# GreenPlum 数据操作指南

## GreenPlum 客户端连接

GreenPlum官方说明，使用方式和Postgres相同

使用Navicat Premium 12 可以通过postgres进行访问数据库，Navicat比较通用使用者比较熟悉

功能更丰富的可用DBeaver有GreenPlum专用配置，可使用其操作复杂操作

## Mysql表结构迁移至GreenPlum中

### 1、通过Navicat Premium将变在本地中进行转换，后导出到GP中

第一步，确保电脑上装有这两个数据库。

第二步，下载一个管理数据库的软件，如navicat premium等。

第三步，此步开始有截图。

选中要转移的表，拖拽至另一个数据库想要存放的位置，或选中要转移的表，复制，到另一个数据库里粘贴，

第四步，之后会出现一个名为“数据传输”的窗口，在这里可以根据需求选多张表到另一个数据库的指定位置，之后点击开始，点击确认。之后会自动传输数据表。

第五步，完成！若电脑是mysql，想把里面的表转移到其他电脑或服务器中其他数据库里，按上述操作，先把表本地转移成功，在转移后的数据库到处表结构sql文件，拷到需要的设备上，运行sql即可，运行最好在右击数据库名运行。

### 2、通过工具进行

参考：https://www.alibabacloud.com/help/zh/doc-detail/35458.htm

操作：

1下载mysql2pgsql二进制安装包

2解压

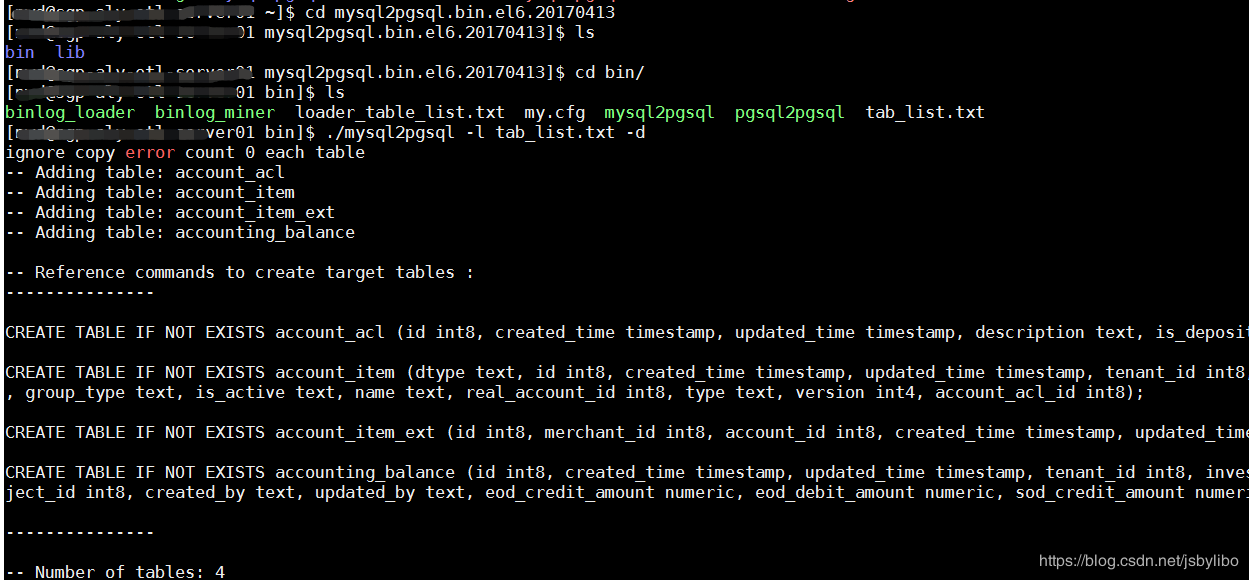
3修改my.cfg参数

4创建文本文件，存放表名

5执行

6源库有多个的话，重复1-5步骤

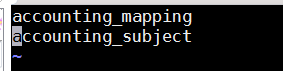
结果：



配置：my.cfg



tab\_list.txt



————————————————

## GreenPlum 数据导入

GreenPlum

### gpfdist工具外部表导入方式（速度最快）

可通过外部表方式进行数据导入，现在支持text和cvs两种格式

#### mysql数据导出成文csv文件

时间格式YMD，如果导出成txt时间格式容易导入失败。

#### 2、将导出的数据文本文件上传到gp主服务器的指定文件

/var/gpw

#### 3、开启外部表

指定上传文件目录和端口

gpfdist -d /data01/gpadmin/gpw -p 8081 &

#### 4、创建外部表

注意数据类型，修改对应的ip和端口

create external table test.user.archive\_feature\_wb (

like test.user.archive\_feature

)

LOCATION (

'gpfdist://192.168.206.180:8081/archive\_feature.csv'

)

FORMAT 'csv' ( delimiter ',' null '' escape '"' quote '"' )

ENCODING 'UTF8';

#### 外部表导入到gp数据库

导入前将

insert into public.monitoring\_12

select \* from public.monitoring\_12\_w

**insert** **into** "user".sys\_role\_permission\_nb2

**select** \* **from** test."user".sys\_role\_permission\_wb

#### 数据清理

--清理表空间

**vacuum** **full** "user".archive\_feature;

#### 7、收集执行计划

--收集标新优化执行计划

**analyze** "user".archive\_feature;

### 通过第三方工具dbsync直接数据库导入到Greenplum

## [Postgresql数据库常用操作命令及SQL语言](https://www.cnblogs.com/EikiXu/p/9698768.html)

### （1）登录

peng@peng-virtual-machine:~$ sudo -u postgres psql

以用户postgres身份登录，postgres为用户名，可有多个用户，登录时会要求输入相应密码，之后就会进入psql环境了

### （2）切换数据库

有时候需要在psql环境下切换数据库，此时执行如下psql命令：  
\c dbname username serverIP port  
其中除了数据库名外，其他的参数都是可选的，如果使用默认值可以使用-作为占位符  
执行这个命令后，也是提示输入密码。

（3）查看帮助

psql提供了很好的在线帮助文档，总入口命令是help，输入这个命令就可以看到  
vsb9=# help  
You are using psql, the command-line interface to PostgreSQL.  
Type:  \copyright for distribution terms  
       \h for help with SQL commands（查看SQL命令帮助)  
       \? for help with psql commands（查看psql命令帮助）  
       \g or terminate with semicolon to execute query（\g表示命令已经输入完成，开始执行查询或操作任务）  
         
       例如上面截图中select id from courses之后#前变"-"表示还在等待输入命令，这一次命令没输入完整，而输入\g就告诉它开始执行吧。select id from courses;（注意这里加了“；”）表示输入完成开始执行。这两种得到的结果是一致的都是返回表中id属性的值。  
       \q to quit（退出登录）

可以看到，标准SQL命令的帮助和psql特有命令的帮助是分开的。输入\?查看psql命令，会发现所有的psql命令都是以\开头，这就很容易和标准的SQL命令进行区分开来。

### 常用数据库命令（mysql为MySQL数据库操作命令，psql为postgresql数据库命令）

1列出所有的数据库  
mysql: show databases  
psql: \l或\list

2切换数据库  
mysql: use dbname  
psql: \c dbname

3列出当前数据库下的所有表  
mysql: show tables   
psql: \d

4列出指定表的所有字段  
mysql: show columns from table name  
psql: \d tablename

### 5查看表的基本情况

mysql: describe tablename  
psql: \d+ tablename

SQL常用命令  
在切换到某数据库环境后：  
select attr from tablename;（加分号立即执行该语句，该表中的属性值，这个表应当是当前数据库中存在的表）  
select attr  
from tablename;（这样也是可以的，二者效果等同，这一形式在其他情况下会用到）

### 更多

：[https://www.cnblogs.com/kaituorensheng/p/4667160.html#\_label20](https://www.cnblogs.com/kaituorensheng/p/4667160.html" \l "_label20)

## 集群性能检测

### greenplum gpcheckperf 命令使用

gpcheckperf -f seg\_host -d /export/gpdata –v

### 详情

<https://cloud.tencent.com/developer/article/1429770>

## 操作QA

### 如何查看Greenplum各节点的数据情况

**select** gp\_segment\_id,**count**(\*) **from** table\_name **group** **by** gp\_segment\_id;

### Greenplum 如何直连segment节点

$ PGOPTIONS='-c gp\_session\_role=utility' psql -p 40005

直连greenplum segment节点的方法, utility模式 :

使用这种方式，不与其他节点通讯，只操作当前节点。也没有数据分布的概念。

如果使用utility模式连接的是master节点，写入数据时，数据不会分布到segment，使用正常模式连接后，写入master的数据也查不出来。

$ PGOPTIONS='-c gp\_session\_role=utility' psql -p 40005

psql (8.2.15)

Type "help" for help.

postgres=# \dt

List of relations

Schema | Name | Type | Owner | Storage

--------+------+-------+----------+---------

public | t | table | digoal | heap

public | test | table | digoal | heap

(2 rows)

postgres=# select \* from pg\_locks;

locktype | database | relation | page | tuple | transactionid | classid | objid | objsubid | transaction | pid | mode | granted | mppsessionid | mppiswriter | gp\_segment\_id

---------------+----------+----------+------+-------+---------------+---------+-------+----------+-------------+--------+-----------------+---------+--------------+-------------+---------------

transactionid | | | | | 136604 | | | | 136604 | 130724 | ExclusiveLock | t | 6 | t | -1

relation | 10899 | 10333 | | | | | | | 136604 | 130724 | AccessShareLock | t | 6 | t | -1

(2 rows)

注意使用PGOPTIONS='-c gp\_session\_role=utility'后，只操作本地节点

$psql -p # master

psql (8.2.15)

Type "help" for help.

postgres=# select count(\*) from test;

count

-------

0

(1 row)

postgres=# \q

$psql -p 40001 # segment

psql (8.2.15)

Type "help" for help.

postgres=# select count(\*) from test;

count

---------

4166801

(1 row)

文件空间

### 修改分布键

**alter** **table** tabname **set** distributed **by**(**name**);