ELK数据倾斜测试

# 测试环境

进入ELK：

cd ${ficlient}

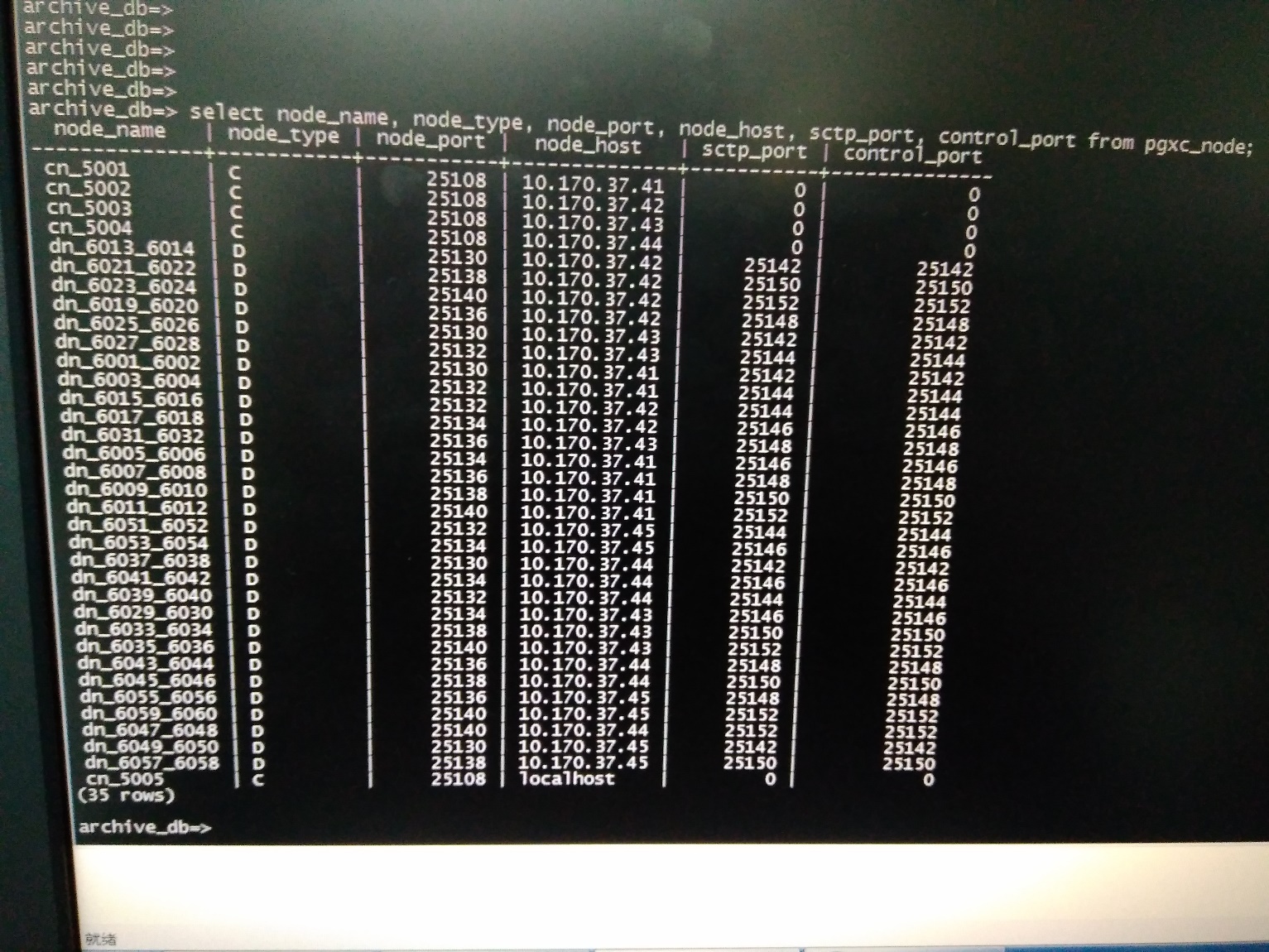
. bigdata\_env

gsql –d archive\_db –h 10.170.37.45 –p 25108 –U hds –W hds@1234

## 平台环境

查看节点分布表：

select node\_name, node\_type, node\_port, node\_host, sctp\_port, control\_port from pgxc\_node;

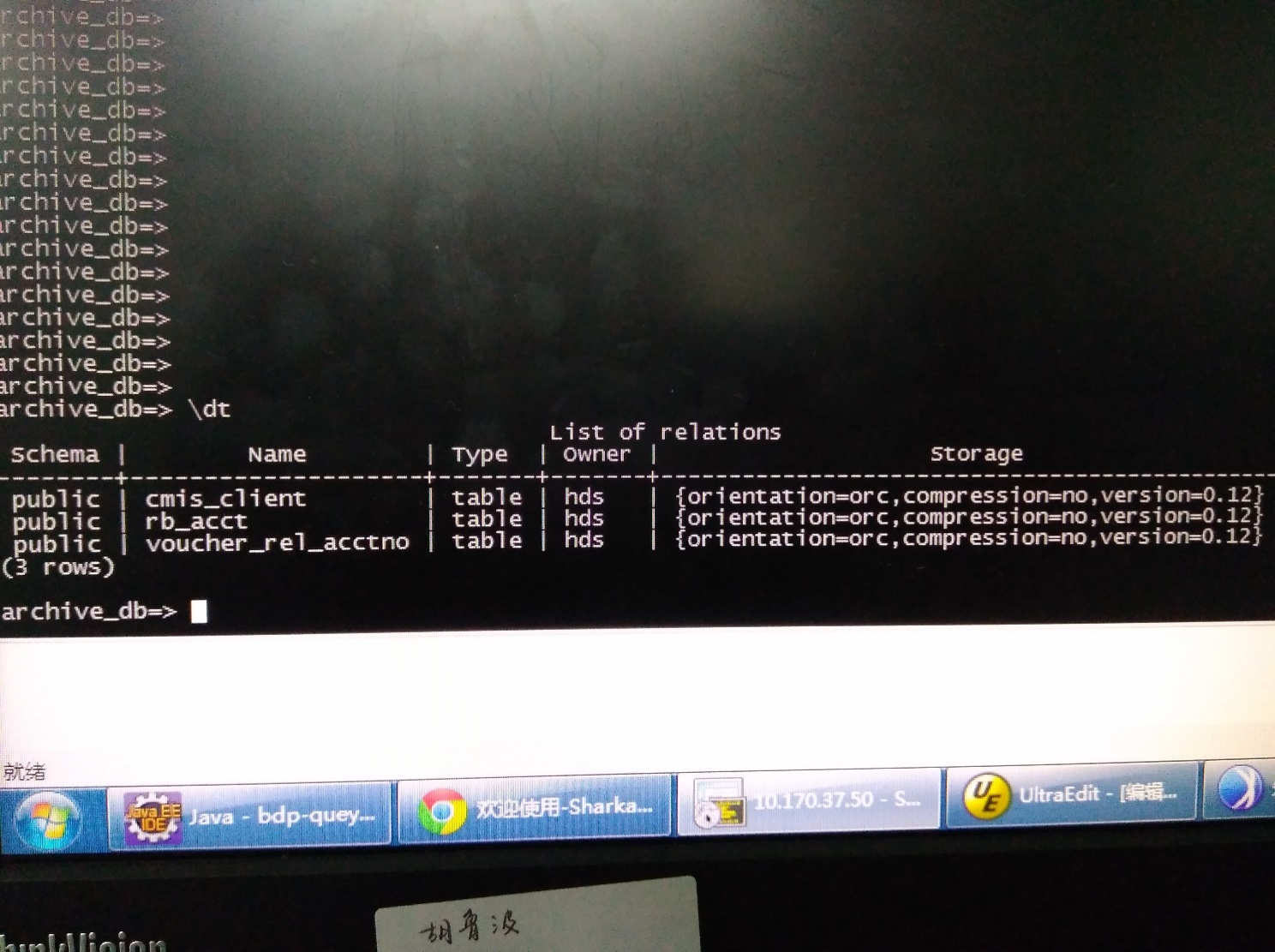


共有35个node结点，其中ControlNode节点5，DataNode节点30个。

## 数据量

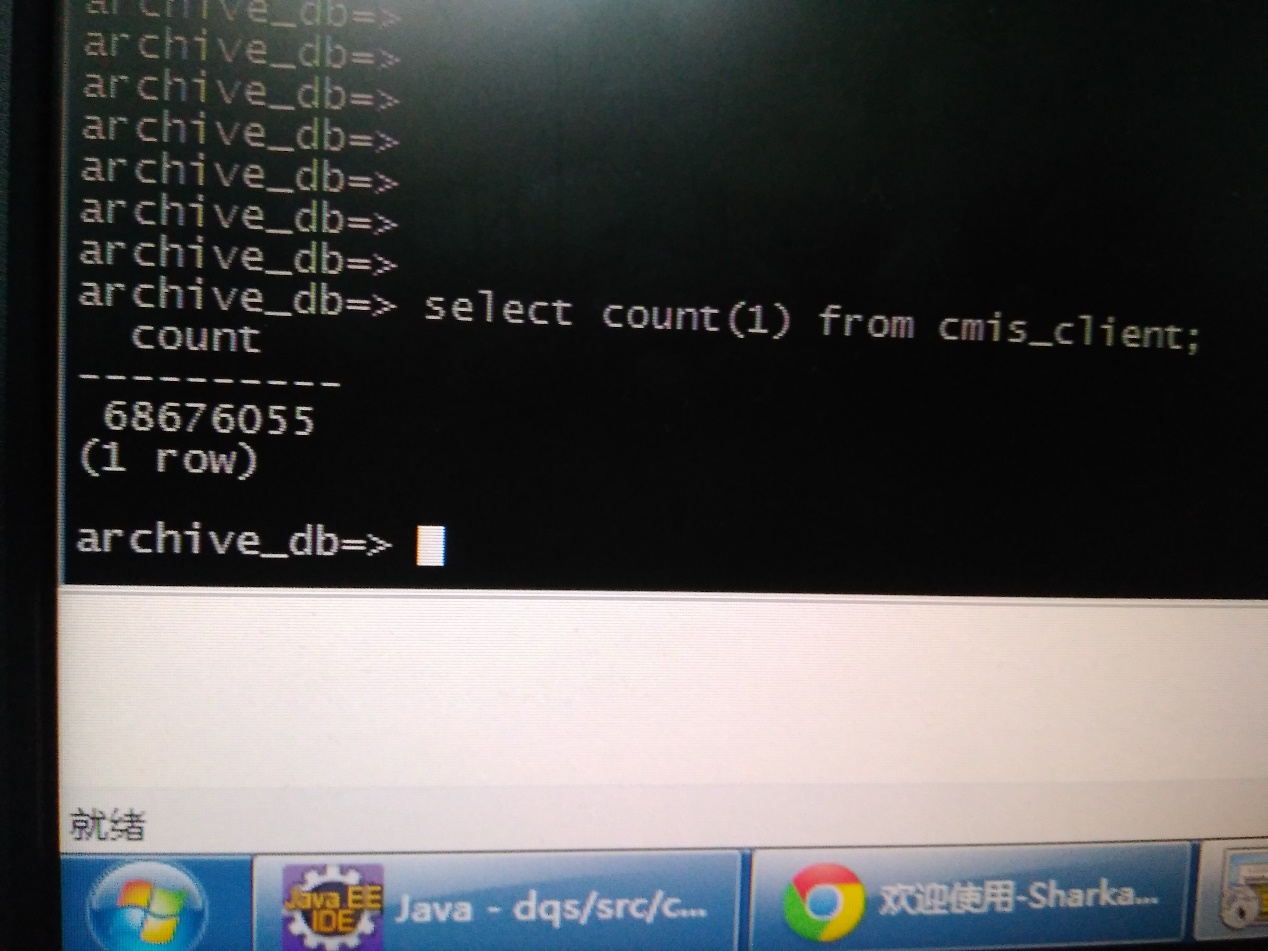
查看所有内表

\dt



查看cmis\_client表的数据量：

Select count(1) from cmis\_client;



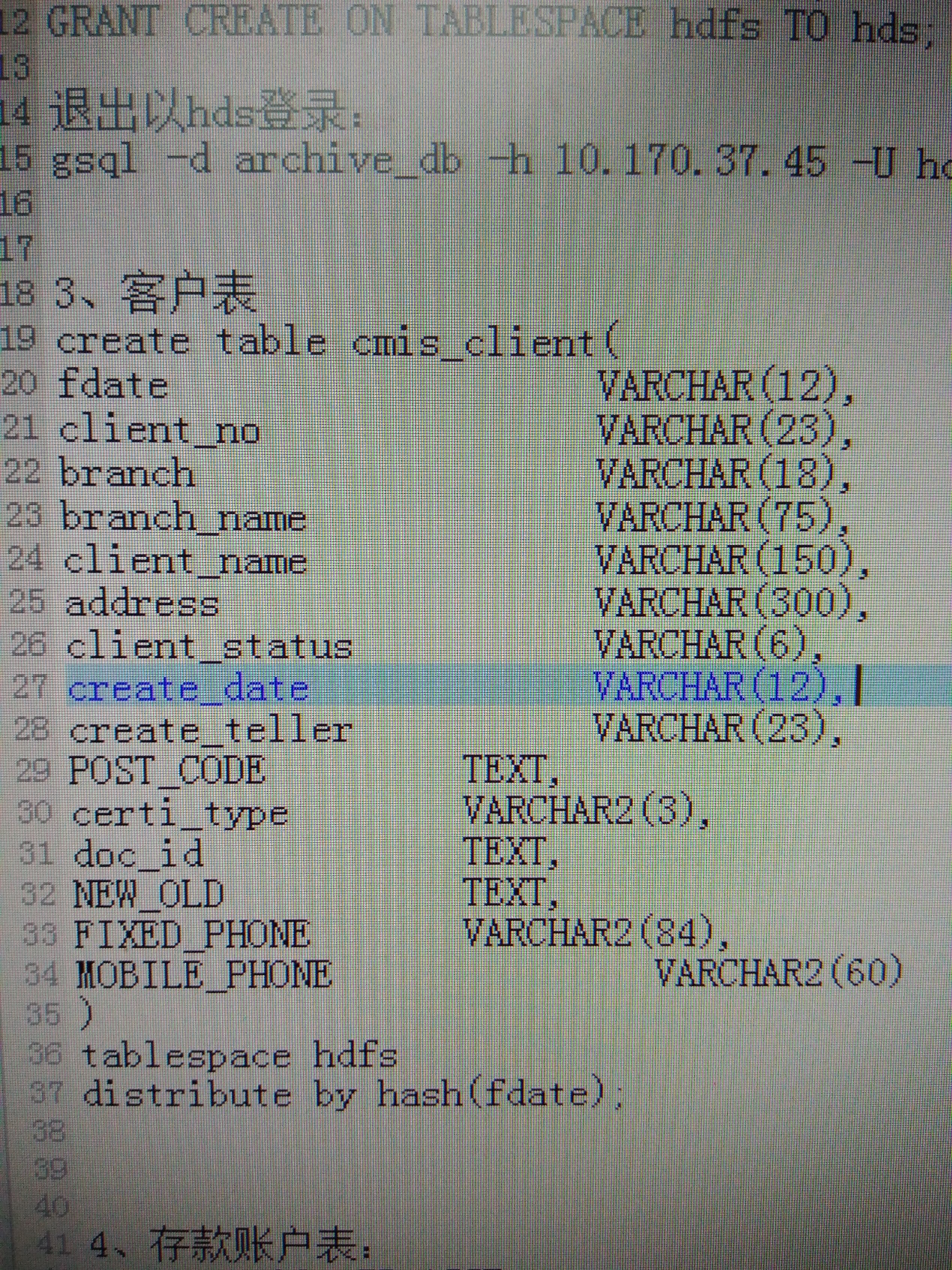
该表有6千多万条数据。

# 测试步骤

## 建表及初始化

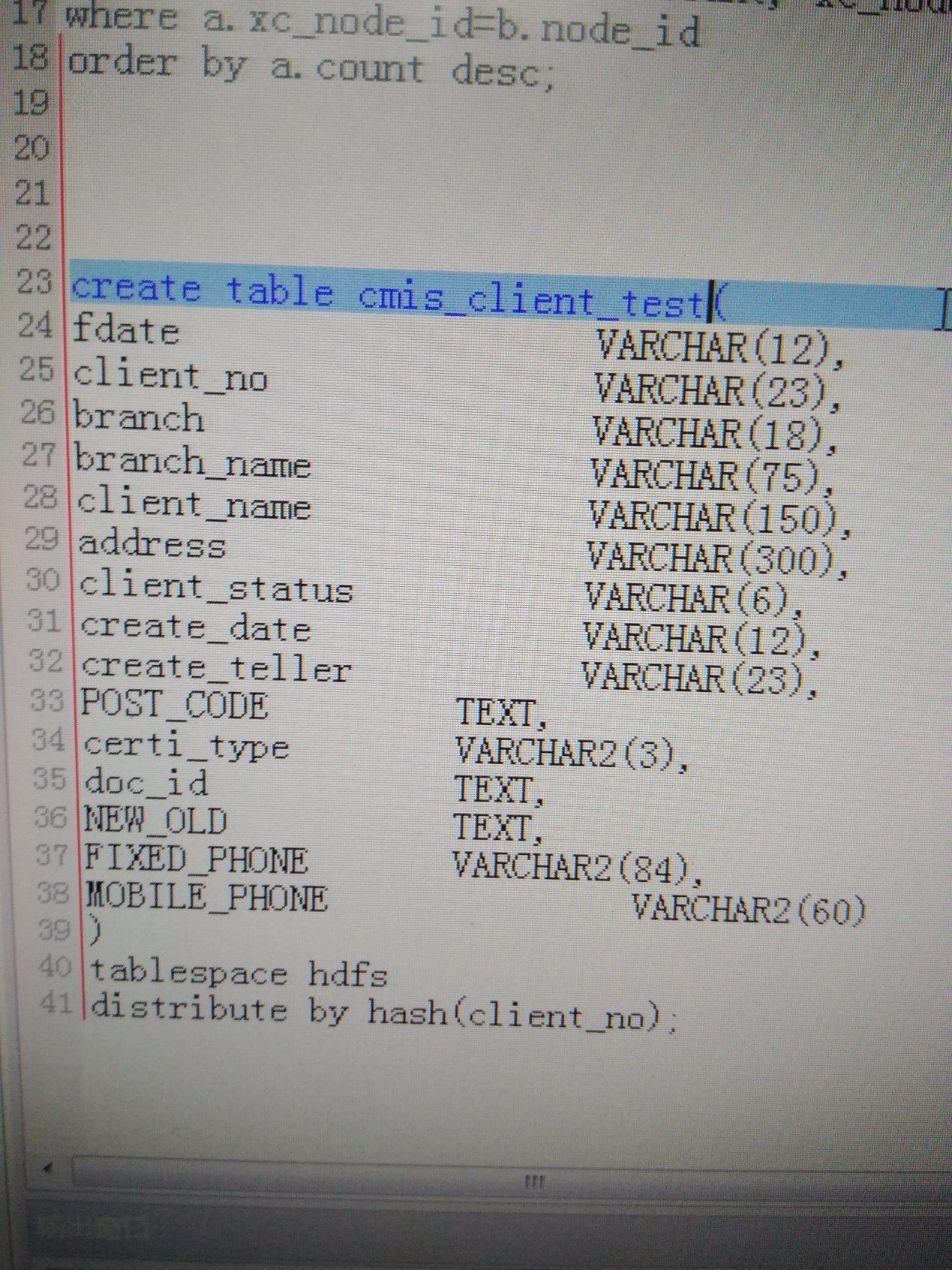
### cmis\_client

cmis\_client客户表是通过卸数、归档、提数到ELK的，以下是建表语句，按fdate的hash值分散。



### cmis\_client

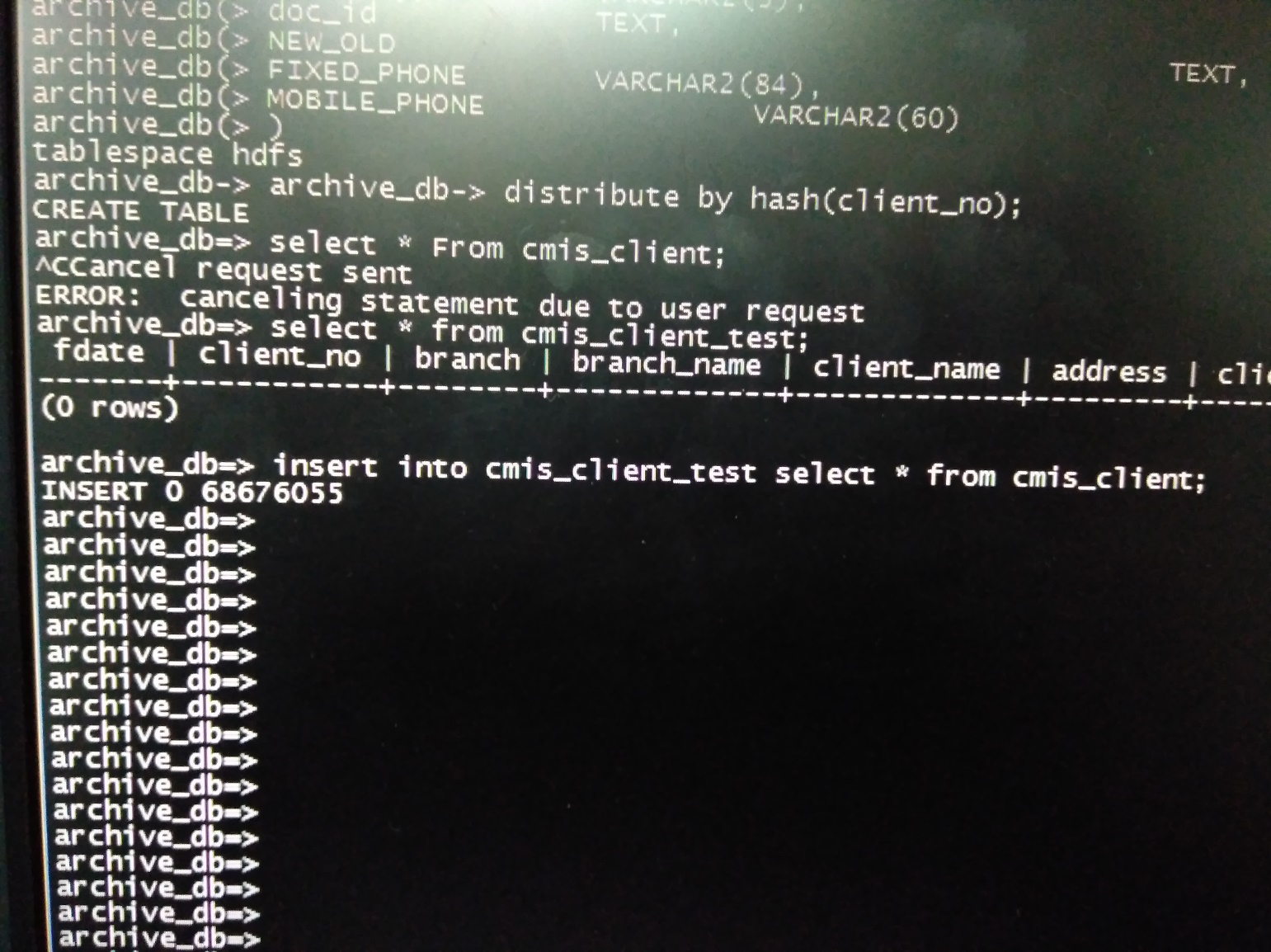
以下生成cmis\_client\_test表，以client\_no的hash值分散。



### 初始化

将cmis\_cient六千多万条数据导入到cmis\_cient\_test中。

Insert into cmis\_client\_test select \* from cmis\_client;



## 分析分布情况

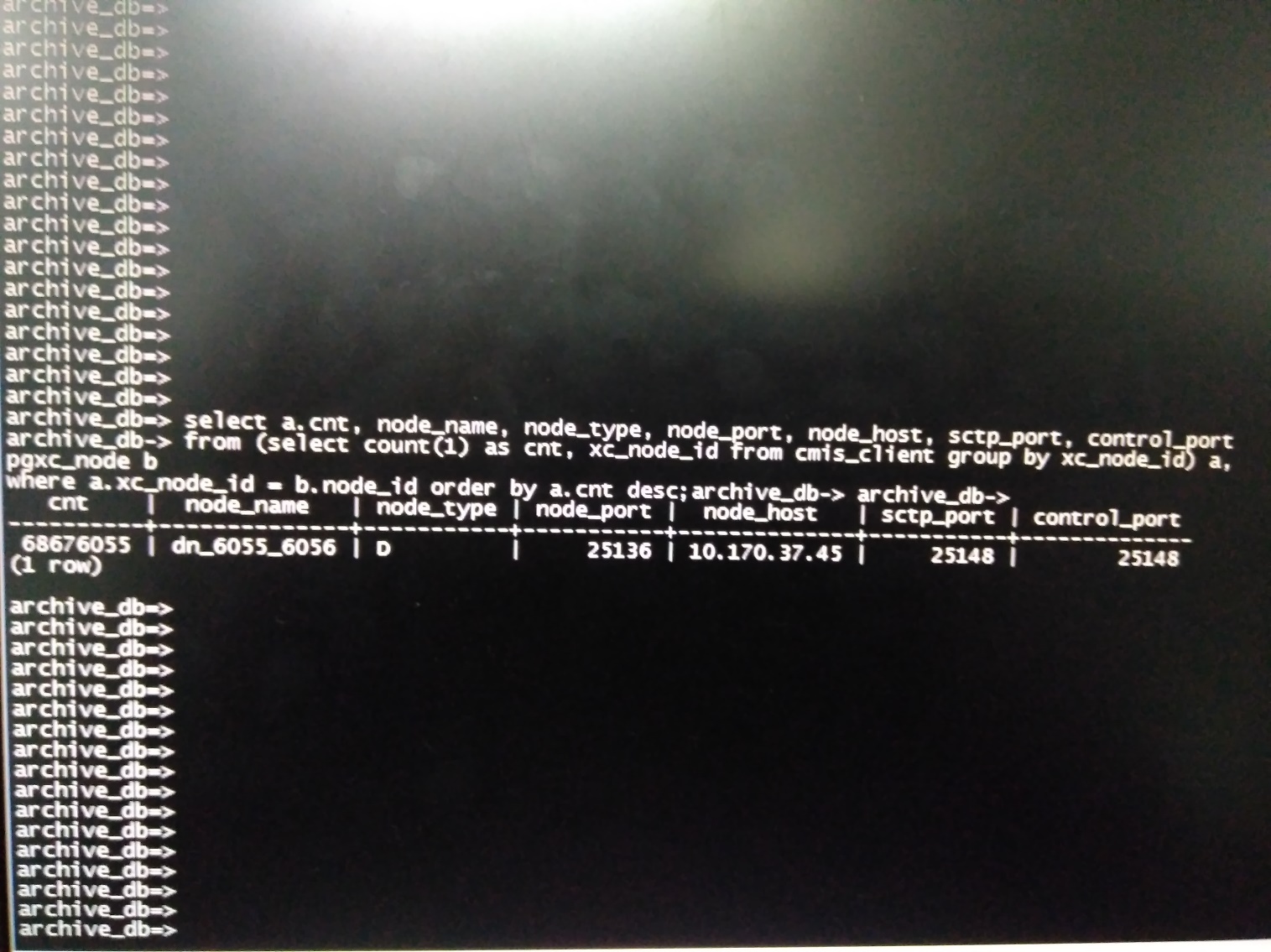
查看cmis\_client表的分布情况：

select a.cnt, node\_name, node\_type, node\_port, node\_host, sctp\_port, control\_port

from (select count(1) as cnt, xc\_node\_id from cmis\_client group by xc\_node\_id) a,

pgxc\_node b

where a.xc\_node\_id = b.node\_id order by a.cnt desc;



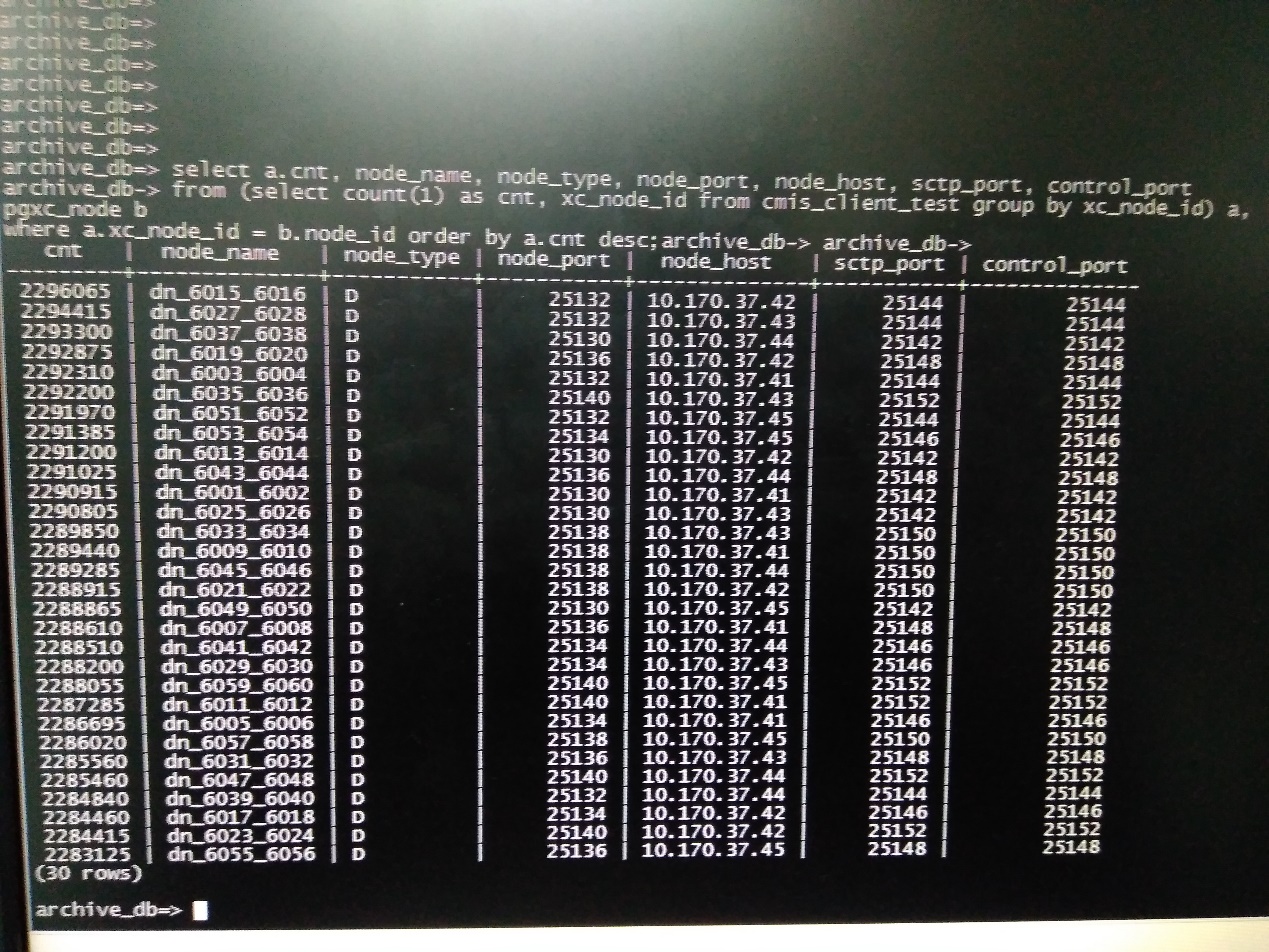
查看cmis\_client\_test表的分布情况：

select a.cnt, node\_name, node\_type, node\_port, node\_host, sctp\_port, control\_port

from (select count(1) as cnt, xc\_node\_id from cmis\_client\_test group by xc\_node\_id)

a, pgxc\_node b

where a.xc\_node\_id = b.node\_id order by a.cnt desc;



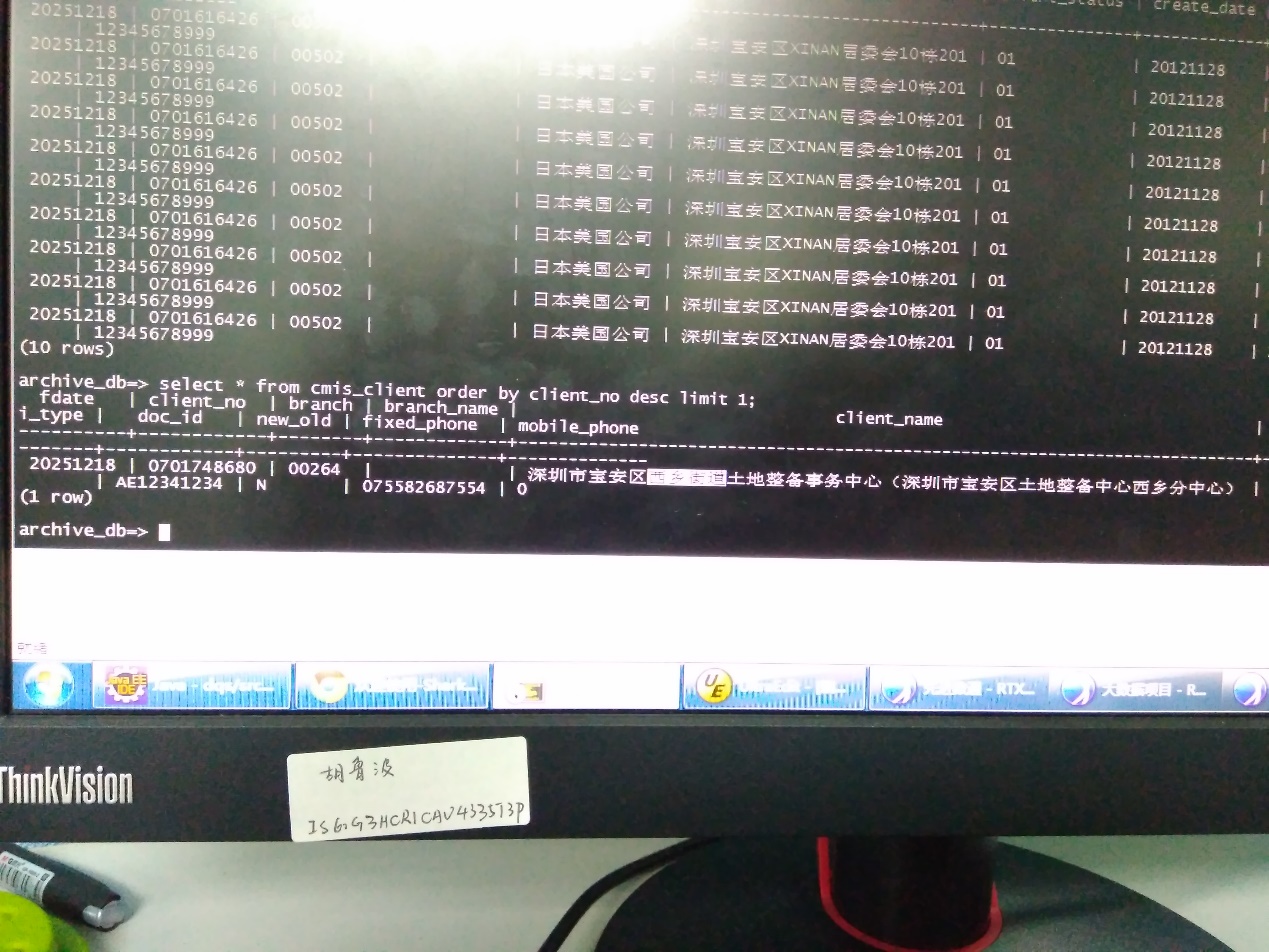
### 总结

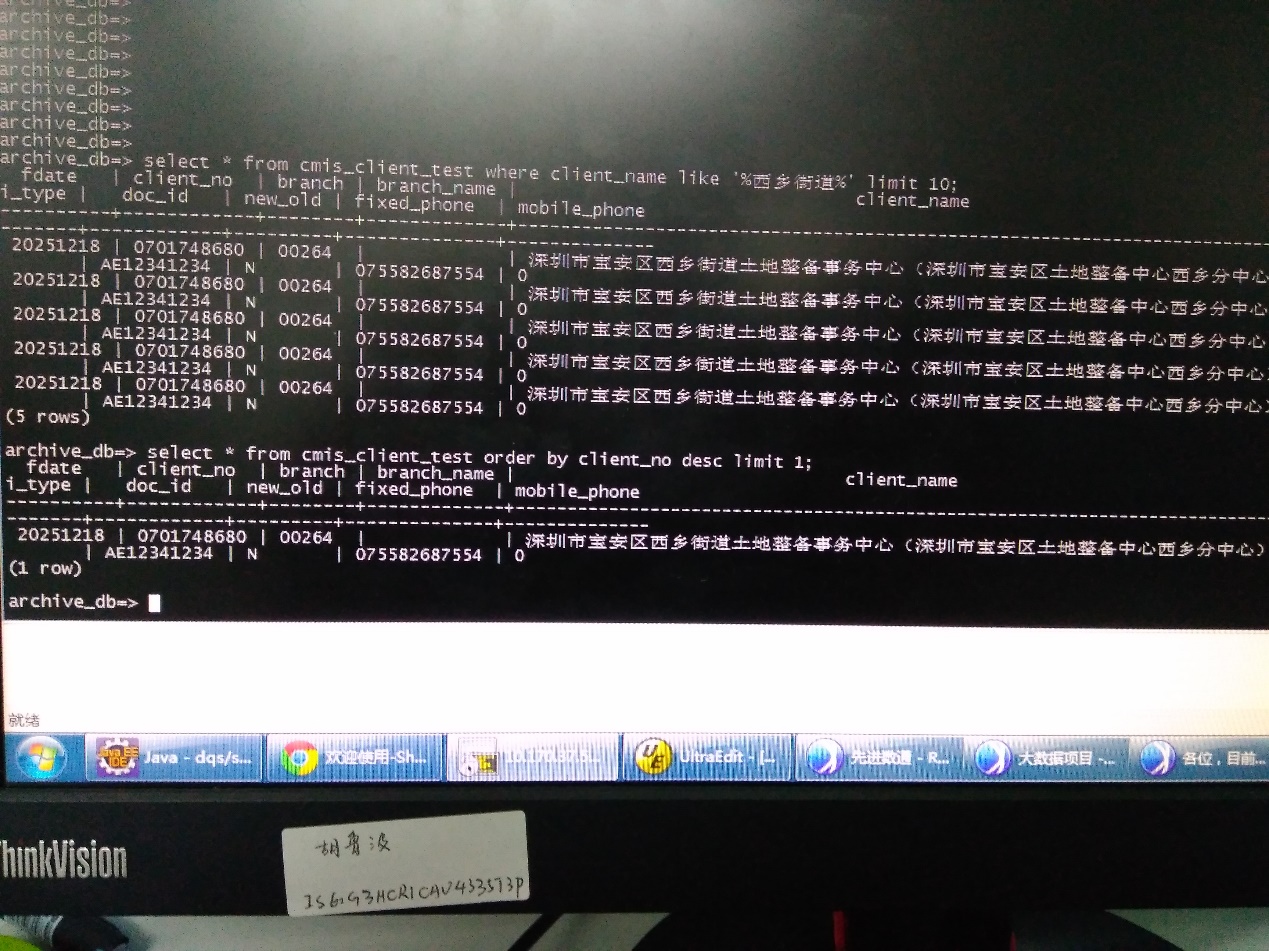
cmis\_client以fdate的hash值分散，同一个日期的数据全部集中在一个结点上，造成数据倾斜；cmis\_client\_test是按client\_no的hash值分布，数据能平均分散到各个节点，实现负载均衡，避免造成数据倾斜。

## 查询速度

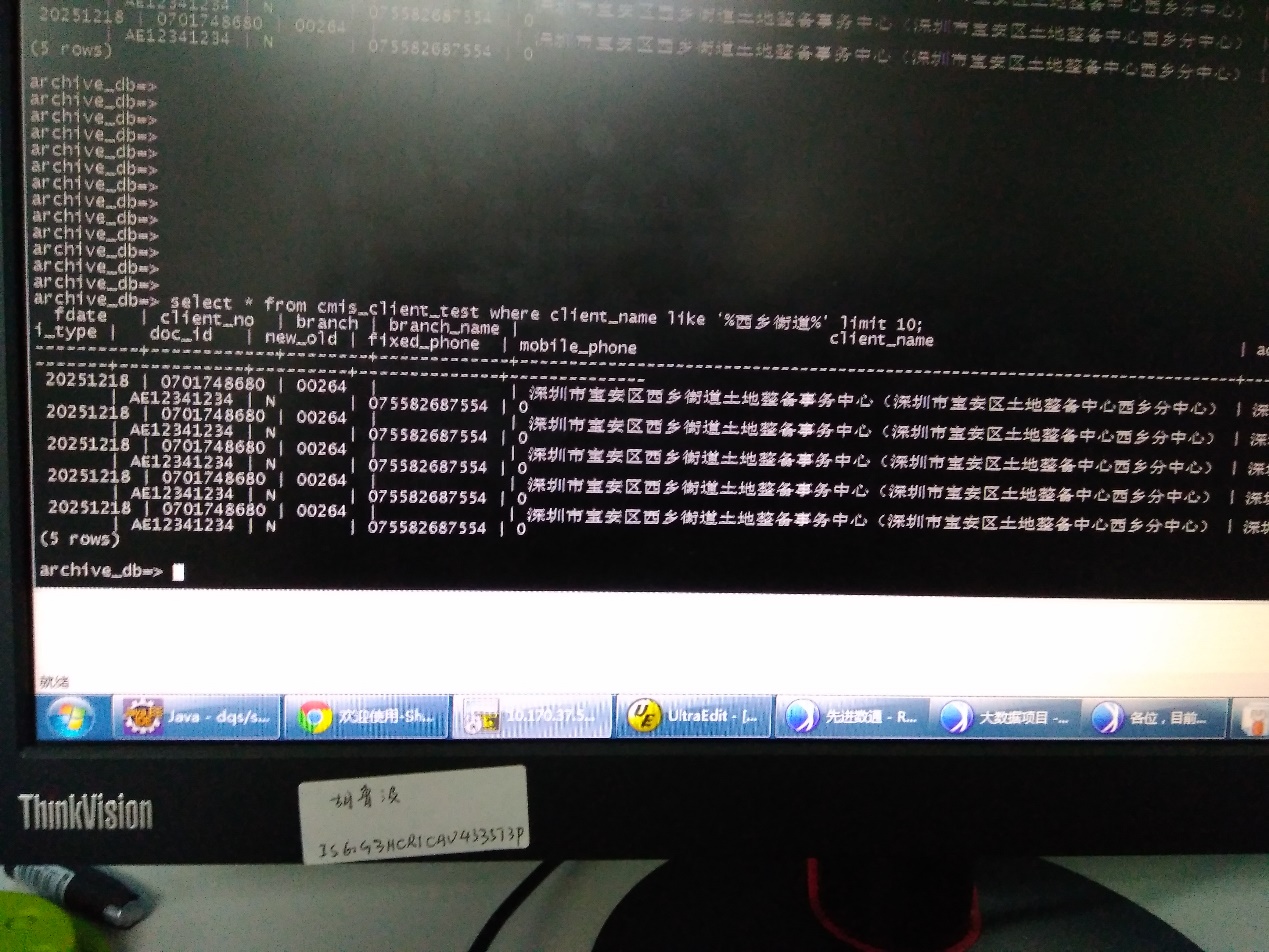
查询正常排序的最后一条数据，对比查询速度。

查询出两张表的最后一条数据，都是client\_name包含”西乡街道”



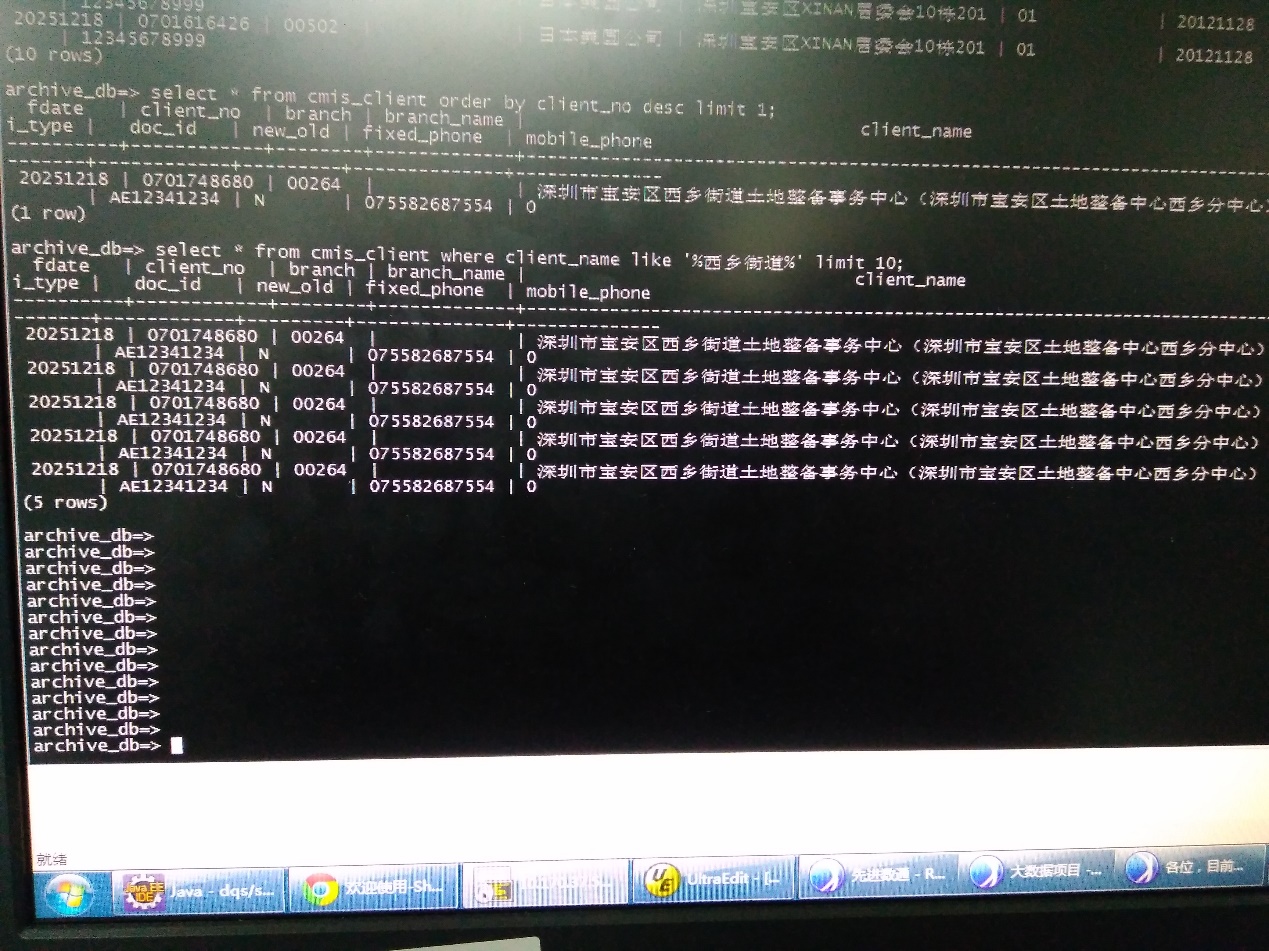


### Cmis\_client表



**查询时间：01:24:6**

### Cmis\_client\_test表



**查询时间：00:04:7**

# 结论

如果ELK表distribute by 字段值基本一样，没有分散的话，数量会集中分布在某几个节点上，造成负载不均衡，数据倾斜的现象，最后在查询数据时会变得很慢，正常情况况ELK查询表的速度应该是在秒级下，测试结果发现，数据倾斜后查询速度达到几分钟，并且数据量越大，查询越慢。

为避免数据倾斜，应正确选择distribute by 字段，尽量选择分散的字段，当字段值都基本一致时，其hash值是一样的，可以按随机数来分散，可以大大提高查询速度。