请参见开通服务和创建项目。

  > 说明 开通的实时计算集群与目标ADB PG版集群必须在同一VPC下。

  ii. 确认并安装实时计算集群3.6.0及以上版本，请参见管理独享集群Blink版本。

  
- 设置6.0版本ADB PG版实例。
  i. 创建实例。

  > 说明 开通的ADB PG版实例与实时计算集群必须在同一VPC下。

  ii. 设置ADB PG版实例白名单。
  a. 在VPC控制台找到对应网段的ip地址。

  

  b. 在ADB PG版控制台点击目标ADB PG版实例ID，在实例详情页面，单击**数据安全性>添加白名单分组**。

  

  c. 将对应的VPC网段添加进ADB PG版实例白名单，单击**确定**。

  
  iii. 创建ADB PG版目标表。

```
create table test15(
b1 bigint,
b2 smallint,
b3 smallint,
b4 int,
b5 boolean,
b6 real,
b7 double precision,
b8 double precision,
b9 date,
b10 time with time zone,
b11 timestamp with time zone,
b12 text,
b15 json
);
```

创建数据写入任务

为了方便介绍，本节的数据源采用随机数据源（random），实际使用中可以根据实际情况创建数据源。

1. 在实时计算控制台上，点击**项目管理>项目列表**，单击项目名进入目标项目。

2. 点击**开发>新建作业**，创建数据写入的Flink SQL作业。

写入ADB PG的作业举例。

```
--SQL
--********************************************************************--
--Author: sqream_test
--CreateTime: 2020-04-27 19:13:44
--********************************************************************--

CREATE TABLE s_member_cart
(
a1 bigint   ,
a2 tinyint    ,
a3 smallint  ,
a4 int     ,
a5 boolean   ,
a6 FLOAT    ,
a7 DECIMAL   ,
```

```
a8 double    ,
a9 date      ,
a10 time     ,
a11 timestamp    ,
a12 tinyint



) WITH (
  type='random'
);

-- ads sink.
CREATE TABLE adsSink (
  `B1` bigint   ,
  `B2` tinyint  ,
  `B3` smallint  ,
  `B4` int      ,
  `B5` boolean,
  `B6` FLOAT    ,
  `B7` FLOAT   ,
  `B8` double    ,
  `B9` date     ,
  `B10` time     ,
  `B11` timestamp    ,
  `B12` varchar,
  `B15` varchar
  --PRIMARY KEY(b1)
) with (
  --type='print'
  type='adbpg',
  version='1.1',
  url='jdbc:postgresql://gp-xxxx:3432/testblink',
  tableName='test',
  userName='xxxx',
  password='xxxxxx',
  timeZone='Asia/Shanghai',
  useCopy='0'
);
```

```
INSERT INTO adsSink
SELECT a1,a2,a3,a4,a5,a6,a6,a8,a9,a10,a11, case when a12 >0 then 'value1' else 'value2' end as b12,'{ "c
ustomer": "value", "items": {"product": "Beer","qty": 6}}'
    from s_member_cart;

--insert into adsSink2 select a2, sum(a4) from s_member_cart group by a2;
```

参数说明

参数	参数说明	是否必填	备注
type	源表类型	是	固定值：adbpg。
url	JDBC连接地址	是	分析型数据库PostgreSQL版数据库的JDBC连接地址 。格式为‘jdbc:postgresql://<yourNetworkAddress>:<PortId>/<yourDatabaseName>’，其中yourNetworkAddress：内网地址。PortId：连接端口。yourDatabaseName：连接的数据库名称。示例：url='jdbc:postgresql://gp-xxxxxx.gpdb.cn-chengdu.rds.aliyuncs.com:3432/postgres'
tableName	表名	是	无。
username	账号	是	无。
password	密码	是	无。
maxRetryTimes	写入重试次数	否	默认为3。

参数	参数说明	是否必填	备注
useCopy	是否采用copy API写入数据	否	参数取值如下 • 1：采用copy API方式写入数据。 • 0：采用其他方式写入数据，例如BATCH INSERT或BATCH UPSERT。 blink 3.6.0 版本默认为0，3.6.4及以上版本默认值为1；当取值为0时，会根据writeMode字段选择数据写入方式。
batchSize	一次批量写入的条数	否	默认值为5000。
exceptionMode	数据写入过程中出现异常时的处理策略	否	支持以下两种处理策略 • ignore（默认值）：忽略出现异常时写入的数据。 • strict：数据写入异常时，Failover报错。
conflictMode	当出现主键冲突或者唯一索引冲突时的处理策略	否	支持以下三种处理策略 • ignore （默认值）：忽略主键冲突，保留之前的数据。 • strict：主键冲突时，Failover报错。 • update：主键冲突时，更新新到的数据。 • upsert：主键冲突时，采用upsert方式写入数据。
targetSchema	Schema名称	否	默认值为public。

参数	参数说明	是否必填	备注
writeMode	在useCopy字段基础上，更细分的写入方式	否	blink 3.6.4 以后版本开始支持，在useCopy字段为0的场景下，可以设定writeMode字段采用其他写入方式，参数取值如下： • 0：采用BATCH INSERT方式写入数据。 • 1（默认值）：采用COPY API写入数据。 • 2：采用BATCH UPSERT方式写入数据。upsert含义见文档

类型映射

实时计算字段类型	分析型数据库PostgreSQL版字段类型
BOOLEAN	BOOLEAN
TINYINT	SAMLLINT
SAMLLINT	SAMLLINT
INT	INT
BIGINT	BIGINT
DOUBLE	DOUBLE PRECISION
VARCHAR	TEXT
DATETIME	TIMESTAMP
DATE	DATE
FLOAT	REAL
DECIMAL	DOUBLE PRECISION
TIME	TIME
TIMESTAMP	TIMESTAMP

启动导入任务

1. 在开发作业页面的右下角确认当前作业版本为3.6.0及以上，如果不符请点击切换版本。

2. 完成作业开发后，依次点击**保存**、**上线**，即可上线该任务。

3. 点击**运维**，在运维页面点击目标项目操作栏中的**启动**即可开始导入。

连接对应ADBPG实例，发现数据已经写入了目标表。

版本变更记录

blink 3.6.4版本：

- 默认写入方式由BATCH INSERT变为BATCH COPY，以提高写入性能。
- 增加writeMode字段
  - 在3.6.4版本以后，如果不设置useCopy字段为1，则writeMode字段无论为何值均采用BATCH COPY方式写入。
  - 例如：采用BATCH INSERT方式写入，需要设定useCopy=0, writeMode=0; 采用BATCH UPSERT方式写入，需要设置useCopy=0,writeMode=2。
  - 在以后的迭代中，会逐步放弃useCopy字段，请尽量采用writeMode字段配置写入方式。
- conflictMode字段增加upsert取值，通过insert on conflict的方式处理主键冲突。

blink 3.6.0 版本：

- blink 3.6.0版本开始支持6.0版本ADB PG版结果表

4.Dataworks数据集成

数据集成是阿里巴巴集团提供的数据同步平台。该平台具备可跨异构数据存储系统、可靠、安全、低成本、可弹性扩展等特点，可为20多种数据源提供不同网络环境下的离线（全量/增量）数据进出通道。详情请参见支持的数据源与读写插件。

应用场景

- AnalyticDB for PostgreSQL 可以通过数据集成的同步任务将数据同步到到其他的数据源中（AnalyticDB for PostgreSQL数据导出），并对数据进行相应的处理。
- 可以通过数据集成的同步任务将处理好的其他数据源数据同步到 AnalyticDB for PostgreSQL（AnalyticDB for PostgreSQL数据导入）。

无论是哪种应用场景，都可以通过DataWorks的数据集成功能完成数据的同步过程，详细的操作步骤（包括创建数据集成任务、数据源配置、作业配置、白名单配置等），请参考DataWorks文档中的使用指南-->数据集成一栏。文章中余下部分会介绍AnalyticDB for PostgreSQL的数据导入导出操作详细步骤。

准备工作

数据集成任务准备

1. 开通准备阿里云账号
2. 开通MaxCompute，自动产生一个默认的MaxCompute数据源，并使用主账号登录DataWorks
3. 创建工作空间。您可在工作空间中协作完成工作流，共同维护数据和任务等，因此使用DataWorks前需要先创建工作空间。

> 说明 如果您想通过子账号创建数据集成任务，可以赋予其相应的权限。详情请参见准备RAM用户

AnalyticDB for PostgreSQL 准备

1. 进行数据导入操作前，请通过 PostgreSQL 客户端创建好 AnalyticDB for PostgreSQL中需要迁入数据的目标数据库和表。
2. 对于数据导出，请登录AnalyticDB for PostgreSQL的管理控制台进行IP白名单设置，详情请参见 添加白名单

数据导入

源端的数据源需要在DataWorks管理控制台进行添加，数据源添加的详细步骤请参考配置AnalyticDB for PostgreSQL数据源

配置同步任务：

配置好数据源后，就可以配置同步任务，完成数据源数据到AnalyticDB for PostgreSQL的数据导入。配置同步任务有两种模式：**向导模式和脚本模式**。

- 向导模式。通过向导模式配置数据集成任务，需要依次完成以下几步：
  i. 新建数据同步节点；
  ii. 选择数据来源；
  iii. 选择数据去向（这里的数据去向一定是AnalyticDB for PostgreSQL）；
  iv. 配置字段的映射关系；

v. 配置作业速率上限、脏数据检查规则等信息；

vi. 配置调度属性。

> 说明 具体操作步骤请参考DataWorks通过向导模式配置任务

- 脚本模式。通过脚本模式配置数据集成任务，需要依次完成以下几步：
  i. 新建数据同步节点；
  ii. 导入模板；
  iii. 配置同步任务的读取端；
  iv. 配置同步任务的写入端（这里写入端一定是AnalyticDB for PostgreSQL）；
  v. 配置字段的映射关系；
  vi. 配置作业速率上限、脏数据检查规则等信息；
  vii. 配置调度属性。

> 说明 具体操作步骤请参考DataWorks通过脚本模式配置任务

数据导出

数据导出的步骤和数据导入的步骤一样，区别是在数据导出中，数据源配置需要配置为AnalyticDB for PostgreSQL（参见配置AnalyticDB for PostgreSQL数据源），而目的端可以配置为其他的数据源类型。

参考信息

更多数据集成详细信息请参考DataWorks文档

5.COPY命令导入或导出本地数据

您可以直接使用 `\COPY` 命令，将本地的文本文件数据导入 AnalyticDB for PostgreSQL数据库实例，或者将数据从AnalyticDB for PostgreSQL中导出到本地文件。这里本地的文本文件要求是格式化的，如通过逗号、分号或特有符号作为分割符号的文件。

> **注意**
>
> - 由于 `\COPY` 命令需要通过 Master 节点进行串行数据写入处理，因此无法实现并行写入大批量数据。如果要进行大量数据的并行写入，请使用基于 OSS 的数据导入方式。
> - `\COPY` 命令是 psql 的操作指令，如果您使用的不是 `\COPY` ，而是数据库指令 `COPY` ，则需要注意只支持 STDIN，不支持 file，因为“根用户”并没有 superuser 权限，不可以进行 file 文件操作。

`\COPY` 导入数据的操作命令参考如下：

```
\COPY table [(column [, ...])] FROM {'file' | STDIN}
[ [WITH]
[OIDS]
[HEADER]
[DELIMITER [ AS ] 'delimiter']
[NULL [ AS ] 'null string']
[ESCAPE [ AS ] 'escape' | 'OFF']
[NEWLINE [ AS ] 'LF' | 'CR' | 'CRLF']
[CSV [QUOTE [ AS ] 'quote']
[FORCE NOT NULL column [, ...]]
[FILL MISSING FIELDS]
[[LOG ERRORS [INTO error_table] [KEEP]
SEGMENT REJECT LIMIT count [ROWS | PERCENT] ]
```

`\COPY` 导出数据的操作命令参考如下：

```
\COPY {table [(column [, ...])] | (query)} TO {'file' | STDOUT}
[ [WITH]
[OIDS]
[HEADER]
[DELIMITER [ AS ] 'delimiter']
[NULL [ AS ] 'null string']
[ESCAPE [ AS ] 'escape' | 'OFF']
[CSV [QUOTE [ AS ] 'quote']
[FORCE QUOTE column [, ...]] ]
[IGNORE EXTERNAL PARTITIONS ]
```

注意

- 云数据库 AnalyticDB for PostgreSQL 还支持用户使用 JDBC 执行 COPY 语句，JDBC 中封装了 CopyIn 方法，详细用法请参见文档Interface CopyIn。
- COPY 命令使用方法请参见文档COPY。

6.OSS外表高速导入或导出OSS数据

OSS 为阿里云对象存储服务，AnalyticDB for PostgreSQL 支持通过 OSS 外部表（即 gpossext 功能），将数据并行从 OSS云存储 导入或导出到 OSS云存储，并支持通过 gzip 进行 OSS 外部表文件压缩，大量节省存储空间及成本。

目前的 gpossext 支持读写text/csv格式的文件或者gzip 压缩格式的 text/csv 文件。

本文内容包括：

操作说明

通过 AnalyticDB for PostgreSQL 使用 OSS 外部表，主要涉及以下操作。

- 创建OSS外部表插件（oss_ext）
- 并行导入数据
- 并行导出数据
- 创建 OSS 外部表语法

创建 OSS 外部表插件（oss_ext）

使用 OSS 外部表时，需要在 AnalyticDB for PostgreSQL 中先创建 OSS 外部表插件（每个数据库需要单独创建）。

- 创建命令为：`CREATE EXTENSION IF NOT EXISTS oss_ext;`
- 删除命令为：`DROP EXTENSION IF EXISTS oss_ext;`

并行导入数据

导入数据时，请执行如下步骤：

1. 将数据均匀分散存储在多个 OSS 文件中。

   > **注意**
   >
   > AnalyticDB for PostgreSQL的每个数据分区（segment）将按轮询方式并行对OSS上的数据文件进行读取，文件的数目建议为 数据节点数（Segment 个数）的整数倍，从而提升读取效率。

2. 在 AnalyticDB for PostgreSQL 中，创建 READABLE 外部表。
3. 执行如下操作，并行导入数据。

```
INSERT INTO <目标表> SELECT * FROM <外部表>
```

并行导出数据

导出数据时，请执行如下步骤：

1. 在 AnalyticDB for PostgreSQL 中，创建 WRITABLE 外部表。
2. 执行如下操作，并行把数据导出到 OSS 中。

```
INSERT INTO <外部表> SELECT * FROM <源表>
```

创建 OSS 外部表语法

创建 OSS 外部表语法，请执行如下命令：

```
CREATE [READABLE] EXTERNAL TABLE tablename
( columnname datatype [, ...] | LIKE othertable )
LOCATION ('ossprotocol')
FORMAT 'TEXT'
        [( [HEADER]
          [DELIMITER [AS] 'delimiter' | 'OFF']
          [NULL [AS] 'null string']
          [ESCAPE [AS] 'escape' | 'OFF']
          [NEWLINE [ AS ] 'LF' | 'CR' | 'CRLF']
          [FILL MISSING FIELDS] )]
       | 'CSV'
        [( [HEADER]
          [QUOTE [AS] 'quote']
          [DELIMITER [AS] 'delimiter']
          [NULL [AS] 'null string']
          [FORCE NOT NULL column [, ...]]
          [ESCAPE [AS] 'escape']
          [NEWLINE [ AS ] 'LF' | 'CR' | 'CRLF']
          [FILL MISSING FIELDS] )]
[ ENCODING 'encoding' ]
[ [LOG ERRORS [INTO error_table]] SEGMENT REJECT LIMIT count
    [ROWS | PERCENT] ]
CREATE WRITABLE EXTERNAL TABLE table_name
( column_name data_type [, ...] | LIKE other_table )
LOCATION ('ossprotocol')
FORMAT 'TEXT'
          [( [DELIMITER [AS] 'delimiter']
          [NULL [AS] 'null string']
          [ESCAPE [AS] 'escape' | 'OFF'] )]
       | 'CSV'
          [([QUOTE [AS] 'quote']
          [DELIMITER [AS] 'delimiter']
```

```
            [DELIMITER [AS] 'delimiter' ]
            [NULL [AS] 'null string']
            [FORCE QUOTE column [, ...]] ]
            [ESCAPE [AS] 'escape'] )]
[ ENCODING 'encoding' ]
[ DISTRIBUTED BY (column, [ ... ] ) | DISTRIBUTED RANDOMLY ]
ossprotocol:
  oss://oss_endpoint prefix=prefix_name
   id=userossid key=userosskey bucket=ossbucket compressiontype=[none|gzip] async=[true|false]
ossprotocol:
  oss://oss_endpoint dir=[folder/[folder/]...]/file_name
   id=userossid key=userosskey bucket=ossbucket compressiontype=[none|gzip] async=[true|false]
ossprotocol:
  oss://oss_endpoint filepath=[folder/[folder/]...]/file_name
   id=userossid key=userosskey bucket=ossbucket compressiontype=[none|gzip] async=[true|false]
```

参数释义

该部分介绍各操作中用到的参数定义，涉及到参数包括：

- 常用参数
- 导入模式参数
- 导出模式参数
- 其他通用参数

常用参数

- 协议和 endpoint：格式为“协议名://oss_endpoint”，其中协议名为 oss，oss_endpoint 为 OSS 对应区域的域名。

  > **注意**
  >
  > 如果是从阿里云的主机访问数据库，应该使用内网域名（即带有“internal”的域名），避免产生公网流量。

- id：OSS 账号的 ID。
- key：OSS 账号的 key。
- bucket：指定数据文件所在的 bucket，需要通过 OSS 预先创建。
- prefix：指定数据文件对应路径名的前缀，不支持正则表达式，仅是匹配前缀，且与 filepath、dir 互斥，三者只能设置其中一个。

- 如果创建的是用于数据导入的 READABLE 外部表，则在导入时含有这一前缀的所有 OSS 文件都会被导入。
  - 如果指定 prefix=test/filename，以下文件都会被导入：
    - test/filename
    - test/filenamexxx
    - test/filename/aa
    - test/filenameyyy/aa
    - test/filenameyyy/bb/aa
  - 如果指定 prefix=test/filename/，只有以下文件会被导入（上面列的其他文件不会被导入）：
    - test/filename/aa
- 如果创建的是用于数据导出的 WRITABLE 外部表，在导出数据时，将根据该前缀自动生成一个唯一的文件名来给导出文件命名。

- dir：OSS 中的虚拟文件夹路径，与 prefix、filepath 互斥，三者只能设置其中一个。
  - 文件夹路径需要以“/”结尾，如 `test/mydir/` 。
  - 在导入数据时，使用此参数创建外部表，会导入指定虚拟目录下的所有文件，但不包括它子目录和子目录下的文件。与 filepath 不同，dir 下的文件没有命名要求。
  - 在导出数据时，使用此参数创建外部表，所有数据会导出到此目录下的多个文件中，输出文件名的形式为 `filename.x` ，x 为数字，但可能不是连续的。

- filepath：OSS 中包含路径的文件名称，与 prefix、dir 互斥，三者只能设置其中一个，并且这个参数只能在创建 READABLE 外部表时指定（即只支持在导入数据时使用）。
  - 该文件名称包含该路径，但不包含 bucket 名。
  - 在导入数据时，文件命名方式必须为 `filename` 或 `filename.x` ，x 要求从 1 开始，且是连续的。例如，如果指定 filepath = filename，而 OSS 中含有如下文件：

    ```
    filename
    filename.1
    filename.2
    filename.4,
    ```

    则将被导入的文件有 filename、filename.1 和 filename.2。而因为 filename.3 不存在，所以 filename.4 不会被导入。

导入模式参数

- async：是否启用异步模式导入数据。
  - 开启辅助线程从 OSS 导入数据，加速导入性能。
  - 默认情况下异步模式是打开的，如果需要关掉，可以使用参数 async = false 或 async = f。
  - 异步模式和普通模式比，会消耗更多的硬件资源。

- compressiontype：导入的文件的压缩格式。
  - 指定为 none（缺省值），说明导入的文件没经过压缩。
  - 指定为 gzip，则导入的格式为 gzip。目前仅支持 gzip 压缩格式。

- compressionlevel：设置写入 OSS 的文件的压缩等级，取值范围为 1 - 9，默认值为 6

导出模式参数

- oss_flush_block_size：单次刷出数据到 OSS 的 buffer 大小，默认为 32 MB，可选范围是 1 到 128 MB。
- oss_file_max_size：设置写入到 OSS 的最大文件大小，超出之后会切换到另一个文件继续写。默认为 1024 MB，可选范围是 8 MB 到 4000 MB。
- num_parallel_worker： 设置写入 OSS 的压缩数据的并行压缩线程个数，取值范围为 1 - 8，默认值为 3。
- compressiontype：导出文件的压缩格式。
  - 指定为 none（缺省值），说明导出的文件没经过压缩。
  - 指定为 gzip，则导出的格式为 gzip。目前仅支持 gzip 压缩格式。

另外，针对导出模式，有如下注意事项：

- WRITABLE 是导出模式外部表的关键字，创建外部表时需要明确指明。
- 导出模式目前只支持 prefix 和 dir 参数模式，不支持 filepath。
- 导出模式的 DISTRIBUTED BY 子句可以使数据节点（Segment）按指定的分布键将数据写入 OSS。

其他通用参数

针对导入模式和导出模式，还有下列容错相关的参数：

- oss_connect_timeout：设置链接超时，单位为秒，默认是 10 秒。
- oss_dns_cache_timeout：设置 DNS 超时，单位为秒，默认是 60 秒。
- oss_speed_limit：设置能容忍的最小速率，默认是 1024，即 1 KB。
- oss_speed_time：设置能容忍的最长时间，默认是 15 秒。

上述参数如果使用默认值，则如果连续 15 秒的传输速率小于 1 KB，就会触发超时。详细描述请参见 OSS SDK 错误处理。

其他参数兼容 Greenplum EXTERNAL TABLE 的原有语法，具体语法解释请参见 Greenplum 外部表语法官方文档。这部分参数主要有：

- FORMAT：支持文件格式，支持 text、csv 等。
- ENCODING：文件中数据的编码格式，如 utf8。
- LOG ERRORS：指定该子句可以忽略掉导入中出错的数据，将这些数据写入error_table，并可以使用count参数指定报错的阈值。

说明

- 通过 `LOG ERRORS` 将错误行信息记录到内部关联文件。

```
create readable external table ossexample
    (date text, time text, open float, high float,
    low float, volume int)
    location('oss://oss-cn-hangzhou.aliyuncs.com
    prefix=osstest/example id=XXX
    key=XXX bucket=testbucket compressiontype=gzip')
    FORMAT 'csv' (QUOTE "" DELIMITER E'\t')
    ENCODING 'utf8'
    LOG ERRORS SEGMENT REJECT LIMIT 5;
```

- 通过函数 `gp_read_error_log('external_table_name')` 可以读取错误行信息。

```
select * from gp_read_error_log('external_table_name');
```

- 内部文件随外表删除而删除，也可以通过函数 `gp_truncate_error_log('external_table_name')` 删除。

```
select gp_truncate_error_log('external_table_name');
```

- 同时4.3版本也支持通过 `LOG ERRORS INTO error_table` 语法指定错误表，其他版本不再支持。

```
create readable external table ossexample
    (date text, time text, open float, high float,
    low float, volume int)
    location('oss://oss-cn-hangzhou.aliyuncs.com
    prefix=osstest/example id=XXX
    key=XXX bucket=testbucket compressiontype=gzip')
    FORMAT 'csv' (QUOTE "" DELIMITER E'\t')
    ENCODING 'utf8'
    LOG ERRORS INTO my_error_rows SEGMENT REJECT LIMIT 5;
```

使用示例

```
# 创建 OSS 导入外部表
create readable external table ossexample
    (date text, time text, open float, high float,
    low float, volume int)
    location('oss://oss-cn-hangzhou.aliyuncs.com
    prefix=osstest/example id=XXX
```

```
        key=XXX bucket=testbucket compressiontype=gzip')
        FORMAT 'csv' (QUOTE '"' DELIMITER E'\t')
        ENCODING 'utf8'
        LOG ERRORS SEGMENT REJECT LIMIT 5;
create readable external table ossexample
        (date text, time text, open float, high float,
        low float, volume int)
        location('oss://oss-cn-hangzhou.aliyuncs.com
        dir=osstest/ id=XXX
        key=XXX bucket=testbucket')
        FORMAT 'csv'
        LOG ERRORS SEGMENT REJECT LIMIT 5;
create readable external table ossexample
        (date text, time text, open float, high float,
        low float, volume int)
        location('oss://oss-cn-hangzhou.aliyuncs.com
        filepath=osstest/example.csv id=XXX
        key=XXX bucket=testbucket')
        FORMAT 'csv'
        LOG ERRORS SEGMENT REJECT LIMIT 5;
# 创建 OSS 导出外部表
create WRITABLE external table ossexample_exp
        (date text, time text, open float, high float,
        low float, volume int)
        location('oss://oss-cn-hangzhou.aliyuncs.com
        prefix=osstest/exp/outfromhdb id=XXX
        key=XXX bucket=testbucket') FORMAT 'csv'
        DISTRIBUTED BY (date);
create WRITABLE external table ossexample_exp
        (date text, time text, open float, high float,
        low float, volume int)
        location('oss://oss-cn-hangzhou.aliyuncs.com
        dir=osstest/exp/ id=XXX
        key=XXX bucket=testbucket') FORMAT 'csv'
        DISTRIBUTED BY (date);
# 创建堆表，数据就装载到这张表中
create table example
        (date text, time text, open float,
         high float, low float, volume int)
         DISTRIBUTED BY (date);
# 数据并行地从 ossexample 装载到 example 中
```

```
insert into example select * from ossexample;
# 数据并行地从 example 导出到 oss
insert into ossexample_exp select * from example;
# 从下面的执行计划中可以看出，每个 Segment 都会参与工作。
# 每个 Segment 从 OSS 并行拉取数据，然后通过 Redistribution Motion 这个执行节点将拿到的数据 HASH 计算后分发给对应的 Segment，接受数据的 Segment 通过 Insert 执行节点进行入库。
explain insert into example select * from ossexample;
                                QUERY PLAN
-----------------------------------------------------------------------------------------------
 Insert (slice0; segments: 4)  (rows=250000 width=92)
   ->  Redistribute Motion 4:4  (slice1; segments: 4)  (cost=0.00..11000.00 rows=250000 width=92)
         Hash Key: ossexample.date
         ->  External Scan on ossexample  (cost=0.00..11000.00 rows=250000 width=92)
(4 rows)
# 从下面的查询计划可以看到，Segment 把本地数据直接导出到 OSS ,没有进行数据重分布
explain insert into ossexample_exp select * from example;
                    QUERY PLAN
-------------------------------------------------------------
 Insert (slice0; segments: 3)  (rows=1 width=92)
   ->  Seq Scan on example  (cost=0.00..0.00 rows=1 width=92)
(2 rows)
```

注意事项

- 创建和使用外部表的语法，除了 location 相关的参数，其余部分和 Greenplum 相同。
- 数据导入的性能和 AnalyticDB for PostgreSQL 集群的资源（CPU、IO、内存、网络等）相关，也和 OSS 相关。为了获取最大的导入性能，建议在创建表时，使用列式存储 + 压缩功能。例如，指定子句“WITH (APPENDONLY=true, ORIENTATION=column, COMPRESSTYPE=zlib, COMPRESSLEVEL=5, BLOCKSIZE=1048576)”，详情请参见 Greenplum Database 表创建语法官方文档。
- 为了保证数据导入的性能，ossendpoint Region 需要匹配 AnalyticDB for PostgreSQL 云上所在 Region，建议 OSS AnalyticDB for PostgreSQL 在同一个 Region 内以获得最好的性能。

TEXT/CSV 格式说明

下列几个参数可以在外表 DDL 参数中指定，用于规定读写 OSS 的文件格式：

- TEXT/CSV 行分割符号是 ‘\n’，也就是换行符。
- DELIMITER 用于定义列的分割符：
  - 当用户数据中包括 DELIMITER 时，则需要和 QUOTE 参数一同使用。
  - 推荐的列分割符有 ‘,’、‘\t‘ 、‘|’ 或一些不常出现的字符。
- QUOTE 以列为单位包裹有特殊字符的用户数据。
  - 用户包含有特殊字符的字符串会被 QUOTE 包裹，用于区分用户数据和控制字符。
  - 如果不必要，例如整数，基于优化效率的考虑，不必使用 QUOTE 包裹数据。

- QUOTE 不能和 DELIMITER 相同，默认 QUOTE 是双引号。
- 当用户数据中包含了 QUOTE 字符，则需要使用转义字符 ESCAPE 加以区分。

- ESCAPE 特殊字符转义
  - 转义字符出现在需要转义的特殊字符前，表示它不是一个特殊字符。
  - ESCAPE 默认和 QUOTE 相同，也就是双引号。
  - 也支持设置成 '\'(MySQL 默认的转义字符)或别的字符。

典型的 TEXT/CSV 默认控制字符

控制字符 \ 格式	TEXT	CSV
DELIMITER（列分割符）	\t （tab）	，（comma）
QUOTE（摘引）	" （double-quote）	"（double-quote）
ESCAPE（转义）	（不适用）	和 QUOTE 相同
NULL（空值）	\N （backslash-N）	（无引号的空字符串）

> **说明**
>
> 所有的控制字符都必须是单字节字符。

SDK 错误处理

当导入或导出操作出错时，错误日志可能会出现如下信息：

- code：出错请求的 HTTP 状态码。
- error_code：OSS 的错误码。
- error_msg：OSS 的错误信息。
- req_id：标识该次请求的 UUID。当您无法解决问题时，可以凭 req_id 来请求 OSS 开发工程师的帮助。

详情请参见OSS API 错误响应，超时相关的错误可以使用 oss_ext 相关参数处理。

常见问题

如果导入过慢，请参见上面“注意事项”中关于导入性能的描述。

参考文档

- OSS endpoint 信息
- OSS help 页面
- OSS SDK 错误处理
- OSS API 错误响应
- Greenplum Database 外部表语法官方文档
- Greenplum Database 表创建语法官方文档

7.基于Client SDK数据写入

AnalyticDB PostgreSQL版Client SDK旨在通过 API 方式提供高性能COPY数据到AnalyticDB PostgreSQL版的方式。

AnalyticDB PostgreSQL版Client SDK通过 API 形式旨在为用户提供高性能写入数据到AnalyticDB PostgreSQL版的方式，支持用户定制化开发或对接写入程序。通过 SDK 开发写入程序，可简化在AnalyticDB PostgreSQL版中写入数据的流程，无需担心连接池、缓存等问题，相比较直接COPY/INSERT写入，通过并行化等内部机制有几倍性能提升

> 说明 AnalyticDB PostgreSQL版Client SDK主要职责是将您传入的数据高效地写入，不负责原始数据的读取、处理等工作。

Maven repositories

您可以通过Maven管理配置新SDK的版本。Maven的配置信息如下：

```
<dependency>
 <groupId>com.alibaba.cloud.analyticdb</groupId>
 <artifactId>adb4pgclient</artifactId>
 <version>1.0.0</version>
</dependency>
```

> 说明
>
> - AnalyticDB PostgreSQL版Client SDK本身依赖*druid(1.1.17)*、*postgresql(jdbc 42.2.5)*、*commons-lang3(3.4)*、*slf4j-api(1.7.24)*、*slf4j-log4j12(1.7.24)*。
> - 如果在使用过程中出现版本冲突，请检查这几个包的版本并解决冲突。

接口列表

DatabaseConfig类

接口名	描述
setHost(String adbHost)	需要连接的AnalyticDB PostgreSQL版的连接地址。
setPort(int port)	需要连接的AnalyticDB PostgreSQL版的端口，默认为3432。
setDatabase(String database)	需要连接的AnalyticDB PostgreSQL版数据库名称。
setUser(String username)	需要连接的AnalyticDB PostgreSQL版使用的用户名。
setPassword(String pwd)	设置连接的AnalyticDB PostgreSQL版使用的密码。

接口名	描述
addTable(List<String> table, String schema)	需要写入的表名List，请按照表所属schema分别添加。该方法可调用多次，但在使用DatabaseConfig构造Adb4PGClient对象之后再调用不再生效。
setColumns(List<String> columns, String tableName, String schemaName)	需要插入表的字段名（若是全字段插入，columnList.add("*")即可, table列表中的所有表都需要设置字段名，不然检查不会通过
setInsertIgnore(boolean insertIgnore)	设置是否忽略发生主键冲突错误的数据行，要根据业务的使用场景进行判断，针对配置的所有表，默认为true
setEmptyAsNull(boolean emptyAsNull)	设置empty数据设置为null，默认false，针对配置的所有表
setParallelNumber(int parallelNumber)	设置写入ADB PG版时的并发线程数，默认4，针对配置的所有表，一般情况不建议修改
setLogger(Logger logger)	设置client中使用的logger对象，此处使用slf4j.Logger
setRetryTimes(int retryTimes)	设置commit时，写入ADB PG版出现异常时重试的次数，默认为3
setRetryIntervalTime(long retryIntervalTime)	设置重试间隔的时间，单位是ms，默认为 1000 ms
setCommitSize(long commitSize)	设置自动提交的数据量（单位Byte），默认为10MB，一般不建议设置

Row类

接口名称	描述
setColumn(int index, Object value)	设置Row字段列表的值，要求必须按照字段的顺序（此种方式，Row实例不可复用，每条数据必须单独的Row实例）
setColumnValues(List<Object> values)	直接将List格式数据行写入Row中
updateColumn(int index, Object value)	更新Row字段列表的值，注意更新的字段数据（此方法，Row实例可以复用，只需更新Row实例中的数据即可

Adb4pgClient 类

接口名称	描述
addRow(Row row, String tableName, String schemaName) / addRows(List<Row> rows, String tableName, String schemaName)	插入对应表的Row格式化的数据，即一条记录，数据会存储在sdk的缓冲区中，等待commit。如果数据量超过commitSize会在addRow/addRows的时候做一次自动commit，然后将最新的数据add进来；如果在自动commit失败的时候失败，调用方需要处理此异常，并且会在异常中得到失败的数据list

接口名称	描述
addMap(Map<String, String> dataMap,String tableName, String schemaName) / addMaps(List<Map<String, String>> dataMaps, String tableName, String schemaName)	对应于addRow，支持map格式数据的写入，如果数据量满了会在addMap/addMaps的时候做一次自动commit，然后将最新的数据add进来；如果在自动commit失败的时候失败，调用方需要处理此异常，并且会在异常中得到失败的数据list
commit()	将缓存的数据进行提交，写入ADB PG版中，若commit失败，会把执行错误的语句放在异常中抛出，调用方需要对此异常进行处理
TableInfo getTableInfo(String tableName, String schemaName)	获取对应table的结构信息
List<ColumnInfo> getColumnInfo(String tableName, String schemaName)	获取对应table的字段列表信息，字段类是ColumnInfo，可以通过columnInfo.isNullable()获取该字段是否能为null
stop()	实例使用完之后，stop释放内部线程池及资源，如果内存中有数据未commit，则会抛Exception，若需要强行stop，请使用forceStop()
forceStop()	强行释放内部线程池及资源，会丢失掉缓存在内存中未commit的数据，一般不推荐使用
Connection getConnection() throws SQLException	从client连接池获取ADB PG Connection连接，调用方可以使用获得的Connection做非copy操作，使用方式和jdbc的连接使用方式一致。 说明 使用结束后一定要释放掉相应的资源（如ResultSet、Statement、Connection）

ColumnInfo类

接口名称	描述
boolean isNullable()	判断该字段是否能为null

错误码名	错误码值	描述

错误码名	错误码值	描述
COMMIT_ERROR_DATA_LIST	101	commit中某些数据出现异常，会返回异常的数据。 说明 通过e.getErrData()即可获得异常数据List<String>，此错误码在addMap(s)、addRow(s)和commit操作的时候都可能会发生，因此在这些操作的时候需要单独处理此错误码的异常
COMMIT_ERROR_OTHER	102	commit中的其他异常
ADD_DATA_ERROR	103	add数据过程中出现的异常
CREATE_CONNECTION_ERROR	104	创建连接出现异常
CLOSE_CONNECTION_ERROR	105	关闭连接出现异常
CONFIG_ERROR	106	配置DatabaseConfig出现配置错误
STOP_ERROR	107	停止实例时的报错
OTHER	999	默认异常错误码

代码示例

```
public class Adb4pgClientUsage {
    public void demo() {
        DatabaseConfig databaseConfig = new DatabaseConfig();
        // Should set your database real host or url
        databaseConfig.setHost("100.100.100.100");
        // Should set your database real port
        databaseConfig.setPort(8888);
        // 连接数据库的用户名
        databaseConfig.setUser("your user name");
        // 连接数据库的密码
        databaseConfig.setPassword("your password");
       // 需要连接的database
        databaseConfig.setDatabase("your database name");
```

```
        databaseConfig.setDatabase("your database name");
        // 设置需要写入的表名列表
        List<String> tables = new ArrayList<String>();
        tables.add("your table name 1");
        tables.add("your table name 2");

        // 不同schema下的表可分别addTable，但是一旦使用databseconfig 创造Client实例之后，table配置是不可修改的/
        // schema传入null, 则默认schema为public
        databaseConfig.addTable(tables, "table schema name");

        // 设置需要写入的表字段
        List<String> columns = new ArrayList<String>();
        columns.add("column1");
        columns.add("column2");
        // 如果是所有字段，字段列表使用 columns.add("*") 即可
        databaseConfig.setColumns(columns, "your table name 1", "table schema name");
        databaseConfig.setColumns(Collections.singletonList("*"),"your table name 2", "table schema name");



        // If the value of column is empty, set null
        databaseConfig.setEmptyAsNull(false);
        // 使用insert ignore into方式进行插入
        databaseConfig.setInsertIgnore(true);
        // commit时，写入ADB出现异常时重试的3次
        databaseConfig.setRetryTimes(3);
        // 重试间隔的时间为1s，单位是ms
        databaseConfig.setRetryIntervalTime(1000);
        // Initialize AdbClient，初始化实例之后，databaseConfig的配置信息不能再修改
        Adb4pgClient adbClient = new Adb4pgClient(databaseConfig);

        // 数据需要攒批，多次add，再commit，具体攒批数量见"注意事项"
        for (int i = 0; i < 10; i++) {
            // Add row(s) to buffer. One instance for one record
            Row row = new Row(columns.size());
            // Set column
            // the column index must be same as the sequence of columns
            // the column value can be any type, internally it will be formatted according to column type
            row.setColumn(0, i); // Number value
            row.setColumn(1, "string value"); // String value
```

```
            // 如果sql长度满了会在addRow或者addMap的时候会进行一次自动提交
            // 如果提交失败会返回AdbClientException异常，错误码为COMMIT_ERROR_DATA_LIST
            adbClient.addRow(row, "your table name 1", "table schema name");
        }

        Row row = new Row();
        row.setColumn(0, 10); // Number value
        row.setColumn(1, "2018-01-01 08:00:00"); // Date/Timestamp/Time value
        adbClient.addRow(row, "your table name 1", "table schema name");
        // Update column. Row实例可复用
        row.updateColumn(0, 11);
        row.updateColumn(1, "2018-01-02 08:00:00");
        adbClient.addRow(row, "your table name 1", "table schema name");

        // Add map(s) to buffer
        Map<String, String> rowMap = new HashMap<String, String>();
        rowMap.put("t1", "12");
        rowMap.put("t2", "string value");
        // 这边需要攒批的，最好多次add之后在进行commit
        adbClient.addMap(rowMap, "your table name 2", "table schema name");

        // Commit buffer to ADS
        // Buffer is cleaned after successfully commit to ADS
        try {
            adbClient.commit();
        } catch (Exception e) {
            // TODO: Handle exception here
        } finally {
            adbClient.stop();
        }
    }

}
```

注意事项

- ADB PG 版Client SDK是非线程安全的，所以如果多线程调用的情况，需要每个线程维护自己的Client对象

> 注意 强烈不建议多线程共用SDK实例，除了线程安全问题外，容易让Client成为写入性能的瓶颈。

- 数据必须在调用commit成功后才能认为是写入ADB PG版成功的。
- 针对Client抛出的异常，调用方要根据错误码的意义自行判断如何处理，如果是数据写入有问题，可以重复提交或者记录下有问题的数据后跳过。
- 很多时候写入线程并不是越多越好，因为业务程序会涉及到攒数据的场景，对内存的消耗是比较明显的，所以业务调用方一定要多多关注应用程序的GC情况。
- 数据攒批数量不要太小，如果太小，攒批写入意义就不大了，条件允许的情况下可以add 10000条进行一次commit。
- DatabaseConfig配置在实例化client对象成功之后是不能再修改的，所有配置项必须在client对象初始化之前完成配置。
- Client SDK目的是对写入（INSERT）提供性能优化，对于其他SQL操作，可以通过getConnection()获得JDBC连接，通过标准JDBC接口进行处理

8.DTS数据迁移及同步方案列表

8.1. RDS MySQL数据同步至AnalyticDB for PostgreSQL

数据传输服务DTS（Data Transmission Service）支持将RDS for MySQL同步至AnalyticDB for PostgreSQL。通过DTS提供的数据同步功能，可以轻松实现数据的流转，将企业数据集中分析。

前提条件

- RDS MySQL中待同步的数据表必须具备主键。
- 已创建目标云原生数据仓库AnalyticDB PostgreSQL实例，如未创建请参见创建云原生数据仓库AnalyticDB PostgreSQL实例。

注意事项

DTS在执行全量数据初始化时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据同步前评估源库和目标库的性能，同时建议您在业务低峰期执行数据同步（例如源库和目标库的CPU负载在30%以下）。

同步限制

- 同步对象仅支持数据表。
- 不支持BIT、VARBIT、GEOMETRY、ARRAY、UUID、TSQUERY、TSVECTOR、TXID_SNAPSHOT类型的数据同步。
- 暂不支持同步前缀索引，如果源库存在前缀索引可能导致数据同步失败。
- 在数据同步时，请勿对源库的同步对象使用gh-ost或pt-online-schema-change等类似工具执行在线DDL变更，否则会导致同步失败。

支持同步的SQL操作

- DML操作：INSERT、UPDATE、DELETE。
- DDL操作：ADD COLUMN、RENAME COLUMN。

> 说明 不支持CREATE TABLE操作，如果您需要将新增的表作为同步对象，则需要执行新增同步对象操作。

支持的同步架构

- 1对1单向同步。
- 1对多单向同步。
- 多对1单向同步。

术语及概念对应关系

MySQL	云原生数据仓库AnalyticDB PostgreSQL
Database	Schema
Table	Table

操作步骤

1. 购买数据同步作业，详情请参见购买流程。

   > 说明 购买时，选择源实例为MySQL，目标实例为AnalyticDB for PostgreSQL，并选择同步拓扑为单向同步。

2. 登录数据传输控制台。
3. 在左侧导航栏，单击数据同步。
4. 在同步作业列表页面顶部，选择同步的目标实例所属地域。

   

5. 定位至已购买的数据同步实例，单击配置同步链路。
6. 配置同步作业的源实例及目标实例信息。

   

类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，建议配置具有业务意义的名称（无唯一性要求），便于后续识别。
源实例信息	实例类型	选择RDS实例。
	实例地区	购买数据同步实例时选择的源实例地域信息，不可变更。
	实例ID	选择源RDS实例ID。
	数据库账号	填入RDS MySQL的数据库账号。 说明 当源RDS实例的数据库类型为MySQL 5.5或MySQL 5.6时，没有数据库账号和数据库密码的配置选项。
	数据库密码	填入该数据库账号对应的密码。
	连接方式	根据需求选择非加密连接或SSL安全连接。如果设置为SSL安全连接，您需要提前开启RDS实例的SSL加密功能，详情请参见设置SSL加密。 说明 目前仅中国内地及中国香港地域支持设置连接方式。
	实例类型	固定为AnalyticDB for PostgreSQL，无需设置。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。

类别	配置	说明
目标实例信息	实例ID	选择目标云原生数据仓库AnalyticDB PostgreSQL实例ID。
	数据库名称	填入同步目标表所属的数据库名称。
	数据库账号	填入云原生数据仓库AnalyticDB PostgreSQL的初始账号，详情请参见设置账号。 说明 您也可以填入具备RDS_SUPERUSER权限的账号，创建方法请参见用户权限管理。
	数据库密码	填入该数据库账号对应的密码。

7. 单击页面右下角的授权白名单并进入下一步。

   说明 此步骤会将DTS服务器的IP地址自动添加到RDS MySQL和云原生数据仓库AnalyticDB PostgreSQL的白名单中，用于保障DTS服务器能够正常连接源集群和目标实例。

8. 配置同步策略及同步对象。

类别	配置	说明
同步策略配置	同步初始化	默认情况下，您需要同时选中结构初始化和全量数据初始化。预检查完成后，DTS会将源实例中待同步对象的结构及数据在目标实例中初始化，作为后续增量同步数据的基线数据。
	目标已存在表的处理模式	○ 清空目标表的数据 在预检查阶段跳过同名对象存在性检查的检查项目。全量初始化之前将目标表的数据清空。适用于完成同步任务测试后的正式同步场景。 ○ 忽略报错并继续执行 在预检查阶段跳过同名对象存在性检查的检查项目。全量初始化时直接追加数据。适用于多张表同步到一张表的汇总同步场景。
	同步操作类型	根据业务需求选择需要同步的操作类型： ○ Insert ○ Update ○ Delete ○ AlterTable

类别	配置	说明
选择同步对象	无	在源库对象框中单击待同步的表，然后单击 图标将其移动至已选择对象框。 说明 ○ 同步对象的选择粒度为表。 ○ 如果需要目标表中的列名称与源表不同，需要使用DTS的字段映射功能，详情请参见设置同步对象在目标实例中的名称。

9. 设置待同步的表在云原生数据仓库AnalyticDB PostgreSQL中的主键列和分布列信息。

   说明 当您在上一步中选择了结构初始化才会出现该页面。关于主键列和分布列的详细说明，请参见表的约束定义和表分布键定义。

10. 上述配置完成后，单击页面右下角的预检查并启动。

    说明
    - 在数据同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动数据同步作业。
    - 如果预检查失败，单击具体检查项后的 图标，查看失败详情。根据提示修复后，重新进行预检查。

11. 在预检查对话框中显示预检查通过后，关闭预检查对话框，同步作业将正式开始。
12. 等待同步作业的链路初始化完成，直至处于同步中状态。

    您可以在数据同步页面，查看数据同步作业的状态。

8.2. POLARDB MySQL数据同步至AnalyticDB for PostgreSQL

数据传输服务DTS（Data Transmission Service）支持将POLARDB MySQL数据同步至AnalyticDB for PostgreSQL，帮助您轻松实现数据的流转，将企业数据集中分析。

前提条件

- PolarDB MySQL集群已开启Binlog，详情请参见如何开启Binlog。
- PolarDB MySQL集群中待同步的数据表必须具备主键。
- 已创建目标云原生数据仓库AnalyticDB PostgreSQL实例，详情请参见创建云原生数据仓库AnalyticDB PostgreSQL实例。

注意事项

- DTS在执行全量数据初始化时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据同步前评估源库和目标库的性能，同时建议您在业务低峰期执行数据同步（例如源库和目标库的CPU负载在30%以下）。
- 全量初始化过程中，并发INSERT会导致目标实例的表碎片，全量初始化完成后，目标实例的表空间比源集群的表空间大。

同步限制

- 同步对象仅支持数据表。
- 不支持BIT、VARBIT、GEOMETRY、ARRAY、UUID、TSQUERY、TSVECTOR、TXID_SNAPSHOT类型的数据同步。
- 暂不支持同步前缀索引，如果源库存在前缀索引可能导致数据同步失败。
- 在数据同步时，请勿对源库的同步对象使用gh-ost或pt-online-schema-change等类似工具执行在线DDL变更，否则会导致同步失败。

支持同步的SQL操作

- DML操作：INSERT、UPDATE、DELETE。
- DDL操作：ADD COLUMN、RENAME COLUMN。

> 说明 不支持CREATE TABLE操作，如果您需要将新增的表作为同步对象，则需要执行新增同步对象操作。

支持的同步架构

- 1对1单向同步。
- 1对多单向同步。
- 多对1单向同步。

术语对应关系

PolarDB MySQL	云原生数据仓库AnalyticDB PostgreSQL
Database	Schema
Table	Table

操作步骤

1. 购买数据同步作业，详情请参见购买流程。

   > 警告 由于DTS暂时不支持直接将PolarDB集群作为同步的数据源，本案例中将其作为通过专线接入的自建数据库来实现数据同步。在购买时，您需要选择源实例为MySQL，目标实例为AnalyticDB for PostgreSQL，并选择同步拓扑为单向同步。

2. 登录数据传输控制台。
3. 在左侧导航栏，单击数据同步。

4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。

5. 定位至已购买的数据同步实例，单击**配置同步链路**。
6. 配置同步通道的源实例及目标实例信息。

类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，建议配置具有业务意义的名称（无唯一性要求），便于后续识别。
源实例信息	实例类型	选择**通过专线/VPN网关/智能网关接入的自建数据库**。 说明 由于DTS暂时不支持直接将PolarDB MySQL作为源实例，此处将其作为通过专线接入的自建数据库来实现数据同步。
	实例地区	购买数据同步实例时选择的源PolarDB集群的地域信息，不可变更。
	对端专有网络	选择PolarDB集群所属的VPC ID。 您可以登录PolarDB控制台，单击目标集群ID，进入**基本信息**页面来获取。 
	数据库类型	固定为**MySQL**，不可变更。
	IP地址	配置PolarDB集群主地址的私网IP地址。 您可以在电脑中ping目标PolarDB集群的**主地址（私网）**来获取私网IP地址。 
	端口	填入PolarDB集群的服务端口，默认为**3306**。
	数据库账号	填入连接PolarDB集群的数据库账号。 说明 该账号需具备REPLICATION CLIENT、REPLICATION SLAVE、SHOW VIEW和所有同步对象的SELECT权限。
	数据库密码	填入该数据库账号对应的密码。
	实例类型	固定为**AnalyticDB for PostgreSQL**，无需设置。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。
	实例ID	选择云原生数据仓库AnalyticDB PostgreSQL实例ID。
	数据库名称	填入云原生数据仓库AnalyticDB PostgreSQL实例中，待同步的目标表所属的数据库名称。

目标实例信息	配置	说明
目标实例	数据库账号	填入云原生数据仓库AnalyticDB PostgreSQL的**初始账号**，详情请参见**设置账号**。 **说明** 您也可以填入具备RDS_SUPERUSER权限的账号，创建方法请参见**用户权限管理**。
	数据库密码	填入数据库账号对应的密码。

7. **单击页面右下角的授权白名单并进入下一步。**

   说明 此步骤会将DTS服务器的IP地址自动添加PolarDB MySQL和云原生数据仓库AnalyticDB PostgreSQL的白名单中，用于保障DTS服务器能够正常连接源集群和目标实例。

8. **配置同步策略及同步对象。**

类别	配置	说明
同步策略配置	同步初始化	默认情况下，您需要同时选中**结构初始化**和**全量数据初始化**。预检查完成后，DTS会将源实例中待同步对象的结构及数据在目标实例中初始化，作为后续增量同步数据的基线数据。
	目标已存在表的处理模式	○ **清空目标表的数据** 在预检查阶段跳过**同名对象存在性检查**的检查项目。全量初始化之前将目标表的数据清空。适用于完成同步任务测试后的正式同步场景。 ○ **忽略报错并继续执行** 在预检查阶段跳过**同名对象存在性检查**的检查项目。全量初始化时直接追加数据。适用于多张表同步到一张表的汇总同步场景。
	同步操作类型	根据业务需求选择需要同步的操作类型： ○ **Insert** ○ **Update** ○ **Delete** ○ **AlterTable**

类别	配置	说明
选择同步对象	无	在源库对象框中单击待同步的表，然后单击 图标将其移动至已选择对象框。 说明 ○ 同步对象的选择粒度为表。 ○ 如果需要目标表中的列名称与源表不同，需要使用DTS的字段映射功能，详情请参见设置同步对象在目标实例中的名称。

9. 设置待同步的表在云原生数据仓库AnalyticDB PostgreSQL中的主键列和分布列信息。

   说明 当您在上一步中选择了结构初始化才会出现该页面。关于主键列和分布列的详细说明，请参见表的约束定义和表分布键定义。

10. 上述配置完成后，单击页面右下角的预检查并启动。

    说明
    - 在数据同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动数据同步作业。
    - 如果预检查失败，单击具体检查项后的 图标，查看失败详情。根据提示修复后，重新进行预检查。

11. 在预检查对话框中显示预检查通过后，关闭预检查对话框，同步作业将正式开始。
12. 等待同步作业的链路初始化完成，直至处于同步中状态。

    您可以在数据同步页面，查看数据同步作业的状态。

8.3. 自建MySQL同步至AnalyticDB for PostgreSQL

数据传输服务DTS（Data Transmission Service）支持将ECS上的自建MySQL同步至AnalyticDB for PostgreSQL，帮助您轻松实现数据的流转，将企业数据集中分析。

前提条件

- ECS上的自建MySQL数据库版本为5.1、5.5、5.6、5.7或8.0版本。
- 源库必须开启binlog，同时建议创建一个账号用于数据同步，详情请参见为自建MySQL创建账号并设置binlog。

> 说明 该账号需具备REPLICATION CLIENT、REPLICATION SLAVE、SHOW VIEW和所有同步对象的SELECT权限。

- 源库中待同步的数据表必须具备主键。
- 已创建目标云原生数据仓库AnalyticDB PostgreSQL实例，如未创建请参见创建云原生数据仓库AnalyticDB PostgreSQL实例。

注意事项

DTS在执行全量数据初始化时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据同步前评估源库和目标库的性能，同时建议您在业务低峰期执行数据同步（例如源库和目标库的CPU负载在30%以下）。

同步限制

- 同步对象仅支持数据表。
- 不支持BIT、VARBIT、GEOMETRY、ARRAY、UUID、TSQUERY、TSVECTOR、TXID_SNAPSHOT类型的数据同步。
- 暂不支持同步前缀索引，如果源库存在前缀索引可能导致数据同步失败。
- 在数据同步时，请勿对源库的同步对象使用gh-ost或pt-online-schema-change等类似工具执行在线DDL变更，否则会导致同步失败。

支持同步的SQL操作

- DML操作：INSERT、UPDATE、DELETE。
- DDL操作：ADD COLUMN、RENAME COLUMN。

> 说明 不支持CREATE TABLE操作，如果您需要将新增的表作为同步对象，则需要执行新增同步对象操作。

支持的同步架构

- 1对1单向同步。
- 1对多单向同步。
- 多对1单向同步。

术语及概念对应关系

MySQL	云原生数据仓库AnalyticDB PostgreSQL
Database	Schema
Table	Table

操作步骤

1. 购买数据同步作业，详情请参见购买流程。

> **说明** 购买时，选择源实例为MySQL，目标实例为AnalyticDB for PostgreSQL，并选择同步拓扑为单向同步。

2. 登录数据传输控制台。
3. 在左侧导航栏，单击**数据同步**。
4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。

5. 定位至已购买的数据同步实例，单击**配置同步链路**。
6. 配置同步作业的源实例及目标实例信息。

类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，建议配置具有业务意义的名称（无唯一性要求），便于后续识别。
源实例信息	实例类型	选择ECS上的自建数据库。
	实例地区	购买数据同步实例时选择的源实例地域信息，不可变更。
	实例ID	选择作为自建MySQL所属的ECS实例ID。
	数据库类型	固定为MySQL，不可变更。
	端口	填入自建数据库的服务端口，默认为3306。
	数据库账号	填入ECS上的自建MySQL的数据库账号。 说明 该账号需具备REPLICATION CLIENT、REPLICATION SLAVE、SHOW VIEW和所有同步对象的SELECT权限。
	数据库密码	填入该数据库账号对应的密码。
目标实例信息	实例类型	固定为AnalyticDB for PostgreSQL，无需设置。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。
	实例ID	选择目标云原生数据仓库AnalyticDB PostgreSQL实例ID。
	数据库名称	填入同步目标表所属的数据库名称。

类别	配置	说明
	数据库账号	填入云原生数据仓库AnalyticDB PostgreSQL的**初始账号**，详情请参见**设置账号**。 **说明** 您也可以填入具备RDS_SUPERUSER权限的账号，创建方法请参见**用户权限管理**。
	数据库密码	填入该数据库账号对应的密码。

7. **单击页面右下角的授权白名单并进入下一步。**

   说明 此步骤会将DTS服务器的IP地址自动添加到ECS实例的内网入方向规则和云原生数据仓库AnalyticDB PostgreSQL的白名单中，用于保障DTS服务器能够正常连接源和目标实例。

8. **配置同步策略及同步对象。**

类别	配置	说明
同步策略配置	同步初始化	默认情况下，您需要同时选中**结构初始化**和**全量数据初始化**。预检查完成后，DTS会将源实例中待同步对象的结构及数据在目标实例中初始化，作为后续增量同步数据的基线数据。
	目标已存在表的处理模式	○ **清空目标表的数据** 在预检查阶段跳过**同名对象存在性检查**的检查项目。全量初始化之前将目标表的数据清空。适用于完成同步任务测试后的正式同步场景。 ○ **忽略报错并继续执行** 在预检查阶段跳过**同名对象存在性检查**的检查项目。全量初始化时直接追加数据。适用于多张表同步到一张表的汇总同步场景。
	同步操作类型	根据业务需求选择需要同步的操作类型： ○ **Insert** ○ **Update** ○ **Delete** ○ **AlterTable**

类别	配置	说明
选择同步对象	无	在源库对象框中单击待同步的表，然后单击 图标将其移动至已选择对象框。 说明 ○ 同步对象的选择粒度为表。 ○ 如果需要目标表中的列名称与源表不同，需要使用DTS的字段映射功能，详情请参见设置同步对象在目标实例中的名称。

9. 设置待同步的表在云原生数据仓库AnalyticDB PostgreSQL中的主键列和分布列信息。

   说明 当您在上一步中选择了结构初始化才会出现该页面。关于主键列和分布列的详细说明，请参见表的约束定义和表分布键定义。

10. 上述配置完成后，单击页面右下角的预检查并启动。

    说明
    - 在数据同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动数据同步作业。
    - 如果预检查失败，单击具体检查项后的 图标，查看失败详情。根据提示修复后，重新进行预检查。

11. 在预检查对话框中显示预检查通过后，关闭预检查对话框，同步作业将正式开始。
12. 等待同步作业的链路初始化完成，直至处于同步中状态。

    您可以在数据同步页面，查看数据同步作业的状态。

8.4. 通过专线/VPN网关/智能网关接入的自建MySQL同步至AnalyticDB for PostgreSQL

数据传输服务DTS（Data Transmission Service）支持将通过专线/VPN网关/智能网关接入的自建MySQL同步至AnalyticDB for PostgreSQL，帮助您轻松实现数据的流转，将企业数据集中分析。

前提条件

- 自建MySQL数据库版本为5.1、5.5、5.6、5.7或8.0版本。
- 源库中待同步的数据表必须具备主键。
- 源库必须开启binlog，同时建议您创建一个账号用于数据同步，详情请参见为自建MySQL创建账号并设置binlog。

> 说明 该账号需具备REPLICATION CLIENT、REPLICATION SLAVE、SHOW VIEW和所有同步对象的SELECT权限。

- 自建数据库所属的本地网络已通过专线、VPN网关或智能接入网关的方式接入至阿里云专有网络，并且已经配置DTS与专线、VPN网关或智能接入网关间的路由。

> 说明 相关接入方案请参见本地IDC接入至阿里云方案概览，本文不做详细介绍。

- 已创建目标云原生数据仓库AnalyticDB PostgreSQL实例，如未创建请参见创建云原生数据仓库AnalyticDB PostgreSQL实例。

注意事项

DTS在执行全量数据初始化时将占用源库和目标库一定的读写资源，可能会导致数据库的负载上升，在数据库性能较差、规格较低或业务量较大的情况下（例如源库有大量慢SQL、存在无主键表或目标库存在死锁等），可能会加重数据库压力，甚至导致数据库服务不可用。因此您需要在执行数据同步前评估源库和目标库的性能，同时建议您在业务低峰期执行数据同步（例如源库和目标库的CPU负载在30%以下）。

同步限制

- 同步对象仅支持数据表。
- 不支持BIT、VARBIT、GEOMETRY、ARRAY、UUID、TSQUERY、TSVECTOR、TXID_SNAPSHOT类型的数据同步。
- 暂不支持同步前缀索引，如果源库存在前缀索引可能导致数据同步失败。
- 在数据同步时，请勿对源库的同步对象使用gh-ost或pt-online-schema-change等类似工具执行在线DDL变更，否则会导致同步失败。

支持同步的SQL操作

- DML操作：INSERT、UPDATE、DELETE。
- DDL操作：ADD COLUMN、RENAME COLUMN。

> 说明 不支持CREATE TABLE操作，如果您需要将新增的表作为同步对象，则需要执行新增同步对象操作。

支持的同步架构

- 1对1单向同步。
- 1对多单向同步。
- 多对1单向同步。

术语及概念对应关系

MySQL	云原生数据仓库AnalyticDB PostgreSQL
Database	Schema
Table	Table

操作步骤

1. 购买数据同步作业，详情请参见购买流程。

   > 说明 购买时，选择源实例为MySQL，目标实例为AnalyticDB for PostgreSQL，并选择同步拓扑为单向同步。

2. 登录数据传输控制台。
3. 在左侧导航栏，单击**数据同步**。
4. 在**同步作业列表**页面顶部，选择同步的目标实例所属地域。

   

5. 定位至已购买的数据同步实例，单击**配置同步链路**。
6. 配置同步作业的源实例及目标实例信息。

   

类别	配置	说明
无	同步作业名称	DTS会自动生成一个同步作业名称，建议配置具有业务意义的名称（无唯一性要求），便于后续识别。
源实例信息	实例类型	选择**通过专线/VPN网关/智能接入网关接入的自建数据库**。
	实例地区	购买数据同步实例时选择的源实例地域信息，不可变更。
	对端专有网络	选择自建数据库接入的VPC ID。
	数据库类型	固定为**MySQL**，不可变更。
	IP地址	填入自建MySQL数据库的服务器IP地址。
	端口	填入自建数据库的服务端口，默认为**3306**。
	数据库账号	填入自建MySQL的数据库账号。 > 说明 该账号需具备REPLICATION CLIENT、REPLICATION SLAVE、SHOW VIEW和所有同步对象的SELECT权限。
	数据库密码	填入该数据库账号对应的密码。
目标实例信息	实例类型	固定为**AnalyticDB for PostgreSQL**，无需设置。
	实例地区	购买数据同步实例时选择的目标实例地域信息，不可变更。
	实例ID	选择目标云原生数据仓库AnalyticDB PostgreSQL实例ID。
	数据库名称	填入同步目标表所属的数据库名称。

类别	配置	说明
	数据库账号	填入云原生数据仓库AnalyticDB PostgreSQL的**初始账号**，详情请参见**设置账号**。 **说明** 您也可以填入具备RDS_SUPERUSER权限的账号，创建方法请参见**用户权限管理**。
	数据库密码	填入该数据库账号对应的密码。

7. **单击页面右下角的授权白名单并进入下一步。**

   说明 此步骤会将DTS服务器的IP地址自动添加到云原生数据仓库AnalyticDB PostgreSQL的白名单中，用于保障DTS服务器能够正常连接目标实例。

8. **配置同步策略及同步对象。**

类别	配置	说明
同步策略配置	同步初始化	默认情况下，您需要同时选中**结构初始化**和**全量数据初始化**。预检查完成后，DTS会将源实例中待同步对象的结构及数据在目标实例中初始化，作为后续增量同步数据的基线数据。
	目标已存在表的处理模式	○ **清空目标表的数据** 在预检查阶段跳过**同名对象存在性检查**的检查项目。全量初始化之前将目标表的数据清空。适用于完成同步任务测试后的正式同步场景。 ○ **忽略报错并继续执行** 在预检查阶段跳过**同名对象存在性检查**的检查项目。全量初始化时直接追加数据。适用于多张表同步到一张表的汇总同步场景。
	同步操作类型	根据业务需求选择需要同步的操作类型： ○ **Insert** ○ **Update** ○ **Delete** ○ **AlterTable**

类别	配置	说明
选择同步对象	无	在源库对象框中单击待同步的表，然后单击 图标将其移动至已选择对象框。 说明 ○ 同步对象的选择粒度为表。 ○ 如果需要目标表中的列名称与源表不同，需要使用DTS的字段映射功能，详情请参见设置同步对象在目标实例中的名称。

9. 设置待同步的表在云原生数据仓库AnalyticDB PostgreSQL中的主键列和分布列信息。

   说明 当您在上一步中选择了结构初始化才会出现该页面。关于主键列和分布列的详细说明，请参见表的约束定义和表分布键定义。

10. 上述配置完成后，单击页面右下角的预检查并启动。

    说明
    - 在数据同步作业正式启动之前，会先进行预检查。只有预检查通过后，才能成功启动数据同步作业。
    - 如果预检查失败，单击具体检查项后的 图标，查看失败详情。根据提示修复后，重新进行预检查。

11. 在预检查对话框中显示预检查通过后，关闭预检查对话框，同步作业将正式开始。
12. 等待同步作业的链路初始化完成，直至处于同步中状态。

    您可以在数据同步页面，查看数据同步作业的状态。

8.5. rds_dbsync迁移/同步MySQL数据到AnalyticDB for PostgreSQL

rds_dbsync

rds_dbsync 为开源的数据同步/迁移工具 ，其 mysql2pgsql 功能支持不落地的把MySQL中的表迁移到AnalyticDB for PostgreSQL/Greenplum Database/PostgreSQL/PPAS。此工具的原理是，同时连接源端MySQL数据库和目的端数据库，从MySQL库中通过查询得到要导出的数据，然后通过 COPY命令导入到目的端。此工具支持多线程导入（每个工作线程负责导入一部分数据库表）。

参数配置

修改配置文件my.cfg、配置源和目的库连接信息。

- 源库MySQL的连接信息如下：

  > 注意 源库MySQL的连接信息中，用户需要有对所有用户表的读权限。

  ```
  [src.mysql]
  host = "192.168.1.1"
  port = "3306"
  user = "test"
  password = "test"
  db = "test"
  encodingdir = "share"
  encoding = "utf8"
  ```

- 目的库pgsql（包括Postgresql、PPAS和AnalyticDB for PostgreSQL）的连接信息如下：

  > 注意 目的库pgsql的连接信息，用户需要对目标表有写的权限。

  ```
  [desc.pgsql]
  connect_string = "host=192.168.1.2 dbname=test port=3432  user=test password=pgsql"
  ```

mysql2pgsql用法

mysql2pgsql的用法如下所示：

```
./mysql2pgsql -l <tables_list_file> -d -n -j <number of threads> -s <schema of target table>
```

参数说明：

- -l：可选参数，指定一个文本文件，文件中含有需要同步的表；如果不指定此参数，则同步配置文件中指定数据库下的所有表。`<tables_list_file>`为一个文件名，里面含有需要同步的表集合以及表上查询的条件，其内容格式示例如下：

  ```
  table1 : select * from table_big where column1 < '2016-08-05'
  table2 :
  table3
  table4: select column1, column2 from tableX where column1 != 10
  table5: select * from table_big where column1 >= '2016-08-05'
  ```

- -d：可选参数，表示只生成目的表的建表DDL语句，不实际进行数据同步。
- -n：可选参数，需要与-d一起使用，指定在DDL语句中不包含表分区定义。
- -j：可选参数，指定使用多少线程进行数据同步；如果不指定此参数，会使用5个线程并发。
- -s：可选参数，指定目标表的schema，目前仅支持设定为public。

典型用法

全库迁移

全库迁移的操作步骤如下所示：

1. 通过如下命令，获取目的端对应表的DDL。

```
./mysql2pgsql -d
```

2. 根据这些DDL，再加入Distribution Key等信息，在目的端创建表。
3. 执行如下命令，同步所有表：

```
./mysql2pgsql
```

此命令会把配置文件中所指定数据库中的所有MySQL 表数据迁移到目的端。过程中使用5个线程（即缺省线程数为5），读取和导入所有涉及的表数据。

部分表迁移

1. 编辑一个新文件 tab_list.txt，放入如下内容：

```
t1
t2 : select * from t2 where c1 > 138888
```

2. 执行如下命令，同步指定的t1和t2表（注意t2表只迁移符合c1 > 138888条件的数据）：

```
./mysql2pgsql -l tab_list.txt
```

下载与说明

- 下载mysql2pgsql二进制安装包下载，请单击这里。
- 查看mysql2pgsql源码编译说明，请单击这里。

8.6. rds_dbsync迁移/同步PostgreSQL数据到AnalyticDB for PostgreSQL

开源工具 rds_dbsync的pgsql2pgsql功能，支持把AnalyticDB for PostgreSQL/Greenplum Database/PostgreSQL/PPAS中的表迁AnalyticDB for PostgreSQL/Greenplum Database/PostgreSQL/PPAS。

pgsql2pgsql支持的功能

pgsql2pgsql支持如下功能：

- PostgreSQL/PPAS/Greenplum Database/AnalyticDB for PostgreSQL全量数据迁移到PostgreSQL/PPAS/Greenplum Database/AnalyticDB for PostgreSQL。
- PostgreSQL/PPAS（版本大于9.4）全量+增量迁移到PostgreSQL/PPAS。

参数配置

修改配置文件postgresql.conf、配置源和目的库连接信息。

- 源库pgsql连接信息如下所示：

> 注意 源库pgsql的连接信息中，用户最好是对应DB的owner。

```
[src.pgsql]
connect_string = "host=192.168.1.1 dbname=test port=3432 user=test password=pgsql"
```

- 本地临时Database pgsql连接信息如下所示：

```
[local.pgsql]
connect_string = "host=192.168.1.2 dbname=test port=3432 user=test2 password=pgsql"
```

- 目的库pgsql连接信息如下所示：

> 注意 目的库pgsql的连接信息，用户需要对目标表有写权限。

```
[desc.pgsql]
connect_string = "host=192.168.1.3 dbname=test port=3432 user=test3 password=pgsql"
```

> 注意
> - 如果要做增量数据同步，连接源库需要有创建replication slot的权限。
> - 由于PostgreSQL 9.4及以上版本支持逻辑流复制，所以支持作为数据源的增量迁移。打开下列内核参数才能让内核支持逻辑流复制功能。
>
> ```
> wal_level = logical
> max_wal_senders = 6
> max_replication_slots = 6
> ```

pgsql2pgsql用法

全库迁移

进行全库迁移，请执行如下命令：

```
./pgsql2pgsql
```

迁移程序会默认把对应pgsql库中所有用户的表数据将迁移到pgsql。

状态信息查询

连接本地临时Database，可以查看到单次迁移过程中的状态信息。这些信息被放在表db_sync_status中，包括全量迁移的开始和结束时间、增量迁移的开始时间和增量同步的数据情况。

下载与说明

- 下载rds_dbsync二进制安装包，请单击这里。
- 查看rds_dbsync源码编译说明，请单击这里。