恒生电子股份有限公司

Oracle 到 LightDB
迁移实施手册

恒生研究院

2022 年 4 月

文档修改记录

版本	修订人	修订说明	批准人	发布日期
1.0.0.0		初稿		20220101
1.0.0.1	姚崇	v22.1 发版更新		20220406
1.0.0.2	姚崇	v22.3 发版更新		20221011
1.0.0.3	姚崇	v23.1 添加第四章节，数据校验		20230324
1.0.0.3	姚崇	v23.1 添加 docker 的迁移方式		20230327

说　明

模板版本信息
编辑部门：EPG
批准日期：2018/9/26

目 录

一、引言

1.1 编写目的

本文档为恒生电子股份有限公司 Oracle 到 LightDB 迁移实施手册说明书，本文档主要阐述 Oracle 到 LightDB 工具的迁移实施的详细功能介绍，完整的数据库功能请参考《LightDB 用户手册》。其迁移实施手册提供了一种异构数据库 Oracle 向 LightDB 迁移方案，旨在满足数据库用户的需求，简化了对 LightDB 数据库的维护和使用。

1.2 预期读者

本文档主要适用于LightDB数据库的：

- 数据库管理员
- 开发工程师
- 测试工程师
- 技术支持工程师

1.3 参考文献

《LightDB 数据库安装手册》

二、LightDB 功能介绍 ora2pg 安装

LightDB 提供两种 ora2pg 的安装，第一种手工安装，第二种使用 docker 安装。
安装可在 lightdb 所在服务器进行，也可以单独使用一台服务器，无硬性要求，推荐 Linux 7 x86_64

2.1 手工安装 ora2pg

2.1.1 安装 perl 依赖

首先要配置好 yum 源

```
[root@localhost ~]$ yum install -y perl perl-ExtUtils-CBuilder
perl-ExtUtils-MakeMaker
Loaded plugins: langpacks, ulninfo
```

```
Resolving Dependencies
--> Running transaction check
---> Package perl.x86_64 4:5.16.3-294.el7_6 will be updated
---> Package perl.x86_64 4:5.16.3-297.el7 will be an update

(省略中间...)

Dependency Installed:
gdbm-devel.x86_64 0:1.10-8.el7
perl-ExtUtils-Install.noarch 0:1.58-297.el7
perl-ExtUtils-ParseXS.noarch 1:3.18-3.el7
perl-Locale-Maketext.noarch 0:1.23-3.el7
perl-Module-CoreList.noarch 1:2.76.02-297.el7
perl-Module-Load-Conditional.noarch 0:0.54-3.el7
perl-Params-Check.noarch 1:0.38-2.el7
perl-Test-Harness.noarch 0:3.28-3.el7
perl-version.x86_64 3:0.99.07-6.el7
systemtap-sdt-devel.x86_64 0:4.0-13.0.1.el7

Updated:
perl.x86_64 4:5.16.3-297.el7

Dependency Updated:
perl-libs.x86_64 4:5.16.3-297.el7

Complete!
```

2.1.2 安装 DBI 模块

DBI，Database Independent Interface，是 Perl 语言连接数据库的接口，下载地址 https://metacpan.org/release/DBI
下载 DBI-1.643.tar.gz
然后解压安装，如果安装失败，多数是 2.11 章节 perl 没配置好

```
[root@node1 ora2pg]# tar -xzvf DBI-1.643.tar.gz
[root@node1 ora2pg]# cd DBI-1.643/
[root@node1 DBI-1.643]# perl Makefile.PL
[root@node1 DBI-1.643]# make && make install
```

2.1.3 安装 DBD::Oracle 模块

添加环境变量，需要在本机安装 Oracle，在 root 下执行 export 即可

```
unzip instantclient-basic-linux.x64-21.6.0.0.0dbru.zip
```

```
unzip instantclient-sdk-linux.x64-21.6.0.0.0dbru.zip
unzip instantclient-sqlplus-linux.x64-21.6.0.0.0dbru.zip
export PATH
export EDITOR=vi
export GGATE=
export NLS_LANG=AMERICAN_AMERICA.AL32UTF8
export ORACLE_BASE=/root/instantclient_21_6
export ORACLE_HOME=/root/instantclient_21_6
export ORACLE_SID=
export PATH=$ORACLE_HOME:$ORACLE_HOME/OPatch:$GGATE:$PATH
export LD_LIBRARY_PATH=$ORACLE_HOME:/usr/lib:$GGATE:$LD_LIBRARY_PATH
export TNS_ADMIN=$ORACLE_HOME/network/admin
```

安装 DBD-Oracle 驱动，下载地址：点开后面连接找到 download 下载 https://metacpan.org/pod/release/PYTHIAN/DBD-Oracle-1.74/lib/DBD/Oracle.pm

【 https://metacpan.org/pod/DBD::Oracle 下载最新 1.83 版本】

或者直接点开 https://cpan.metacpan.org/authors/id/P/PY/PYTHIAN/DBD-Oracle-1.74.tar.gz 下载 DBD-Oracle-1.74.tar.gz

```
[root@node1 ora2pg]# tar -xzvf DBD-Oracle-1.83.tar.gz
[root@node1 ora2pg]# cd DBD-Oracle-1.83/
[root@node1 DBD-Oracle-1.83]# perl Makefile.PL
[root@localhost /usr/local/DBD-Oracle-1.83]$ make && make install
...略...
Installing /usr/local/share/man/man3/DBD::Oracle::Troubleshooting::Win64.3pm
Installing /usr/local/share/man/man3/DBD::Oracle::Troubleshooting::Cygwin.3pm
Installing /usr/local/share/man/man3/DBD::Oracle::Troubleshooting::Hpux.3pm
Installing /usr/local/share/man/man3/DBD::Oracle::GetInfo.3pm
Appending installation info to /usr/lib64/perl5/perllocal.pod
```

最后安装成功信息如上所示

2.1.4 安装 DBD::Pg 模块

安装 DBD-Pg 驱动，下载地址 https://metacpan.org/release/DBD-Pg，下载出 DBD-Pg-3.14.2.tar.gz

登录到 lightdb 用户，配置 pg_config

```
cd $PGHOME/bin
ln -s lt_config pg_config
```

root 用户配置环境变量并安装

```
export LIGHTDB_PORT=5432
export PGUSER=lightdb
export LIGHTDB_HOST=10.0.4.4
export POSTGRES_HOME=/home/lightdb/base/lightdb-x/13.3-22.1
export PGDATA=/home/lightdb/data
```

```
export
PATH=${POSTGRES_HOME}/bin:${POSTGRES_HOME}/tools/iftop/bin:${POSTGRES_HOME}/
tools/iotop/bin:${POSTGRES_HOME}/tools/linux-ftools/bin:${POSTGRES_HOME}/too
ls/vmtouch/bin:${PATH}
export
LD_LIBRARY_PATH=${POSTGRES_HOME}/lib:${POSTGRES_HOME}/lib/ltext:${LD_LIBRARY
_PATH}


[root@node1 ora2pg]# tar -zxvf DBD-Pg-3.15.1.tar.gz
[root@node1 ora2pg]# cd DBD-Pg-3.15.1/

[root@localhost /usr/local]$ perl Makefile.PL
[root@localhost /usr/local]$ make
[root@localhost /usr/local]$ make install
Files found in blib/arch: installing files in blib/lib into architecture dependent
library tree
Installing /usr/local/lib64/perl5/auto/DBD/Pg/Pg.so
Installing /usr/local/lib64/perl5/auto/DBD/Pg/Pg.bs
Installing /usr/local/lib64/perl5/DBD/Pg.pm
Installing /usr/local/lib64/perl5/Bundle/DBD/Pg.pm
Installing /usr/local/share/man/man3/Bundle::DBD::Pg.3pm
Installing /usr/local/share/man/man3/DBD::Pg.3pm
Appending installation info to /usr/lib64/perl5/perllocal.pod
```

最后安装成功信息如上所示

2.1.5 安装 ora2pg

下载地址 https://sourceforge.net/projects/ora2pg/

```
[root@node1 ora2pg]# cd ora2pg
[root@node1 ora2pg]# ls
changelog  doc  INSTALL  lib  LICENSE  Makefile.PL  MANIFEST  packaging  README
scripts
[root@node1 ora2pg]# perl Makefile.PL
Checking if your kit is complete...
[root@node1 ora2pg]# make
cp lib/Ora2Pg.pm blib/lib/Ora2Pg.pm
cp lib/Ora2Pg/GEOM.pm blib/lib/Ora2Pg/GEOM.pm
cp lib/Ora2Pg/PLSQL.pm blib/lib/Ora2Pg/PLSQL.pm
cp lib/Ora2Pg/Oracle.pm blib/lib/Ora2Pg/Oracle.pm
cp lib/Ora2Pg/MySQL.pm blib/lib/Ora2Pg/MySQL.pm
cp scripts/ora2pg blib/script/ora2pg
/usr/bin/perl -MExtUtils::MY -e 'MY->fixin(shift)' -- blib/script/ora2pg
```

```
cp scripts/ora2pg_scanner blib/script/ora2pg_scanner
/usr/bin/perl -MExtUtils::MY -e 'MY->fixin(shift)' --
blib/script/ora2pg_scanner
Manifying blib/man3/ora2pg.3
[root@node1 ora2pg]# make install
Installing /usr/local/share/perl5/Ora2Pg.pm
Installing /usr/local/share/perl5/Ora2Pg/GEOM.pm
Installing /usr/local/share/perl5/Ora2Pg/PLSQL.pm
Installing /usr/local/share/perl5/Ora2Pg/Oracle.pm
Installing /usr/local/share/perl5/Ora2Pg/MySQL.pm
Installing /usr/local/share/man/man3/ora2pg.3
Installing /usr/local/bin/ora2pg_scanner
Installing /usr/local/bin/ora2pg
Installing default configuration file (ora2pg.conf.dist) to /etc/ora2pg
Appending installation info to /usr/lib64/perl5/perllocal.pod
```

2.1.6 查看软件是否安装成功

```
[root@node1 ora2pg]# ora2pg --help
Usage: ora2pg [-dhpqv --estimate_cost --dump_as_html] [--option value]

   -a | --allow str  : Comma separated list of objects to allow from export.
                       Can be used with SHOW_COLUMN too.
```

如上，ora2pg 命令正确显示即安装成功

2.2 使用 docker 安装

2.2.1 导入镜像

下载 docker 镜像并执行下面命令导入：

```
$ docker load -i /home/lightdb/ora2pg.tar
sudo docker run -it --rm -v /home/lightdb/config:/config ora2pg:1 /bin/bash
```

```
[lightdb@hs docker]$ docker load -i /home/lightdb/ora2pg.tar
764055ebc9a7: Loading layer [==================================================>]  72.53MB/72.53MB
4e5c615cbaf5: Loading layer [==================================================>]  3.584kB/3.584kB
0331048e9df6: Loading layer [==================================================>]  59.81MB/59.81MB
4b3f02c5470d: Loading layer [==================================================>]  4.096kB/4.096kB
a71002aa012d: Loading layer [==================================================>]  269.7MB/269.7MB
4745c22c676c: Loading layer [==================================================>]  4.868MB/4.868MB
806722b8e583: Loading layer [==================================================>]  54.15MB/54.15MB
6db07d8402ec: Loading layer [==================================================>]   5.12kB/5.12kB
b341f07d521c: Loading layer [==================================================>]  228.2MB/228.2MB
1060577ff0b6: Loading layer [==================================================>]  62.35MB/62.35MB
ec2511903b6d: Loading layer [==================================================>]  2.048kB/2.048kB
d434d295b3ea: Loading layer [==================================================>]  131.6kB/131.6kB
8970e81d8a49: Loading layer [==================================================>]  2.048kB/2.048kB
962c60586eae: Loading layer [==================================================>]  4.608kB/4.608kB
Loaded image ID: sha256:0aa0eb8cb1b6563bd2d4e76e3a2718cd3e1f5f99392daafe914f988119c1abcb
```

2.2.2 重命名镜像

```
$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
<none>        <none>    0aa0eb8cb1b6   20 months ago   741MB
$ sudo docker tag  0aa0eb8cb1b6 ora2pg:1
$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
ora2pg        1         0aa0eb8cb1b6   20 months ago   741MB
```

到此 docker 镜像方式安装完成

三、ora2pg 迁移数据

3.1 编辑配置文件 ora2pg.conf

用 root 账户，编辑配置文件 ora2pg.conf

```
### Oracle 客户端的 ORACLE_HOME
ORACLE_HOME /oracle/app/product/19.3.0/db_1
### 源端连接串  MySQL 如:   dbi:mysql:host=192.168.1.10;database=tpch;port=3306
ORACLE_DSN  dbi:Oracle:host=10.0.4.4;sid=orcl1;port=1521

PG_VERSION 13
TRUNCATE_TABLE      1 #If set 1, a TRUNCATE TABLE instruction will be add before
loading data. This is usable only during INSERT or COPY export type.
PG_NUMERIC_TYPE     0
PG_INTEGER_TYPE     0 #指定 0 转换成 bigint 或者 bigint
#PG_INTEGER_TYPE    1 #指定 1 转换成 numeric
EXPORT_SCHEMA       1 #ALTER SCHEMA fund60trans1 OWNER TO fund60trans1;  SET
search_path = fund60trans1,public;
DROP_IF_EXISTS      1 #CREATE SCHEMA IF NOT EXISTS fund60trans1;
PREFIX_PARTITION    1 #导出的分区表加上主表文件名前缀
PREFIX_SUB_PARTITION 1 #同上，针对的对象是子分区
FILE_PER_CONSTRAINT  1 #将导出的约束单独放在一个文件中
FILE_PER_INDEX      1 #将导出的索引单独放在一个文件中
FILE_PER_FKEYS      1 #将导出的外键放在单独的文件中
USE_RESERVED_WORDS   1 #如果 oracle 中导出的表名或列名有关键字，则导出时自动为其加上双引
号，尽量询问应用看能否更改 PG 中的表名或字段名
TRANSACTION readonly   #设置为只读事务，避免误操作 Oracle 端数据
DISABLE_UNLOGGED    1 #禁止转换 unlogged 表，避免出现 unlogged 表
#DEFAULT_NUMERIC    float
```

```
JOBS              6

NLS_LANG    AMERICAN_AMERICA.UTF8
```

3.2 编辑导出脚本

vim exp 添加如下内容

```
filedate=`date +"%Y%m%d_%H%M%S"`
if [ "$1" = "ora2pg" ]; then
   if [ "$2" = "" ]; then
     echo "Usage: ora ora2pg <exp/imp>"
     exit 0
   fi

   if [ "$2" = "exp" ]; then
     echo ""
     echo "First edit ora2pg config file ora2pg.conf to make sure datasource is
correct"
     echo "vim ora set data_type_list parameter to set export object type"
     echo ""
     while true
     do

       read -p "please input schema name[exit or EXIT]:" schema
       read -p "please input schema password[exit or EXIT]:" password
       if [ -z "${schema}" ];then
       echo "The username or password is empty, program exit"
       exit 0
       fi
       if [ -z "${password}" ];then
       echo "The username or password is empty, program exit"
       exit 0
       fi
       if [ "$schema" = "exit" -o "$schema" = "EXIT" -o "$schema" = "exit" -o
"$schema" = "EXIT" ]; then
         echo "input exit, program exit"
                exit 0          ###执行退出命令
       fi
       connect=`ora2pg -t SHOW_VERSION -c ora2pg.conf -u $schema -w $password`
       echo $connect
       ora28000_flag="ORA-28000"
       ora01017_flag="ORA-01017"
```

```
      oracle_logon_flag="Oracle"
      if [[ "$connect" =~ ^"${ora01017_flag}".* ]]; then
        echo $connect
        echo "ORA-01017: invalid username/password; logon denied"
      elif [[ "$connect" =~ ^"${ora28000_flag}".* ]]; then
        echo $connect
        echo "ORA-28000: The account is locked"
      elif [[ "$connect" =~ ^"${oracle_logon_flag}".* ]]; then
              mkdir -p $schema
              # default export object
        data_type_list='TABLE
        PARTITION
        COPY
        SEQUENCE
        SYNONYM'
        #PROCEDURE
        #FUNCTION
        #PACKAGE
        #GRANT
        #VIEW
        for data_type in $data_type_list
        do
          echo 'exporting' ${data_type}' please wait...'
          # 添加 -P 10 指定并行
          ora2pg -c ora2pg.conf -t SHOW_REPORT --estimate_cost -u $schema -w
$password -n $schema -t$data_type -b $schema -o
${data_type}_${schema}_${filedate}.sql >
${data_type}_${schema}_${filedate}.log 2>&1 &
        done
        echo 'background exporting...'
      fi
    done
   elif [ "$2" = "imp" ]; then
    echo ""
    echo "Make sure target Lightdb has been created database and schemas for the
importing database"
    echo ""
    while true
    do
      read -p "please input superuser [exit or EXIT]:" username
      read -p "please input superuser password[exit or EXIT]:" password
      read -p "please input database name[exit or EXIT]:" db_name
      read -p "please input target ip address[exit or EXIT]:" ip
      read -p "please input target lightdb port[exit or EXIT]:" port
```

```
      read -p "please input data folder[exit or EXIT]:" data_folder
      read -p "import option?
[table_only/view_only/data_only/index_only/foreign_key_only/all/ | exit or
EXIT]:" import_option

      if [ -z "${username}" ];then
        echo "The superuser name is empty, program exit"
        exit 0
      fi
      if [ -z "${password}" ];then
        echo "The superuser password is empty, program exit"
        exit 0
      fi
      if [ -z "${db_name}" ];then
        echo "The target database name empty, program exit"
        exit 0
      fi
      if [ -z "${ip}" ];then
        echo "The target database ip information is empty, program exit"
        exit 0
      fi
      if [ -z "${port}" ];then
        echo "The target database port information is empty, program exit"
        exit 0
      fi
      if [ -z "${data_folder}" ];then
        echo "The data folder information is empty, program exit"
        exit 0
      fi
      if [ -z "${import_option}" ];then
        echo "The import option parameter is empty, program exit"
        exit 0
      fi
      if [ "$import_option" = "exit" -o "$import_option" = "EXIT" -o "$username"
= "exit" -o "$username" = "EXIT" -o "$password" = "exit" -o "$password" = "EXIT"
-o "$db_name" = "exit" -o "$db_name" = "EXIT" -o "$ip" = "exit" -o "$ip" = "EXIT"
-o "$port" = "exit" -o "$port" = "EXIT" -o "$data_folder" = "exit" -o "$data_folder"
= "EXIT" ]; then
        echo "input exit, program exit"
               exit 0  ###执行退出命令
      fi

      #-v ON_ERROR_STOP=ON
# 如果要单独导入数据，需要先删除掉外键
```

```
# alter table act_ge_bytearray drop constraint if EXISTS act_fk_bytearr_depl;
# 然后再根据 FKEYS_中的内容去创建
# ALTER TABLE act_ge_bytearray ADD CONSTRAINT act_fk_bytearr_depl FOREIGN KEY
(deployment_id_) REFERENCES act_re_deployment(id_) ON DELETE NO ACTION NOT
DEFERRABLE INITIALLY IMMEDIATE;
      if [ "$import_option" = "data_only" ]; then
        echo 'Importing data begin at '`date +"%Y-%m-%d %H:%M:%S"`'...'
> imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &&
        PGPASSWORD=$password ltsql -U $username -h $ip -p $port -d $db_name -f
${data_folder}/`ls ${data_folder} | grep '^COPY_'`     >>
imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &&
        echo 'analyze verbose begin at '`date +"%Y-%m-%d %H:%M:%S"`'...'
>> imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &&
        PGPASSWORD=$password ltsql -U $username -h $ip -p $port -d $db_name -c
"analyze verbose"                                    >>
imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &

      elif [ "$import_option" = "view_only" ]; then
        echo 'Importing views begin at '`date +"%Y-%m-%d %H:%M:%S"`'...'
> imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &&
        PGPASSWORD=$password ltsql -U $username -h $ip -p $port -d $db_name -f
${data_folder}/`ls ${data_folder} | grep '^VIEW_'`     >>
imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &&
        echo 'analyze verbose begin at '`date +"%Y-%m-%d %H:%M:%S"`'...'
>> imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &&
        PGPASSWORD=$password ltsql -U $username -h $ip -p $port -d $db_name -c
"analyze verbose"                                    >>
imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &

      elif [ "$import_option" = "index_only" ]; then
        echo 'Importing indexes begin at '`date +"%Y-%m-%d %H:%M:%S"`'...'
> imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &&
        PGPASSWORD=$password ltsql -U $username -h $ip -p $port -d $db_name -f
${data_folder}/`ls ${data_folder} | grep '^INDEXES_'`  >>
imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &&
        echo 'analyze verbose begin at '`date +"%Y-%m-%d %H:%M:%S"`'...'
>> imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &&
        PGPASSWORD=$password ltsql -U $username -h $ip -p $port -d $db_name -c
"analyze verbose"                                    >>
imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &

      elif [ "$import_option" = "foreign_key_only" ]; then
        echo 'Importing foreign key begin at '`date +"%Y-%m-%d %H:%M:%S"`'...'
> imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &&
```

```
       PGPASSWORD=$password ltsql -U $username -h $ip -p $port -d $db_name -f
${data_folder}/`ls ${data_folder} | grep '^FKEYS_'`     >>
imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &&
       echo 'analyze verbose begin at '`date +"%Y-%m-%d %H:%M:%S"`'...'
>> imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &&
       PGPASSWORD=$password ltsql -U $username -h $ip -p $port -d $db_name -c
"analyze verbose"                                    >>
imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &
      elif [ "$import_option" = "table_only" ]; then
       echo 'Importing only tables begin at '`date +"%Y-%m-%d %H:%M:%S"`'...'
> imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &&
       PGPASSWORD=$password ltsql -U $username -h $ip -p $port -d $db_name -f
${data_folder}/`ls ${data_folder} | grep '^TABLE_'`     >>
imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &&
       echo 'analyze verbose begin at '`date +"%Y-%m-%d %H:%M:%S"`'...'
>> imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &&
       PGPASSWORD=$password ltsql -U $username -h $ip -p $port -d $db_name -c
"analyze verbose"                                    >>
imp_${db_name}_${data_folder}_${import_option}.log 2>&1 &
      elif [ "$import_option" = "all" ]; then
       echo 'Importing all folder please wait...'
       ## 如果要不导入表注释掉下两行...
       echo 'Importing tables begin at '`date +"%Y-%m-%d %H:%M:%S"`'...'
> imp_${db_name}_${data_folder}.log 2>&1 &&
       PGPASSWORD=$password ltsql -U $username -h $ip -p $port -d $db_name -f
${data_folder}/`ls ${data_folder} | grep '^TABLE_'` >>
imp_${db_name}_${data_folder}.log 2>&1 &&
       for sql_file in `ls ${data_folder} | grep -v '^TABLE_' | grep -v '^FKEYS_'`
        do
         echo 'Importing '${sql_file}' begin at '`date
+"%Y-%m-%d %H:%M:%S"`'...'                                    >>
imp_${db_name}_${data_folder}.log 2>&1
         PGPASSWORD=$password ltsql -U $username -h $ip -p $port -d $db_name -f
${data_folder}/${sql_file}            >> imp_${db_name}_${data_folder}.log
2>&1
        done &&
       echo 'analyze verbose begin at '`date +"%Y-%m-%d %H:%M:%S"`'...'
>> imp_${db_name}_${data_folder}.log 2>&1 &&
       PGPASSWORD=$password ltsql -U $username -h $ip -p $port -d $db_name -c
"analyze verbose"                                 >>
imp_${db_name}_${data_folder}.log 2>&1 &&
       echo 'Importing foreign key begin at '`date +"%Y-%m-%d %H:%M:%S"`'...'
>> imp_${db_name}_${data_folder}.log 2>&1 &&
```

```
       PGPASSWORD=$password ltsql -U $username -h $ip -p $port -d $db_name -f
${data_folder}/`ls ${data_folder} | grep '^FKEYS_'` >>
imp_${db_name}_${data_folder}.log 2>&1 &&
       echo 'Importing complete at '`date +"%Y-%m-%d %H:%M:%S"`'...'
>> imp_${db_name}_${data_folder}.log 2>&1 &
      fi
    done
  echo 'importing complete'
  fi
fi
```

3.3 自定义导出对象

多数情况下 Oracle 数据库中是有分区表的，则需要在 ora2pg.conf 中的 data_type_list 同时指定 TABLE 和 PARTITION

```
[root@node1 ora2pg]# ./exp ora2pg exp

First edit ora2pg config file ora2pg.conf to make sure datasource is correct
vim ora set data_type_list parameter to set export object type

please input schema name[exit or EXIT]:scott
please input schema password[exit or EXIT]:tiger
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0
exporting TABLE please wait...
exporting COPY please wait...
exporting SEQUENCE please wait...
exporting SYNONYM please wait...
background exporting...
please input schema name[exit or EXIT]:
please input schema password[exit or EXIT]:
The username or password is empty, program exit
[root@node1 ora2pg]# cd scott
[root@node1 scott]# ls -ltr
total 32
-rw-r--r-- 1 root root  356 Jun 25 12:02 SEQUENCE_scott_20220625_120153.sql
-rw-r--r-- 1 root root  354 Jun 25 12:02 SYNONYM_scott_20220625_120153.sql
-rw-r--r-- 1 root root  784 Jun 25 12:02 PARTITION_scott_20220625_120153.sql
-rw-r--r-- 1 root root 1333 Jun 25 12:02 TABLE_scott_20220625_120153.sql
-rw-r--r-- 1 root root  354 Jun 25 12:02 INDEXES_TABLE_scott_20220625_120153.sql
-rw-r--r-- 1 root root  463 Jun 25 12:02 FKEYS_TABLE_scott_20220625_120153.sql
-rw-r--r-- 1 root root  511 Jun 25 12:02
CONSTRAINTS_TABLE_scott_20220625_120153.sql
-rw-r--r-- 1 root root 3827 Jun 25 12:02 COPY_scott_20220625_120153.sql
```

3.4 自定义导入表结构

```
[root@node1 ora2pg]# ./exp ora2pg imp

Make sure target Lightdb has been created database and schemas for the importing
database

please input superuser [exit or EXIT]:scott
please input superuser password[exit or EXIT]:scott
please input database name[exit or EXIT]:scott
please input target ip address[exit or EXIT]:10.20.30.199
please input target lightdb port[exit or EXIT]:5435
please input data folder[exit or EXIT]:scott
import option?
[table_only/view_only/data_only/index_only/foreign_key_only/all/ | exit or
EXIT]:table_only
please input superuser [exit or EXIT]:
please input superuser password[exit or EXIT]:
please input database name[exit or EXIT]:
please input target ip address[exit or EXIT]:
please input target lightdb port[exit or EXIT]:
please input data folder[exit or EXIT]:
import option?
[table_only/view_only/data_only/index_only/foreign_key_only/all/ | exit or
EXIT]:
The superuser name is empty, program exit
```

查看导入日志

```
[root@node1 ora2pg]# more imp_scott_scott_table_only.log
Importing only tables begin at 2022-06-25 12:19:55...
SET
SET
ltsql:scott/TABLE_scott_20220625_120153.sql:11: NOTICE:  schema "scott"
already exists, skipping
CREATE SCHEMA
ALTER SCHEMA
SET
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
```

```
SET
SET
SET
```

3.5 自定义导入表数据

```
[root@node1 ora2pg]# ./exp ora2pg imp

Make sure target Lightdb has been created database and schemas for the importing
database

please input superuser [exit or EXIT]:scott
please input superuser password[exit or EXIT]:scott
please input database name[exit or EXIT]:scott
please input target ip address[exit or EXIT]:10.20.30.199
please input target lightdb port[exit or EXIT]:5435
please input data folder[exit or EXIT]:scott
import option?
[table_only/view_only/data_only/index_only/foreign_key_only/all/ | exit or
EXIT]:data_only
please input superuser [exit or EXIT]:
please input superuser password[exit or EXIT]:
please input database name[exit or EXIT]:
please input target ip address[exit or EXIT]:
please input target lightdb port[exit or EXIT]:
please input data folder[exit or EXIT]:
import option?
[table_only/view_only/data_only/index_only/foreign_key_only/all/ | exit or
EXIT]:
The superuser name is empty, program exit
```

查看导入日志

```
[root@node1 ora2pg]# more imp_scott_scott_data_only.log
Importing data begin at 2022-06-25 12:23:24...
SET
BEGIN
SET
SET
SET
TRUNCATE TABLE
COPY 0
SET
SET
```

```
SET
```

3.6 目标端数据确认

```
scott@scott=# select * from orders;
 o_orderkey |     o_orderdate     |  o_name
------------+---------------------+----------
      1.00 | 2021-11-11 00:00:00 | xiaoming
      2.00 | 2022-01-11 00:00:00 | xiaogang
      3.00 | 2022-02-11 00:00:00 | xiaoju
(3 rows)

scott@scott=# \d orders
                  Partitioned table "scott.orders"
   Column    |           Type           | Collation | Nullable | Default
-------------+--------------------------+-----------+----------+---------
 o_orderkey  | numeric(20,2)            |           | not null |
 o_orderdate | timestamp without time zone |           | not null |
 o_name      | character varying(79)    |           | not null |
Partition key: RANGE (o_orderdate)
Number of partitions: 3 (Use \d+ to list them.)
```

3.7 使用 docker 方式进行数据迁移

如果使用 docker 方式进行迁移，首先需要确认工作目录，比如您想将数据导入到 /home/lightdb/config，需要现将 ora2pg.conf 和 ora 导出文件放到该目录中，然后执行下面命令登录到 docker 中，执行

```
sudo docker run -it --rm -v /home/lightdb/config:/config ora2pg:1 /bin/bash
```

登录后进行导出执行

```
root@f91cb5dac1b5:~# cd /config
root@f91cb5dac1b5:/config# ls
ora ora2pg.conf
root@f91cb5dac1b5:/config# ./ora ora2pg exp
First edit ora2pg config file ora2pg.conf to make sure datasource is correct
vim ora set data_type_list parameter to set export object type
If specify table input null, exporting schema mode,else exporting specify table

please input schema name[exit or EXIT]:
```

四、数据校验

数据从 Oracle 迁移到 LightDB，要进行数据校验对比，下面给出比对表数量，字段类型、索引数量、具体每个表的数量、约束类型和数量的具体方法，我们以 hr 用户为例：

4.1 Oracle 源端操作

创建精确统计表数量的方法，注意要在业务用户内创建，比如你迁移的是 hr 用户，那么要用 hr 用户登录进行创建，执行

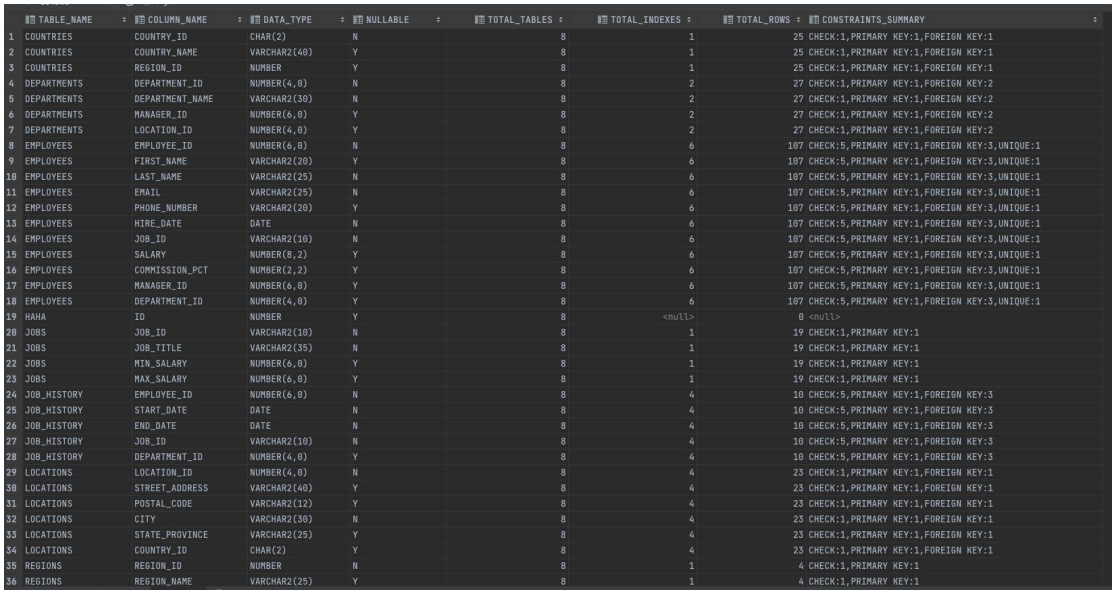
```
CREATE TYPE table_data_count_type AS OBJECT (
  table_name VARCHAR2(50),
  total_rows NUMBER
);
/
CREATE TYPE table_data_count_table AS TABLE OF table_data_count_type;
/
CREATE OR REPLACE FUNCTION get_table_data_counts
RETURN table_data_count_table
AS
  table_data table_data_count_table := table_data_count_table();
  table_name_list sys_refcursor;
  table_name VARCHAR2(50);
  total_rows NUMBER;
BEGIN
  OPEN table_name_list FOR
    SELECT table_name
    FROM user_tables;

  LOOP
    FETCH table_name_list INTO table_name;
    EXIT WHEN table_name_list%NOTFOUND;

    EXECUTE IMMEDIATE 'SELECT COUNT(*) FROM "' || table_name || '"' INTO
total_rows;

    table_data.EXTEND;
    table_data(table_data.COUNT) := table_data_count_type(table_name,
total_rows);
  END LOOP;

  CLOSE table_name_list;

  RETURN table_data;
```

```
END;
/
```

4.1.1 进行查询统计

```
WITH table_count AS (
  SELECT COUNT(*) AS total_tables FROM user_tables
),
index_count AS (
  SELECT table_name, COUNT(*) AS total_indexes
  FROM user_indexes
  GROUP BY table_name
),
constraint_count AS (
  SELECT table_name, LISTAGG(constraint_type_desc || ':' || total_constraints,
',') WITHIN GROUP (ORDER BY constraint_type) AS constraints_summary
  FROM (
    SELECT table_name, constraint_type,
          CASE constraint_type
             WHEN 'C' THEN 'CHECK'
             WHEN 'P' THEN 'PRIMARY KEY'
             WHEN 'R' THEN 'FOREIGN KEY'
             WHEN 'U'  THEN 'UNIQUE'
             ELSE constraint_type
          END AS constraint_type_desc,
          COUNT(*) AS total_constraints
    FROM user_constraints
    GROUP BY table_name, constraint_type
  ) subquery
  GROUP BY table_name
),
table_data_count AS (
  SELECT * FROM TABLE(get_table_data_counts())
)
SELECT utc.table_name, utc.column_name,
  CASE
    WHEN utc.data_type = 'VARCHAR2' THEN utc.data_type || '(' || utc.data_length
|| ')'
    WHEN utc.data_type = 'NUMBER' AND utc.data_precision IS NOT NULL AND
utc.data_scale IS NOT NULL THEN utc.data_type || '(' || utc.data_precision ||
',' || utc.data_scale || ')'
    WHEN utc.data_type = 'NUMBER' AND utc.data_precision IS NULL AND
utc.data_scale IS NULL THEN utc.data_type
```

```
   WHEN utc.data_type = 'CHAR' THEN utc.data_type || '(' || utc.data_length ||
')'
   WHEN utc.data_type = 'DATE' THEN utc.data_type
   WHEN utc.data_type = 'TIMESTAMP' THEN utc.data_type || '(' ||
utc.data_precision || ')'
   WHEN utc.data_type = 'TIMESTAMP WITH TIME ZONE' THEN utc.data_type || '(' ||
utc.data_precision || ')'
   WHEN utc.data_type = 'TIMESTAMP WITH LOCAL TIME ZONE' THEN utc.data_type ||
'(' || utc.data_precision || ')'
   WHEN utc.data_type = 'INTERVAL YEAR TO MONTH' THEN utc.data_type
   WHEN utc.data_type = 'INTERVAL DAY TO SECOND' THEN utc.data_type || '(' ||
utc.data_precision || ')'
   WHEN utc.data_type = 'LONG' THEN utc.data_type
   WHEN utc.data_type = 'CLOB' THEN utc.data_type
   WHEN utc.data_type = 'BLOB' THEN utc.data_type
   WHEN utc.data_type = 'BFILE' THEN utc.data_type
   WHEN utc.data_type = 'RAW' THEN utc.data_type || '(' || utc.data_length ||
')'
   WHEN utc.data_type = 'LONG RAW' THEN utc.data_type
   WHEN utc.data_type = 'ROWID' THEN utc.data_type
   WHEN utc.data_type = 'UROWID' THEN utc.data_type || '(' || utc.data_length
|| ')'
   WHEN utc.data_type = 'NCHAR' THEN utc.data_type || '(' || utc.data_length ||
')'
   WHEN utc.data_type = 'NVARCHAR2' THEN utc.data_type || '(' || utc.data_length
|| ')'
   WHEN utc.data_type = 'NCLOB' THEN utc.data_type
   ELSE utc.data_type
  END AS data_type,
  utc.nullable, tc.total_tables, idx.total_indexes, tdc.total_rows,
cc.constraints_summary
FROM user_tab_columns utc
JOIN table_count tc ON 1=1
LEFT JOIN index_count idx ON utc.table_name = idx.table_name
JOIN table_data_count tdc ON utc.table_name = tdc.table_name
LEFT JOIN constraint_count cc ON utc.table_name = cc.table_name
ORDER BY utc.table_name, utc.column_id;
```



	TABLE_NAME	COLUMN_NAME	DATA_TYPE	NULLABLE	TOTAL_TABLES	TOTAL_INDEXES	TOTAL_ROWS	CONSTRAINTS_SUMMARY
1	COUNTRIES	COUNTRY_ID	CHAR(2)	N	8	1	25	CHECK:1,PRIMARY KEY:1,FOREIGN KEY:1
2	COUNTRIES	COUNTRY_NAME	VARCHAR2(40)	Y	8	1	25	CHECK:1,PRIMARY KEY:1,FOREIGN KEY:1
3	COUNTRIES	REGION_ID	NUMBER	Y	8	1	25	CHECK:1,PRIMARY KEY:1,FOREIGN KEY:1
4	DEPARTMENTS	DEPARTMENT_ID	NUMBER(4,0)	N	8	2	27	CHECK:1,PRIMARY KEY:1,FOREIGN KEY:2
5	DEPARTMENTS	DEPARTMENT_NAME	VARCHAR2(30)	N	8	2	27	CHECK:1,PRIMARY KEY:1,FOREIGN KEY:2
6	DEPARTMENTS	MANAGER_ID	NUMBER(6,0)	Y	8	2	27	CHECK:1,PRIMARY KEY:1,FOREIGN KEY:2
7	DEPARTMENTS	LOCATION_ID	NUMBER(4,0)	Y	8	2	27	CHECK:1,PRIMARY KEY:1,FOREIGN KEY:2
8	EMPLOYEES	EMPLOYEE_ID	NUMBER(6,0)	N	8	6	107	CHECK:5,PRIMARY KEY:1,FOREIGN KEY:3,UNIQUE:1
9	EMPLOYEES	FIRST_NAME	VARCHAR2(20)	Y	8	6	107	CHECK:5,PRIMARY KEY:1,FOREIGN KEY:3,UNIQUE:1
10	EMPLOYEES	LAST_NAME	VARCHAR2(25)	N	8	6	107	CHECK:5,PRIMARY KEY:1,FOREIGN KEY:3,UNIQUE:1
11	EMPLOYEES	EMAIL	VARCHAR2(25)	N	8	6	107	CHECK:5,PRIMARY KEY:1,FOREIGN KEY:3,UNIQUE:1
12	EMPLOYEES	PHONE_NUMBER	VARCHAR2(20)	Y	8	6	107	CHECK:5,PRIMARY KEY:1,FOREIGN KEY:3,UNIQUE:1
13	EMPLOYEES	HIRE_DATE	DATE	N	8	6	107	CHECK:5,PRIMARY KEY:1,FOREIGN KEY:3,UNIQUE:1
14	EMPLOYEES	JOB_ID	VARCHAR2(10)	N	8	6	107	CHECK:5,PRIMARY KEY:1,FOREIGN KEY:3,UNIQUE:1
15	EMPLOYEES	SALARY	NUMBER(8,2)	Y	8	6	107	CHECK:5,PRIMARY KEY:1,FOREIGN KEY:3,UNIQUE:1
16	EMPLOYEES	COMMISSION_PCT	NUMBER(2,2)	Y	8	6	107	CHECK:5,PRIMARY KEY:1,FOREIGN KEY:3,UNIQUE:1
17	EMPLOYEES	MANAGER_ID	NUMBER(6,0)	Y	8	6	107	CHECK:5,PRIMARY KEY:1,FOREIGN KEY:3,UNIQUE:1
18	EMPLOYEES	DEPARTMENT_ID	NUMBER(4,0)	Y	8	6	107	CHECK:5,PRIMARY KEY:1,FOREIGN KEY:3,UNIQUE:1
19	HAHA	ID	NUMBER	Y	8	<null>	0	<null>
20	JOBS	JOB_ID	VARCHAR2(10)	N	8	1	19	CHECK:1,PRIMARY KEY:1
21	JOBS	JOB_TITLE	VARCHAR2(35)	N	8	1	19	CHECK:1,PRIMARY KEY:1
22	JOBS	MIN_SALARY	NUMBER(6,0)	Y	8	1	19	CHECK:1,PRIMARY KEY:1
23	JOBS	MAX_SALARY	NUMBER(6,0)	Y	8	1	19	CHECK:1,PRIMARY KEY:1
24	JOB_HISTORY	EMPLOYEE_ID	NUMBER(6,0)	N	8	4	10	CHECK:5,PRIMARY KEY:1,FOREIGN KEY:3
25	JOB_HISTORY	START_DATE	DATE	N	8	4	10	CHECK:5,PRIMARY KEY:1,FOREIGN KEY:3
26	JOB_HISTORY	END_DATE	DATE	N	8	4	10	CHECK:5,PRIMARY KEY:1,FOREIGN KEY:3
27	JOB_HISTORY	JOB_ID	VARCHAR2(10)	N	8	4	10	CHECK:5,PRIMARY KEY:1,FOREIGN KEY:3
28	JOB_HISTORY	DEPARTMENT_ID	NUMBER(4,0)	Y	8	4	10	CHECK:5,PRIMARY KEY:1,FOREIGN KEY:3
29	LOCATIONS	LOCATION_ID	NUMBER(4,0)	N	8	4	23	CHECK:1,PRIMARY KEY:1,FOREIGN KEY:1
30	LOCATIONS	STREET_ADDRESS	VARCHAR2(40)	Y	8	4	23	CHECK:1,PRIMARY KEY:1,FOREIGN KEY:1
31	LOCATIONS	POSTAL_CODE	VARCHAR2(12)	Y	8	4	23	CHECK:1,PRIMARY KEY:1,FOREIGN KEY:1
32	LOCATIONS	CITY	VARCHAR2(30)	N	8	4	23	CHECK:1,PRIMARY KEY:1,FOREIGN KEY:1
33	LOCATIONS	STATE_PROVINCE	VARCHAR2(25)	Y	8	4	23	CHECK:1,PRIMARY KEY:1,FOREIGN KEY:1
34	LOCATIONS	COUNTRY_ID	CHAR(2)	Y	8	4	23	CHECK:1,PRIMARY KEY:1,FOREIGN KEY:1
35	REGIONS	REGION_ID	NUMBER	N	8	1	4	CHECK:1,PRIMARY KEY:1
36	REGIONS	REGION_NAME	VARCHAR2(25)	Y	8	1	4	CHECK:1,PRIMARY KEY:1

4.2 LightDB 目标端操作

创建精确统计表数量的方法，注意要在业务用户内创建，比如你迁移的是 hr 用户，那么要在 LightDB 端用 hr 用户登录进行创建，执行，另外如果你的用户名是“其他的”，那么脚本中的 hr 要对应替换掉

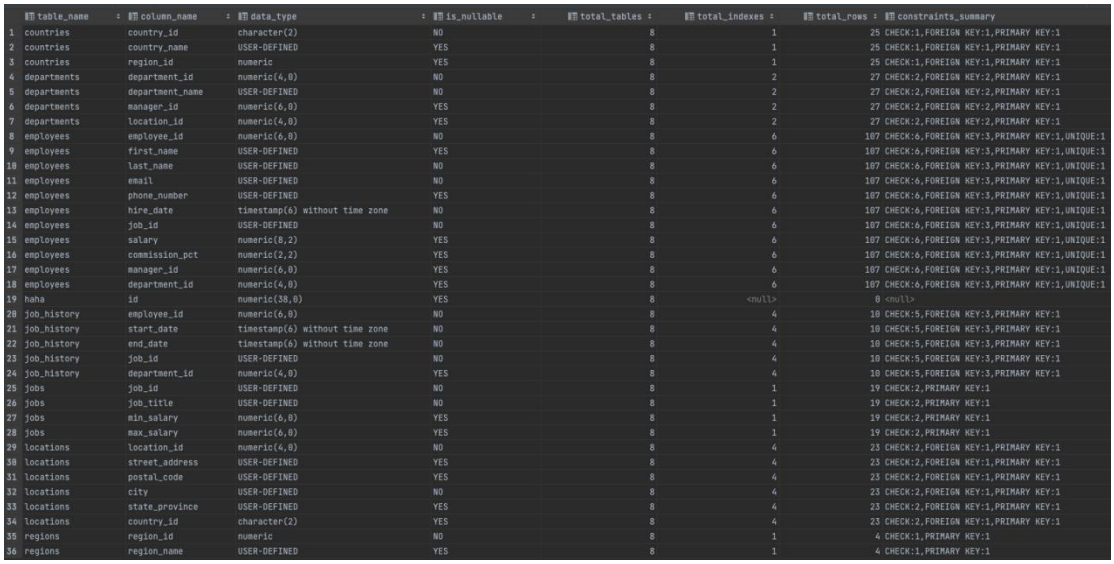
```
CREATE OR REPLACE FUNCTION get_table_data_counts()
RETURNS TABLE (
  table_name TEXT,
  total_rows BIGINT
) AS $$
DECLARE
  current_table_name TEXT;
  current_total_rows BIGINT;
  tables_cur CURSOR FOR SELECT t.table_name FROM information_schema.tables t
WHERE t.table_schema = 'hr';
BEGIN
  OPEN tables_cur;
  LOOP
    FETCH NEXT FROM tables_cur INTO current_table_name;
    EXIT WHEN NOT FOUND;

    EXECUTE format('SELECT COUNT(*) FROM %I', current_table_name) INTO
current_total_rows;

    table_name := current_table_name;
    total_rows := current_total_rows;
    RETURN NEXT;
  END LOOP;
```

```
  CLOSE tables_cur;
END; $$ LANGUAGE plpgsql;
```

4.2.1 进行查询统计

```
WITH table_count AS (
  SELECT COUNT(*) AS total_tables FROM information_schema.tables WHERE
table_schema = 'hr'
),
index_count AS (
  SELECT tablename, COUNT(*) AS total_indexes
  FROM pg_indexes
  WHERE schemaname = 'hr'
  GROUP BY tablename
),
constraint_count AS (
  SELECT table_name, string_agg(constraint_type || ':' ||
total_constraints::text, ',') AS constraints_summary
  FROM (
    SELECT table_name, constraint_type, COUNT(*) AS total_constraints
    FROM information_schema.table_constraints
    WHERE table_schema = 'hr'
    GROUP BY table_name, constraint_type
  ) subquery
  GROUP BY table_name
),
table_data_count AS (
  SELECT * FROM get_table_data_counts()
)
SELECT ic.table_name, ic.column_name,
  CASE
    WHEN ic.data_type = 'character varying' THEN ic.data_type || '(' ||
ic.character_maximum_length || ')'
    WHEN ic.data_type = 'numeric' AND ic.numeric_precision IS NOT NULL AND
ic.numeric_scale IS NOT NULL THEN ic.data_type || '(' || ic.numeric_precision
|| ',' || ic.numeric_scale || ')'
    WHEN ic.data_type = 'numeric' AND ic.numeric_precision IS NULL AND
ic.numeric_scale IS NULL THEN ic.data_type
    WHEN ic.data_type = 'character' THEN ic.data_type || '(' ||
ic.character_maximum_length || ')'
    WHEN ic.data_type = 'timestamp without time zone' THEN 'timestamp(' ||
ic.datetime_precision || ') without time zone'
```

```
    WHEN ic.data_type = 'timestamp with time zone' THEN 'timestamp(' ||
ic.datetime_precision || ') with time zone'
    WHEN ic.data_type = 'time without time zone' THEN 'time(' ||
ic.datetime_precision || ') without time zone'
    WHEN ic.data_type = 'time with time zone' THEN 'time(' || ic.datetime_precision
|| ') with time zone'
    WHEN ic.data_type = 'text' THEN ic.data_type
    WHEN ic.data_type = 'bigint' THEN ic.data_type
    WHEN ic.data_type = 'smallint' THEN ic.data_type
    WHEN ic.data_type = 'integer' THEN ic.data_type
    WHEN ic.data_type = 'bytea' THEN ic.data_type
    WHEN ic.data_type = 'real' THEN ic.data_type
    WHEN ic.data_type = 'double precision' THEN ic.data_type
    WHEN ic.data_type = 'boolean' THEN ic.data_type
    WHEN ic.data_type = 'date' THEN ic.data_type
    WHEN ic.data_type = 'uuid' THEN ic.data_type
    WHEN ic.data_type = 'json' THEN ic.data_type
    WHEN ic.data_type = 'money' THEN ic.data_type
    WHEN ic.data_type = 'interval' THEN ic.data_type
    ELSE ic.data_type
  END AS data_type,
  ic.is_nullable, tc.total_tables, idx.total_indexes, tdc.total_rows,
cc.constraints_summary
FROM information_schema.columns ic
JOIN table_count tc ON 1=1
LEFT JOIN index_count idx ON ic.table_name = idx.tablename
JOIN table_data_count tdc ON ic.table_name = tdc.table_name
LEFT JOIN constraint_count cc ON ic.table_name = cc.table_name
WHERE ic.table_schema = 'hr'
ORDER BY ic.table_name, ic.ordinal_position;
```



	table_name	column_name	data_type	is_nullable	total_tables	total_indexes	total_rows	constraints_summary
1	countries	country_id	character(2)	NO	8	1	25	CHECK:1,FOREIGN KEY:1,PRIMARY KEY:1
2	countries	country_name	USER-DEFINED	YES	8	1	25	CHECK:1,FOREIGN KEY:1,PRIMARY KEY:1
3	countries	region_id	numeric	YES	8	1	25	CHECK:1,FOREIGN KEY:1,PRIMARY KEY:1
4	departments	department_id	numeric(4,0)	NO	8	2	27	CHECK:2,FOREIGN KEY:2,PRIMARY KEY:1
5	departments	department_name	USER-DEFINED	NO	8	2	27	CHECK:2,FOREIGN KEY:2,PRIMARY KEY:1
6	departments	manager_id	numeric(6,0)	YES	8	2	27	CHECK:2,FOREIGN KEY:2,PRIMARY KEY:1
7	departments	location_id	numeric(4,0)	YES	8	2	27	CHECK:2,FOREIGN KEY:2,PRIMARY KEY:1
8	employees	employee_id	numeric(6,0)	NO	8	6	107	CHECK:6,FOREIGN KEY:3,PRIMARY KEY:1,UNIQUE:1
9	employees	first_name	USER-DEFINED	YES	8	6	107	CHECK:6,FOREIGN KEY:3,PRIMARY KEY:1,UNIQUE:1
10	employees	last_name	USER-DEFINED	NO	8	6	107	CHECK:6,FOREIGN KEY:3,PRIMARY KEY:1,UNIQUE:1
11	employees	email	USER-DEFINED	NO	8	6	107	CHECK:6,FOREIGN KEY:3,PRIMARY KEY:1,UNIQUE:1
12	employees	phone_number	USER-DEFINED	YES	8	6	107	CHECK:6,FOREIGN KEY:3,PRIMARY KEY:1,UNIQUE:1
13	employees	hire_date	timestamp(6) without time zone	NO	8	6	107	CHECK:6,FOREIGN KEY:3,PRIMARY KEY:1,UNIQUE:1
14	employees	job_id	USER-DEFINED	NO	8	6	107	CHECK:6,FOREIGN KEY:3,PRIMARY KEY:1,UNIQUE:1
15	employees	salary	numeric(8,2)	YES	8	6	107	CHECK:6,FOREIGN KEY:3,PRIMARY KEY:1,UNIQUE:1
16	employees	commission_pct	numeric(2,2)	YES	8	6	107	CHECK:6,FOREIGN KEY:3,PRIMARY KEY:1,UNIQUE:1
17	employees	manager_id	numeric(6,0)	YES	8	6	107	CHECK:6,FOREIGN KEY:3,PRIMARY KEY:1,UNIQUE:1
18	employees	department_id	numeric(4,0)	YES	8	6	107	CHECK:6,FOREIGN KEY:3,PRIMARY KEY:1,UNIQUE:1
19	haha	id	numeric(38,0)	YES	8	<null>	0	<null>
20	job_history	employee_id	numeric(6,0)	NO	8	4	10	CHECK:5,FOREIGN KEY:3,PRIMARY KEY:1
21	job_history	start_date	timestamp(6) without time zone	NO	8	4	10	CHECK:5,FOREIGN KEY:3,PRIMARY KEY:1
22	job_history	end_date	timestamp(6) without time zone	NO	8	4	10	CHECK:5,FOREIGN KEY:3,PRIMARY KEY:1
23	job_history	job_id	USER-DEFINED	NO	8	4	10	CHECK:5,FOREIGN KEY:3,PRIMARY KEY:1
24	job_history	department_id	numeric(4,0)	YES	8	4	10	CHECK:5,FOREIGN KEY:3,PRIMARY KEY:1
25	jobs	job_id	USER-DEFINED	NO	8	1	19	CHECK:2,PRIMARY KEY:1
26	jobs	job_title	USER-DEFINED	NO	8	1	19	CHECK:2,PRIMARY KEY:1
27	jobs	min_salary	numeric(6,0)	YES	8	1	19	CHECK:2,PRIMARY KEY:1
28	jobs	max_salary	numeric(6,0)	YES	8	1	19	CHECK:2,PRIMARY KEY:1
29	locations	location_id	numeric(4,0)	NO	8	4	23	CHECK:2,FOREIGN KEY:1,PRIMARY KEY:1
30	locations	street_address	USER-DEFINED	YES	8	4	23	CHECK:2,FOREIGN KEY:1,PRIMARY KEY:1
31	locations	postal_code	USER-DEFINED	YES	8	4	23	CHECK:2,FOREIGN KEY:1,PRIMARY KEY:1
32	locations	city	USER-DEFINED	NO	8	4	23	CHECK:2,FOREIGN KEY:1,PRIMARY KEY:1
33	locations	state_province	USER-DEFINED	YES	8	4	23	CHECK:2,FOREIGN KEY:1,PRIMARY KEY:1
34	locations	country_id	character(2)	YES	8	4	23	CHECK:2,FOREIGN KEY:1,PRIMARY KEY:1
35	regions	region_id	numeric	NO	8	1	4	CHECK:1,PRIMARY KEY:1
36	regions	region_name	USER-DEFINED	YES	8	1	4	CHECK:1,PRIMARY KEY:1

五、总结与说明

1．如果使用 ZIP 压缩版的 Instant Client，环境变量 LD_LIBRARY_PATH 和 ORACLE_HOME 需要设置为相同的值，也就是安装文件的所在目录
2．先导入表结构，然后在导入数据；
3．注意 LightDB 的保留的关键字是否为 oracle 表字段名；
4．所有外键约束最后导入。
5．直接通过 ora2pg 迁移数据，不要将 oracle 表数据导出为 csv 格式，然后再导入 LightDB 数据库。
6．如果遇到源端 Oracle 字段为 ctid 等 LightDB 关键字需要特殊处理
参考链接：https://ora2pg.darold.net/index.html