Vending迁移方案

（南非Centlec现场）

迁移前注意事项：app.properties中Customization需要配置成vending,centlec

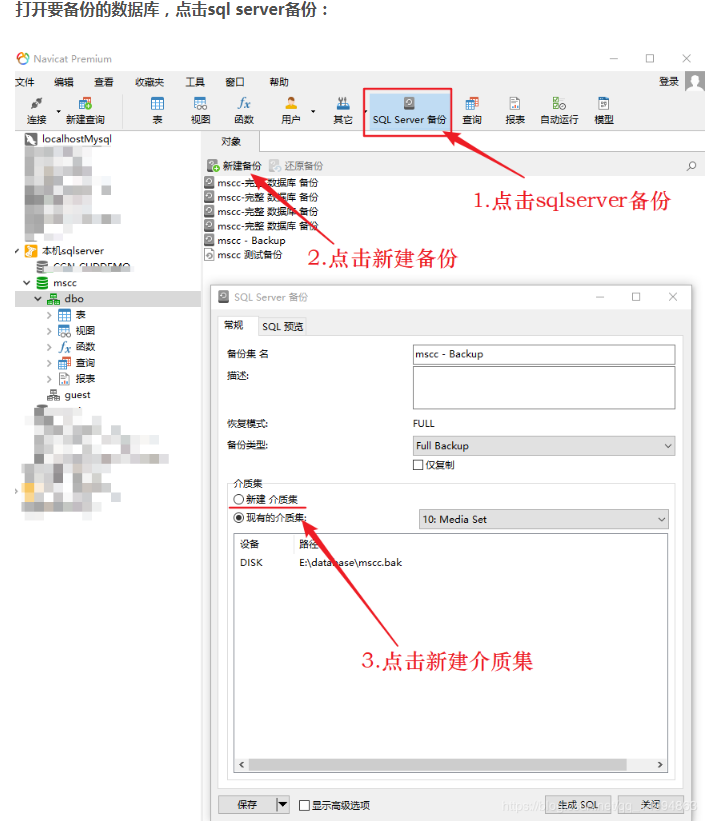
# 一、迁移范围

数据迁移从SQLserver至mysql，范围见附件《需求范围.xlsx》

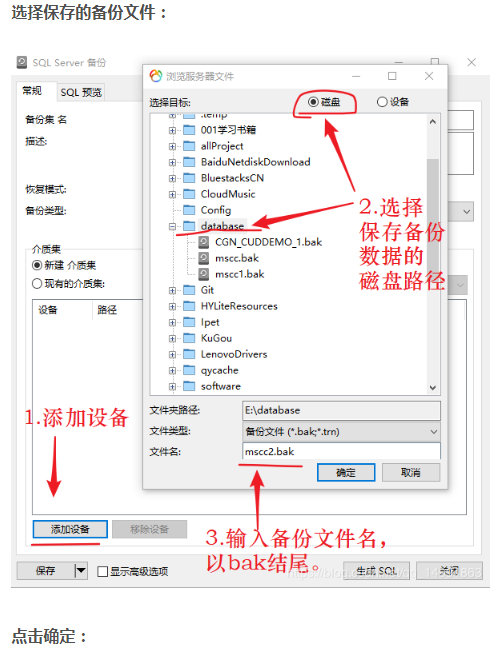
# 二、DB与Navicat

## 1、SQLserver备份与还原

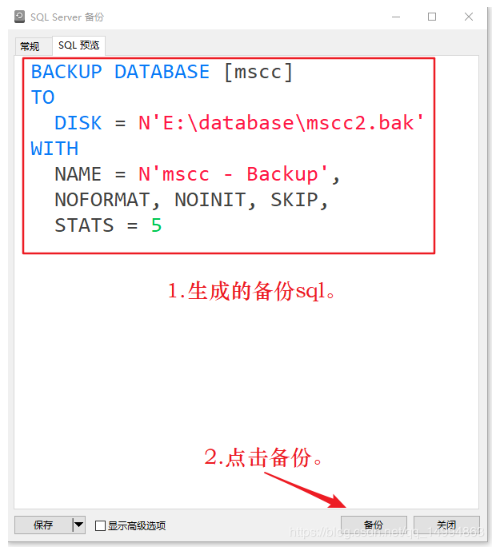
### 2.1.1打开要备份的数据库，点击sql server备份



### 2.1.2选择保存的备份文件



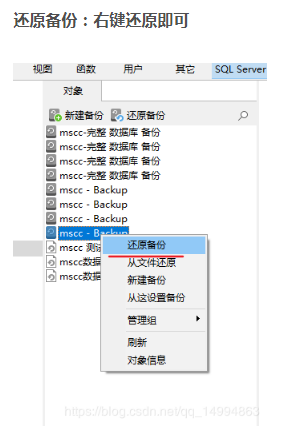
### 2.1.3生成备份SQL



### 2.1.4保存

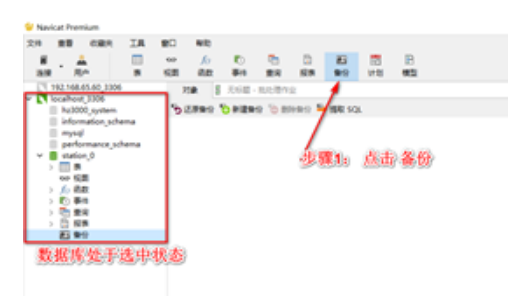


### 2.1.5还原

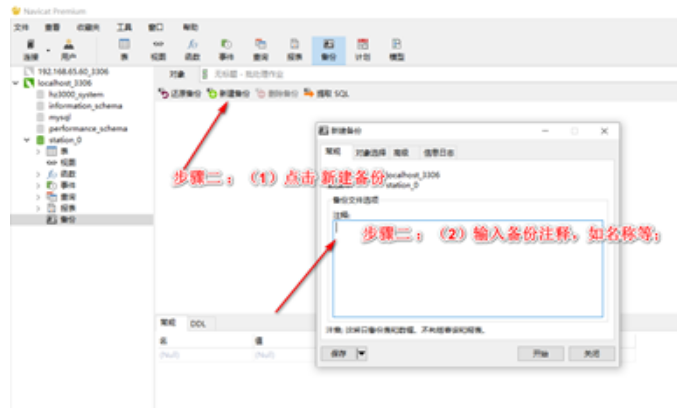


## 2、mysql备份与还原

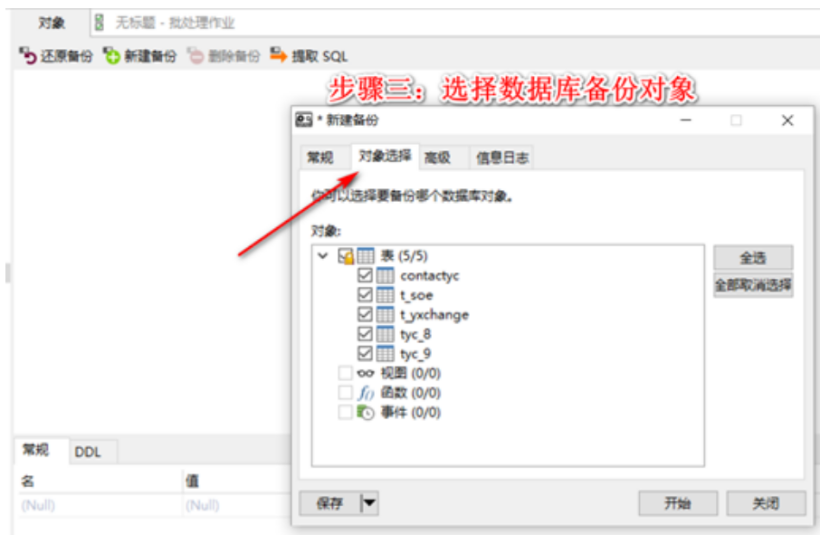
### 2.2.1手动备份步骤1



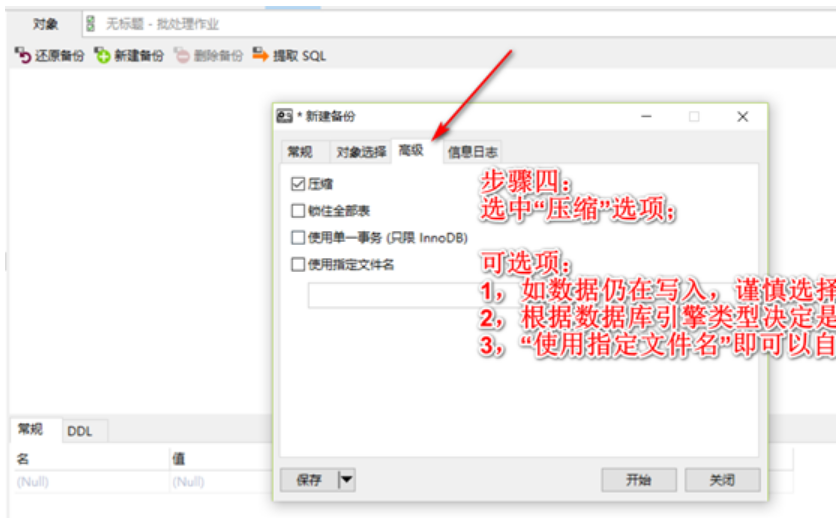
### 2.2.2手动备份步骤2



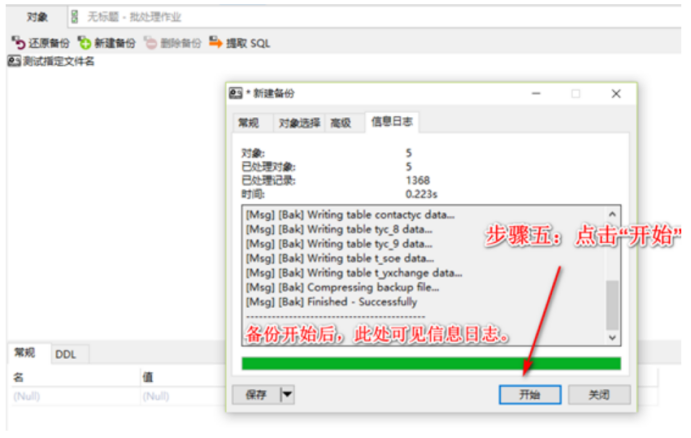
### 2.2.3手动备份步骤3



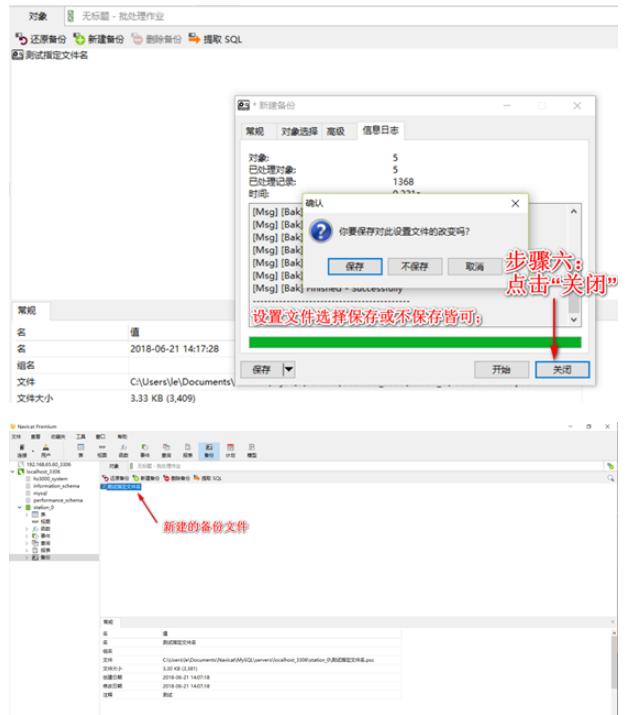
### 2.2.4手动备份步骤4



### 2.2.5手动备份步骤5



### 2.2.6手动备份步骤6

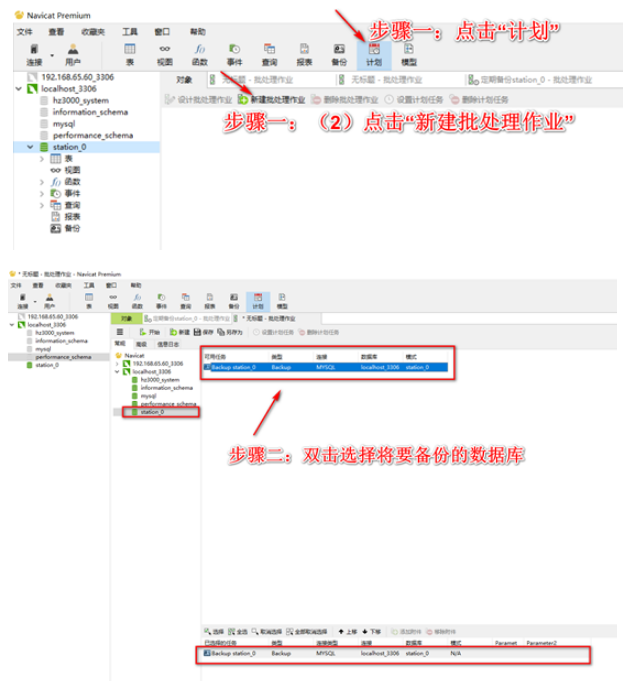


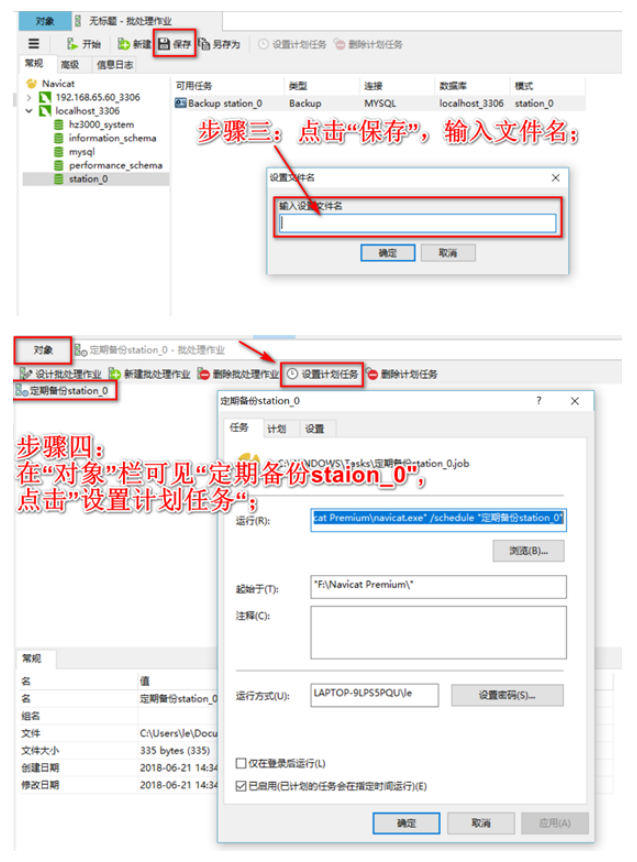
### 2.2.7自动备份

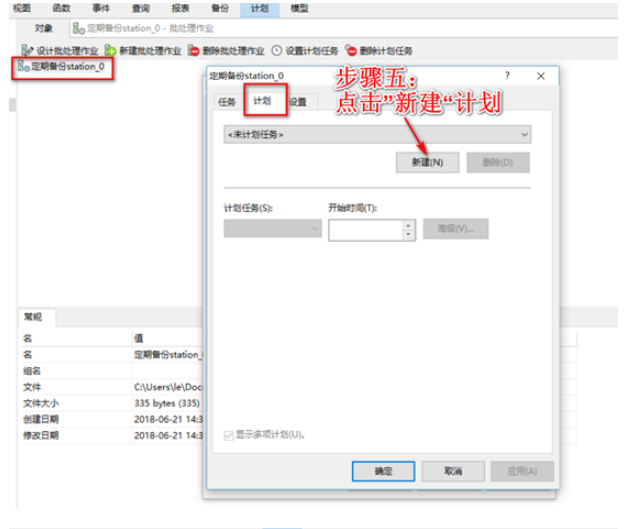
完整备份一般一段时间进行一次，且在网站访问量最小的时候，这样常借助批处理文件定时备份。

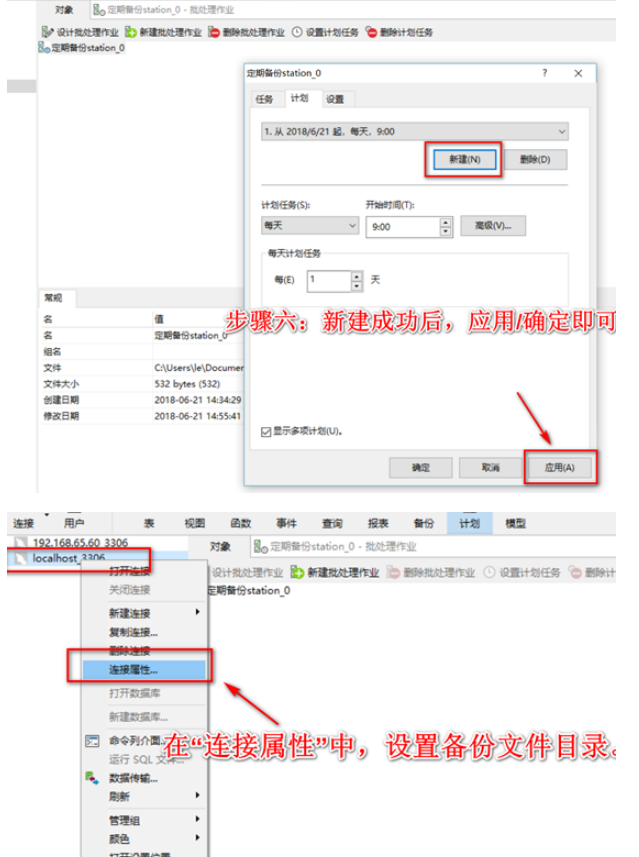
主要原理是写一个批处理文件，在里面写上处理程序的绝对路径然后把要处理的东西写在后面。

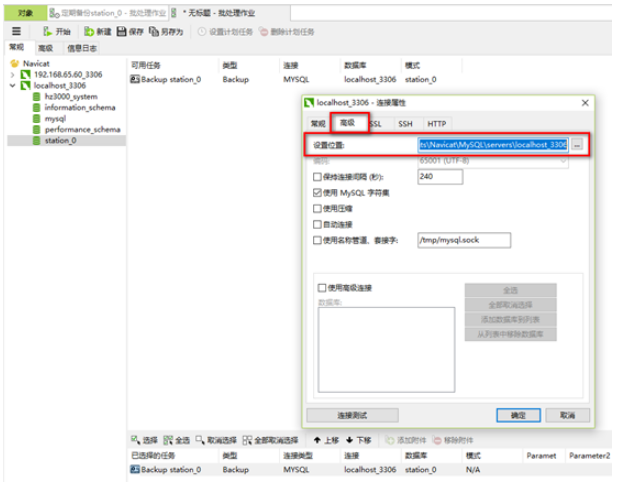
然后在控制面板上，创建任务。任务指定该批处理文件的执行时间。但是执行一次该文件产生的备份都会覆盖上一次的文件。











## 3、Navicat从Excel 导入

① 打开Navicat premium>>表>>导入向导，导入运行时秘钥sheet页数据，注意选择数据行开始位置；

② 导入过程中输入临时表名称：tmp\_\*\*；

测试导出用户数据至Excel：

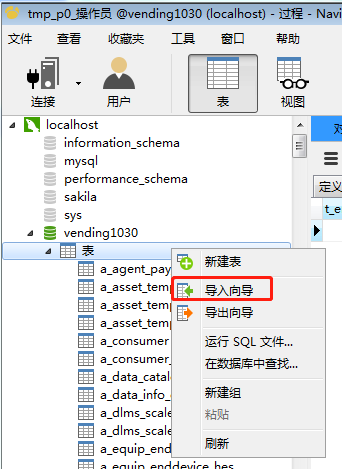
269166条数据，耗时1505s约等于25min，16.8M。

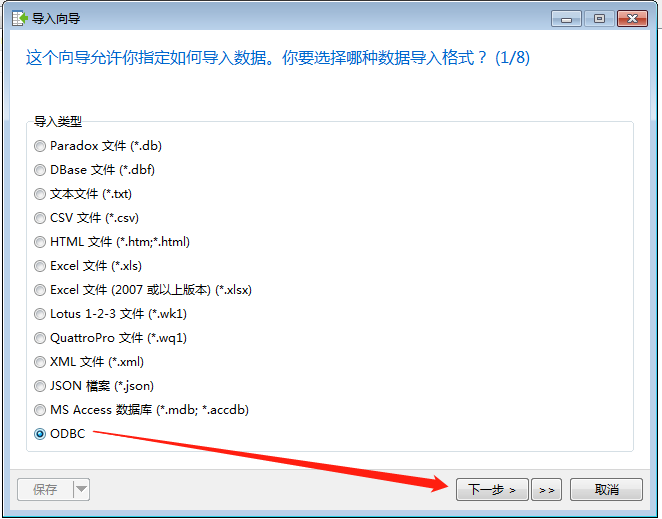
弃用，选择从db直接导入。

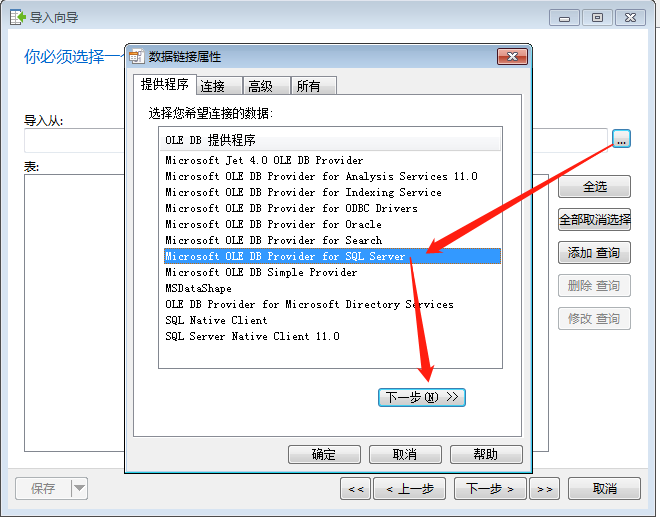
注意：在Navicat中导入2007版本以上Excel的时候可能会提示“无法打开文件”，需要先双击文件打开，再执行导入动作，可能是Navicat的bug吧。

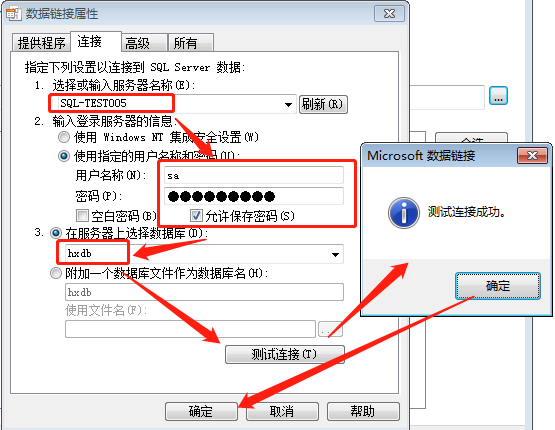
## 4、Navicat从ODBC导入

过程参照下图

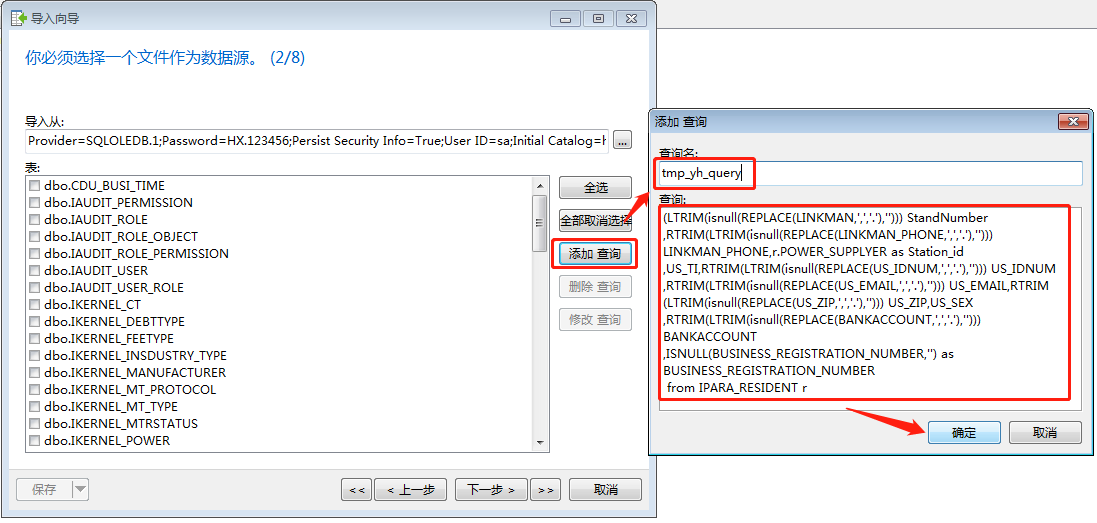


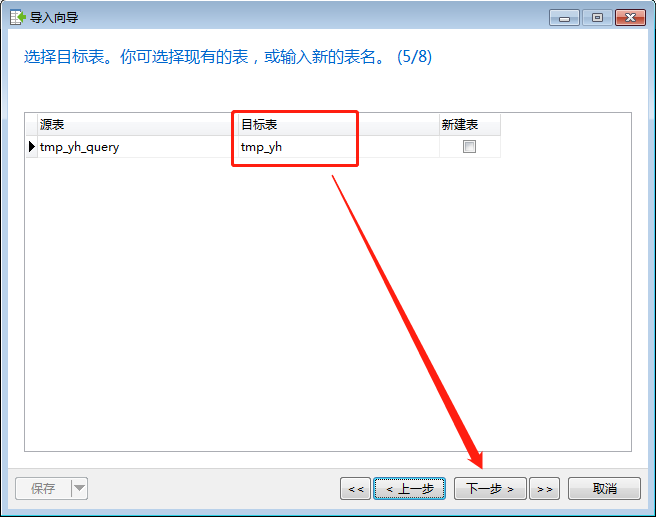


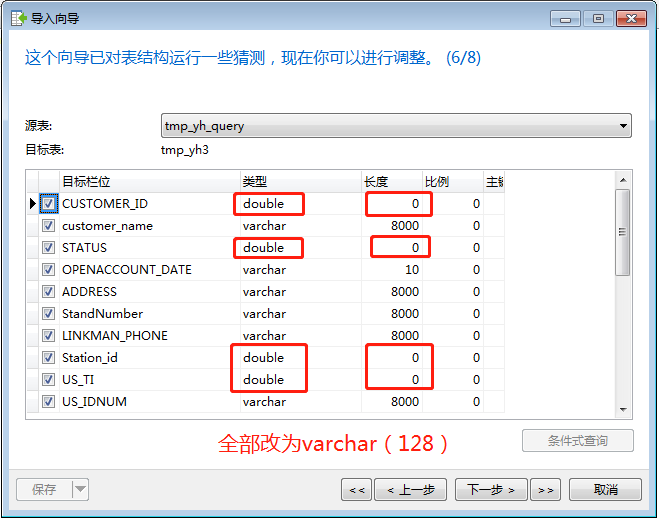


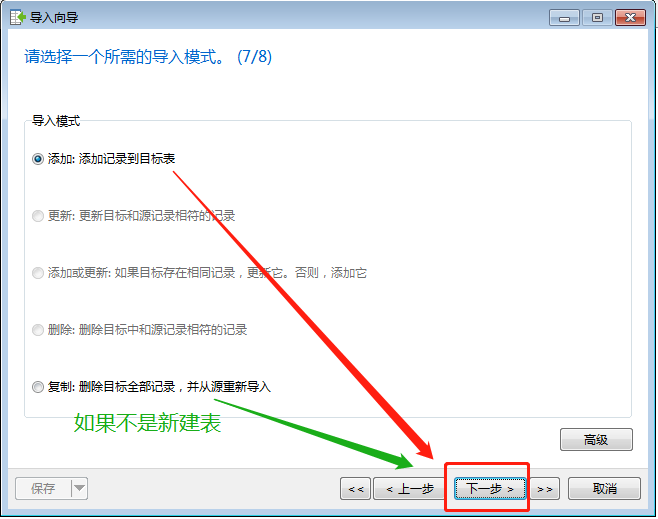


注意：一定要勾选[允许保存密码]否则报错 无法打开文件Provider=SQLOLEDB...









最后点击开始，若提示：disk内存不足，则清理下内存/重启下机器

## 5、Navicat创建存储过程

打开Navicat premium>>函数>>新建存储过程

# 三、数据迁移

由郭工提供导出SQL，要求导出结果形式严格符合《需求范围.xlsx》。

uap添加中文版本：tomcat/uapconf/uiconfig.json中的languages添加{“code”: “zh\_CN”, “text”: “中文”}

## 手动：配置角色

**数据源**：《需求范围.xlsx》角色.sheet。

**操作过程**：

① MDCAdmin登录vending8.0，系统设置>>权限管理>>角色管理，

MDC Role开启所有菜单权限

手动新增Centlec角色。14个

## 脚本0：运行时秘钥（略）

**数据源**：《需求范围.xlsx》.运行时秘钥sheet。

**操作过程**：

① 采用2.3导入临时表tmp\_vk并验证（起止行3,4,23）；

② 存储过程参照附件：脚本0：运行时秘钥.sql；

③ 执行存储过程；参数为’加密盒编号’,’加密服务地址’，例如：’456165156’,’http://172.30.12.125:8086/RncrptionService/rest/sts’；

④ 验证表：D\_EBOX、VD\_P\_EBOX\_RUN、P\_THIRD\_PARTY、VD\_P\_VK、VD\_P\_VK\_RUN。

注意：重跑此脚本需要重跑后续脚本.

## 脚本1：单位（依赖运行时秘钥）

**数据源**：《需求范围.xlsx》.单位sheet。

**操作过程**：

① 采用2.3导入临时表tmp\_dw并验证（起止行9,10,93）；

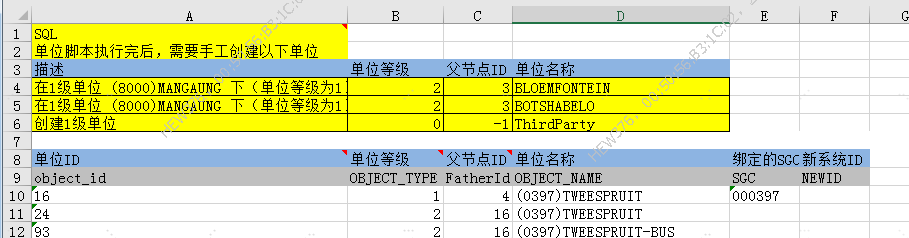
② 存储过程参照附件：脚本1：单位.sql

③ 执行存储过程；

④ 验证表：UAP\_ORGANIZATION、VD\_P\_ORG\_VK\_REL、TMP\_DW\_P

UAP\_ORGANIZATION中父单位、No不能为空

⑤ 需要手动添加三个单位，严格意义上不算单位，没有主键，不是独立CDU，没有绑定SGC，只是用来挂代理商



注意：重跑此脚本需要重跑后续脚本.

## 脚本2：电网结构/站线变：TMP\_ZXB

**数据源**：《需求范围.xlsx》.站线变sheet。

**操作过程**：

① 采用2.3导入临时表tmp\_zxb并验证（起止行3,4,616）；

② 存储过程参照附件：脚本2：站线变.sql

③ 执行存储过程；

④ 验证表：A\_GRID\_LINE\_TF、A\_GRID\_SUBS\_LINE\_RELA、A\_GRID\_SUBS、A\_GRID\_LINE、A\_GRID\_TRANSFORMER

注意：重跑此脚本需要重跑后续脚本.

## 手动：费率

**数据源**：《南非对应8.0角色.xlsx》.费率sheet。

**操作过程**：

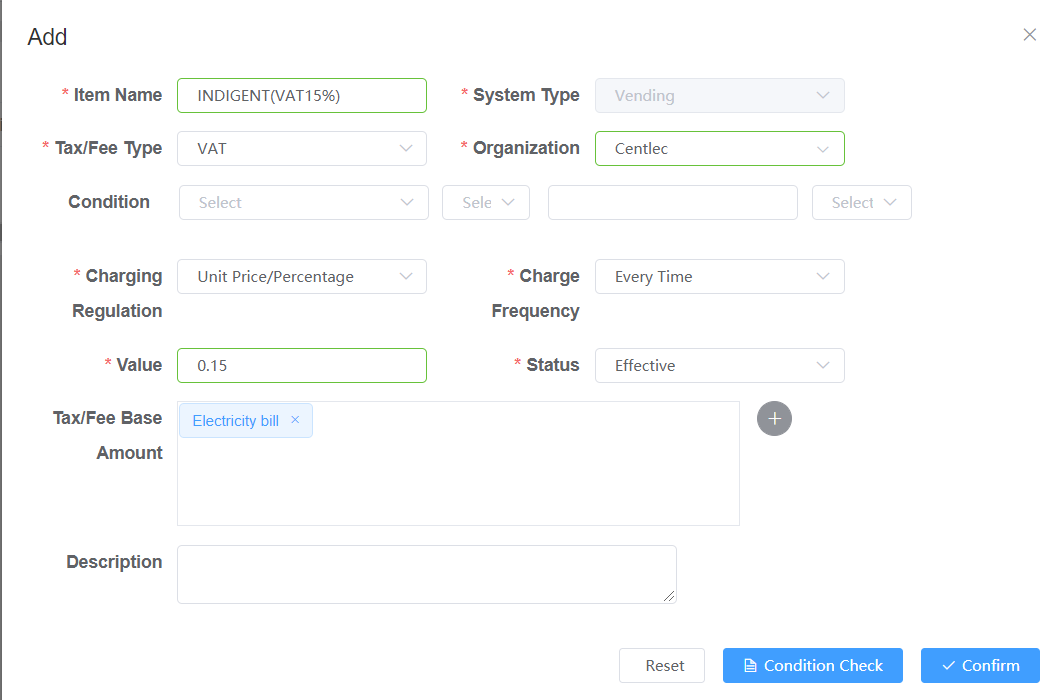
① 手动创建：计费项

INDIGENT(VAT15%)

COMMON(VAT15%)

BUSINESS(VAT15%)

MANGAUNGNON PROFIT(VAT15%)





② 手动创建：费率方案、版本、审核，并添加对应的VAT，版本生效时间一个为当前，一个为2020.6.1。

|  |  |  |
| --- | --- | --- |
| 序号 | 方案名 | 版本名规则 |
| 1 | MANGAUNG-TG1(FBE) | 方案名-20190901  方案名-20200601 |
| ... | ... | ... |
| 17 | NALEDI-NON PROFIT | 方案名-20190601 |

③ 将所有费率版本ti更新为1（前提是确认老系统的费率ti都是1）

Select ti from TARIFF\_INDEX group by ti;

Select ti from TARIFF\_TABLE group by ti;

Update vd\_e\_bill\_pkg\_ver set ti = 1;

④ 注意配置定时任务，如果着急版本生效

Update vd\_e\_bill\_pkg\_ver set STATUS = ‘05’ where STATUS = ‘03’ and VER\_NO = ‘.................’;

注意：

0、所有费率方案均可挂在根单位下，

暂时统一选居民=低压居民，商业=非居民

1、版本-单次充值上限500，不做限制，代理商那边有限制

2、对于2个阶梯单价相同的，无需合并成一个阶梯

3、不存在特殊阶梯

## 脚本3、表计、批次

*由于用户状态依赖此临时表，遂先导入该表。*

select ACTUAL\_CUSTOMER\_ID,COUNT(1) from IPARA\_MTRPOINT m group by m.ACTUAL\_CUSTOMER\_ID having COUNT(1)>1

**数据源**：ODBC===《需求范围.xlsx》.表计sheet。

**操作过程**：

空表状态下，先检查A\_EQUIP\_METER\_VENDING表字符集为utf8mb4

① ~~手工在系统创建一个批次，按表号1-99999999999，量控-南非STS1，绑定VK999910，ti为1，没有预置额度。~~

② 做以下检查，然后采用2.4导入临时表tmp\_bj并验证；

查看老库有没有重复的表号（新库不允许重复表号存在）

select MT\_COMM\_ADDR from IPARA\_MTRPOINT group by MT\_COMM\_ADDR having count(MT\_COMM\_ADDR)>1

确定无需转移的表号ID并写入查询SQL

③ 存储过程参照附件：脚本3：表计.sql

④ 执行存储过程；

⑤ 验证表：A\_EQUIP\_METER、A\_EQUIP\_METER\_VENDING、A\_MP\_EQUIPMENT\_RELA（空）

SQL验证A\_EQUIP\_METER.meter\_model、mgt\_status都插入了code

SQL验证A\_EQUIP\_METER\_VENDING为每个表都插入了VK

Select \* from tmp\_bj b where not exists

(select 1 from vd\_p\_vk c where b.mus\_sgcid = c.sgc and b.mus\_keyvision = c.krn and b.mus\_keyexpiry = c.ken and c.ms = ‘02’)

⑥ 验证每种表型都插入了相应批次：35个批次；



将来 STS2协议的，按表号为每个厂商建立一个批次

如：海兴，表号范围14+最小STS2协议表号 至 14+999999999（11位表号）



注意：重跑此脚本需要重跑后续脚本.

遗留问题：

老系统MeterStatus字段可以控制表计可否售电，新系统无匹配的功能字段。

批次问题，可能需要重新整理脚本导入，根据实际出厂VK。

## 脚本4：用户/计量点

**数据源**：ODBC===《需求范围.xlsx》.用户sheet。

**操作过程**：

① 做以下检查，采用2.4导入临时表tmp\_yh并验证；

查看有没有一户多表的情况

select ACTUAL\_CUSTOMER\_ID,COUNT(1) from IPARA\_MTRPOINT m group by m.ACTUAL\_CUSTOMER\_ID having COUNT(1)>1;

② 存储过程参照附件：脚本4：用户.sql

③ 执行存储过程；

④ 验证表：A\_CONSUMER、A\_USAGEPOINT、A\_MP\_EQUIPMENT\_RELA，VD\_BANK\_ACC、VD\_C\_CERT、VD\_C\_OBJ\_CERT\_REL、VD\_C\_CONTACT、VD\_C\_OBJECT\_CONTACT\_REL、VD\_C\_CONTACT\_INFO，VD\_A\_USER\_DEBT（空）

## 手动：配置免费额度

**数据源**：《需求范围.xlsx》.表计sheet.费率为FBE。

**操作过程**：

① ~~存储过程参照附件：脚本5.1：免费额度配置~~

② 验证表：VD\_OFFER\_SET、VD\_OFFER\_SET\_DETAIL

页面验证已添加如下配置

不用为33710个单个用户插入免费额度，通过页面配置，通过用电类别201-乡村居民用电统一配置，仅一条。



## 脚本5：累计值（依赖用户、表计）

**数据源**：ODBC===《需求范围.xlsx》.累计值sheet。

**操作过程**：

空表状态下，先检查 累计值表起止时间 从datetime改为DATE

① 采用2.4导入临时表tmp\_ljz并验证；

② 存储过程参照附件：脚本5.2：累计值.sql

③ 执行存储过程；

④ 验证表：VD\_C\_CUMU\_VALUE（注意表计对应的免费额度是否存入累计值）

## 脚本6：债务（依赖用户、表计）

**数据源**：ODBC===《需求范围.xlsx》.债务sheet。

**操作过程**：

① 采用2.4导入临时表tmp\_zw并验证；（注意CREATE\_DATE要用datetime去接收timestamping）

② 存储过程参照附件：脚本6：债务.sql

③ 执行存储过程；

④ 验证表：VD\_A\_USER\_DEBT、VD\_A\_USER\_DEBT\_SET

## 脚本7：代理商（依赖单位）

**数据源**：《需求范围.xlsx》.代理商sheet。

**操作过程**：

① 采用2.4导入临时表tmp\_dls并验证；

② 存储过程参照附件：脚本7：代理商.sql

③ 执行存储过程；

④ 验证表：

VD\_AGT\_AGENT、VD\_A\_ACCOUNT、VD\_AGT\_CHARGE\_LIMT、

VD\_AGT\_AGENT\_OPERATOR

现场需要打开“代理商员工”菜单

Update uap\_menu\_display set state = 1 where name like ‘%代理商员工%’

遗留问题：

1、需要补充添加3个单位，纯粹用于挂代理商（参照单位表），

但是目前一个单位只能挂一个自营代理商，

所以准备修改设计，使一个单位可挂多个自营代理商。

2、一个代理商需要绑定多个操作员，可能需要再次遍历 tmp\_czy 表，查看操作员所属代理商...

3、老系统代理商单个迁移时需要知道老系统的【信用额度】，【当前余额】

① VD\_A\_ACCOUNT 中的 “可用余额”=“账户余额”初始为0，

在页面充值【当前余额】-【信用额度】（负数时走页面调账）

② VD\_AGT\_AGENT 中的 “最小余额限制”，

在页面设置为 -【信用额度】

需要整理到《迁移交接手册》

## 脚本8：操作员（依赖单位、代理商）

**数据源**：《需求范围.xlsx》操作员.sheet。

**操作过程**：

① ~~采用2.3导入临时表tmp\_czy并验证，~~**~~设置起止时间的字段类型为DATE~~**~~，否则导入时间错误；~~

采用2.4导入临时表tmp\_czy并验证；

验证user\_account唯一性：

Select user\_account from tmp\_czy group by user\_account having count(1) > 1 where roleName is not null;

② 存储过程参照附件：脚本8：操作员.sql

③ 执行存储过程（内部调用修改Email字段长度的存储过程，统一设置所有操作员的密码是CENTLEC123456）

④ 验证表：UAP\_USER、UAP\_USER\_ROLE

⑤ 执行pwd程序，注意配置数据库路径，运行结束重启uap，所有密码更新为CENTLEC123456。

⑥ （脚本里已经写了）uap\_user状态改为3，提示修改初始密码；

设置租户的密码过期时间：（否则新登录新密码提示过期）

Admin登录>>Application Settings>>Multi-Tenant>>过期时间建议90天

遗留问题：

登录主站，页面报uap500错误，debug级别重启uap后错误消失，原因不明。

## 脚本9：还债记录（依赖操作员、单位）

**数据源**：ODBC===《需求范围.xlsx》.还债记录sheet。

**操作过程**：

① 采用2.4导入临时表tmp\_zw并验证；

② 应收表、收费明细表、实收表需要增加两列临时字段

CALL PR\_MOD\_COL(‘VD\_A\_RCVBL\_FLOW’,’ADD’,’DEBTID’,’VARCHAR(64)’,’NULL’,’NULL’,’’);

CALL PR\_MOD\_COL(‘VD\_A\_RCVBL\_FLOW’,’ADD’,’ORDERID’,’VARCHAR(128)’,’NULL’,’NULL’,’’);

CALL PR\_MOD\_COL(‘VD\_A\_PAY\_FLOW’,’ADD’,’DEBTID’,’VARCHAR(64)’,’NULL’,’NULL’,’’);

CALL PR\_MOD\_COL(‘VD\_A\_PAY\_FLOW’,’ADD’,’ORDERID’,’VARCHAR(128)’,’NULL’,’NULL’,’’);

CALL PR\_MOD\_COL(‘VD\_A\_RCVED\_FLOW’,’ADD’,’DEBTID’,’VARCHAR(64)’,’NULL’,’NULL’,’’);

CALL PR\_MOD\_COL(‘VD\_A\_RCVED\_FLOW’,’ADD’,’ORDERID’,’VARCHAR(128)’,’NULL’,’NULL’,’’);

③ 存储过程参照附件：脚本9：还债记录.sql

## 配置：单位权限

MDCAdmin登录主站>>组织体系管理>>管理单位配置>>点根单位，选择MDCAdmin>>添加单位访问权限 （UAP\_ USER\_ORG \_MANAGE）>>添加全部单位；

## 配置：小票样式

① 清空 vd\_pre\_model；

② 手动跑 V2\_0\_104\_\_VD\_PRE\_MODEL\_DATA.sql 脚本；

如果工具不识别脚本中的特殊字符，可以尝试自动执行：

① 清空 vd\_pre\_model；

② 删除flyway脚本执行记录；

DELETE FROM BIZ\_SYS\_SCHEMA\_VERSION WHERE SCRIPT = ‘V2\_0\_104\_\_ VD\_PRE\_MODEL\_DATA.sql’;

③ 重启Tomcat会自动执行；

# 四、常见的问题

秘钥绑定》》编辑》》单位不显示

---需要先为相关操作员赋予相关单位访问权限

# 五、待解决的问题

秘钥绑定》》HexingText绑了工厂秘钥，换绑时不显示（略）

费率版本》》生效时间》》0点跳动问题

费率版本》》同电价阶梯，验证算费结果

~~操作员-角色关系少了7条记录~~

Cashier--销售管理--收费查询--单个单位--没有权限

待定：280个操作员统一赋予所有单位的权限（脚本）

确认mdc-config中数据库连接时区配置

# 六、数据融合

## 1、tmp→tmp1

① 选择夜深人静、无人购电无人维护档案的时间

② 执行脚本0.2将tmp\_bj, tmp\_yh, tmp\_zw后缀全部+1。

## 2、快速导入最新数据

② tmp\_bj, tmp\_yh, tmp\_zw的取数据脚本

采用2.4新建4个查询导入；

注意tmp\_zw里面的CREATE\_DATE使用Datetime接收；

注意所有tmp表都是varchar长度128，都指定主键！！！

③ 新库验证下ti有没有变（默认没有变化）

select \* from tmp\_yh t,tmp\_yh1 t1

where t.customer\_id = t1.customer\_id

and t.us\_ti != t1.us\_ti

④ （月底24点？）

select max(lastvenddate) from tmp\_bj1;

获取上次导入的最后售电时间，作为**tmp\_ljz**的参数；

指定联合主键

**tmp\_ljz**的取数据脚本，累计值记录一直在增加……

## 3、<脚本1：运行时秘钥.txt>；

/\*1-更新计量点关联费率方案\*/ UPDATE a\_usagepoint

/\*2-更新用户电压等级\*/ UPDATE a\_consumer

/\*3-更新计量点管理单位\*/ UPDATE a\_usagepoint

/\*4-更新表计管理单位\*/ UPDATE a\_equip\_meter

/\*5-更新计量点关联战线变\*/ UPDATE a\_usagepoint

/\*6-更新用户管理单位\*/ update a\_consumer

① 先复制这三张表

② BCompare对比或者sql验证：

SELECT yh.\*,yh1.tariffname AS tariffnameold FROM tmp\_yh yh INNER JOIN tmp\_yh1 yh1 ON yh.customer\_id = yh1.customer\_id AND yh.tariffname!=yh1.tariffname;

select \* from vd\_e\_bill\_package where PKG\_NAME = 'MOHOKARE-TG4'; -- 139896 -- 567337 -- 8 -- 10

select \* from vd\_e\_bill\_package where PKG\_NAME = 'KOPANONG-TG5';

select \* from vd\_e\_bill\_package where PKG\_NAME = 'MANGAUNG-TG1'; -- 238101 -- 673519 -- 2 -- 1

select \* from vd\_e\_bill\_package where PKG\_NAME = 'MANGAUNG-TG1(FBE)';

select \* from vd\_e\_bill\_package where PKG\_NAME = 'MANGAUNG-TG1'; -- 268076 -- 706598 -- 2 -- 1

select \* from vd\_e\_bill\_package where PKG\_NAME = 'MANGAUNG-TG1(FBE)';

select \* from vd\_e\_bill\_package where PKG\_NAME = 'MANTSOPA-TG2'; -- 282893 -- 721606 -- 4 -- 12

select \* from vd\_e\_bill\_package where PKG\_NAME = 'MANTSOPA-TG2(BUS)';

select \* from a\_consumer where CONS\_NO = 'yh\_282893';

select \* from tmp\_bj where customer\_id = '268076';

select \* from tmp\_bj1 where customer\_id = '268076';

select \* from a\_equip\_meter where assetno = '13500735678'; -- 482646

select \* from a\_usagepoint where MP\_NO like '%13500735678'

SELECT yh.\*,yh1.station\_id AS station\_idold FROM tmp\_yh yh INNER JOIN tmp\_yh1 yh1 ON yh.customer\_id = yh1.customer\_id AND yh1.station\_id!=yh.station\_id;

select \* from a\_consumer where CONS\_NO in

('yh\_139896', -- 567337 -- 002004001002 -- 33 -- 54 -- 002006008002

'yh\_238101', -- 673519 -- 002002001002 -- 11 -- 10 -- 002002001001

'yh\_268076', -- 706598 -- 002002001002 -- 11 -- 10 -- 002002001001

'yh\_280627', -- 719697 -- 002003003002 -- 22 -- 24 -- 002003001002

'yh\_282893'); -- 721606 -- 002003003002 -- 22 -- 92 -- 002003003003

select \* from uap\_organization where code = 'dw\_92';

select \* from tmp\_bj where customer\_id in ('139896','238101','268076','280627','282893') limit 10;

select \* from a\_equip\_meter where assetno in

('07040886538', -- 002004001002

'13500735678', -- 002002001002

'13500857092', -- 002002001002

'45033314035', -- 002003003002

'81130292972') -- 002003003002

## 4、<脚本2：表计秘钥.txt>

0-更新表计vk\_id UPDATE a\_equip\_meter\_vending

1-更新表计ti UPDATE a\_equip\_meter\_vending

① 先复制这2张表

② ~~BCompare对比~~或者sql验证：

SELECT bj.\*,bj1.mus\_sgcid as mus\_sgcidold,bj1.mus\_keyvision as mus\_keyvisionold,bj1.mus\_keyexpiry as mus\_keyexpiryold

FROM tmp\_bj bj INNER JOIN tmp\_bj1 bj1 ON bj.mt\_comm\_addr=bj1.mt\_comm\_addr AND (bj1.mus\_sgcid!=bj.mus\_sgcid or bj1.mus\_keyvision!= bj.mus\_keyvision

or bj1.mus\_keyexpiry!= bj.mus\_keyexpiry);

select \* from a\_equip\_meter\_vending where meter\_id in

(

select METER\_ID from a\_equip\_meter where assetno in

('07040886538', -- 8003 -- 12 -- 8009 -- 18

'07049014116', -- 8000 -- 9 -- 8006 -- 15

'45033314035') -- 8024 -- 26 -- 0397 -- 8

);

select \* from vd\_p\_vk;

SELECT bj.\*,bj1.mus\_ti AS mus\_tiold

FROM tmp\_bj bj INNER JOIN tmp\_bj1 bj1 ON bj.mt\_comm\_addr=bj1.mt\_comm\_addr AND bj1.mus\_ti!=bj.mus\_ti;

select \* from a\_equip\_meter\_vending where ti <> 1

## 5、<脚本3.1：用户.txt>

/\*1-更新用户状态\*/ /\*01-新开户;02-已投运；03-已销户\*/

UPDATE a\_consumer

# 2-插入用户档案 INSERT INTO a\_consumer

① 先复制这1张表

② BCompare对比或者sql验证：

SELECT yh.customer\_id,yh.status,yh1.status AS statusold,

(SELECT MT\_COMM\_ADDR FROM tmp\_bj bj WHERE bj.customer\_id=yh.customer\_id) MT\_COMM\_ADDR,

(SELECT MT\_COMM\_ADDR FROM tmp\_bj bj1 WHERE bj1.customer\_id=yh1.customer\_id) MT\_COMM\_ADDRold

FROM tmp\_yh yh INNER JOIN tmp\_yh1 yh1 ON yh.customer\_id = yh1.customer\_id AND yh1.status!=yh.status;

select COUNT(1) from tmp\_yh;

select COUNT(1)

FROM tmp\_yh a

LEFT JOIN uap\_organization c ON CONCAT('dw\_',a.station\_id) = c.CODE

WHERE NOT EXISTS(SELECT cons.CONS\_NO FROM a\_consumer cons WHERE CONCAT('yh\_',a.customer\_id) = cons.CONS\_NO);

select COUNT(1) from a\_consumer;

## 6、<脚本3.2：用户换表.txt>

/\*1-更新表计旧系统状态\*/

/\*1-更新表计状态，如果表绑定到了新户，则status为运行02，否则拆回04\*/ UPDATE a\_equip\_meter

#2-更新计量点设备关联状态\*/ DELETE FROM a\_mp\_equipment\_rela

/\*3-删除计量点\*/ DELETE FROM a\_usagepoint

① 先复制这3张表

② BCompare对比或者sql验证：

SELECT bj.mt\_comm\_addr,bj.meterstatus,bj.customer\_id,bj1.customer\_id AS customer\_idold FROM tmp\_bj bj INNER JOIN tmp\_bj1 bj1 ON bj1.mt\_comm\_addr = bj.mt\_comm\_addr

AND bj1.customer\_id!=bj.customer\_id;

③ 暂无此类数据（此2表记录数不会变化）

select COUNT(1) from a\_mp\_equipment\_rela; -- 261482

select COUNT(1) from a\_usagepoint; -- 261482

注意：脚本只是去除老的关系，新的换表绑定关系在5.2中重新插入

## 6.1、临时添加<脚本3.3：用户拆表.txt>

SELECT bj1.mt\_comm\_addr,bj1.meterstatus,bj1.customer\_id,bj1.customer\_id AS customer\_idold

FROM tmp\_bj1 bj1

WHERE NOT EXISTS(SELECT bj.mt\_comm\_addr FROM tmp\_bj bj WHERE bj.mt\_comm\_addr = bj1.mt\_comm\_addr);

发现用户最近有在老库删除表（有可能是用了乔磊工的工具，有可能是因为这些表不符合Luhn规则，出现删除表档案的情况），我们在新库保留表档案，状态改为04删除，并删除关联关系。

① 先复制这3张表

② BCompare对比或者sql验证：

select \* FROM a\_equip\_meter meter

WHERE NOT EXISTS(SELECT b.mt\_comm\_addr FROM tmp\_bj b WHERE meter.assetno = b.mt\_comm\_addr);

select \* from tmp\_bj where mt\_comm\_addr = '14326600717';

select \* from tmp\_bj1 where mt\_comm\_addr = '14326600717';

select \* from tmp\_yh where customer\_id = '284160';

select \* from tmp\_yh1 where customer\_id = '284160';

select \* from a\_mp\_equipment\_rela where EQUIPMENTID = '507581';

select \* from a\_usagepoint where MP\_ID = '1302718';

## 7、<脚本4：表计.txt>

# 1-插入表计档案 INSERT INTO a\_equip\_meter

# 2-插入表计预付费表 INSERT INTO a\_equip\_meter\_vending

① 先复制这2张表

② BCompare对比或者sql验证：

select COUNT(1) from a\_equip\_meter; -- 261488 -- 261708 == 220

select \*

FROM tmp\_bj b

WHERE NOT EXISTS(SELECT meter.assetno FROM a\_equip\_meter meter WHERE b.mt\_comm\_addr = meter.assetno);

select COUNT(1) from a\_equip\_meter\_vending; -- 261486 -- 261706 == 220

select \*

FROM

A\_EQUIP\_METER a

INNER JOIN tmp\_bj b ON a.assetno = b.mt\_comm\_addr

INNER JOIN vd\_p\_vk c ON b.mus\_sgcid = c.sgc

AND b.mus\_keyvision = c.krn

AND b.mus\_keyexpiry = c.ken

AND c.ms = '02'

AND c.base\_time=1993

WHERE a.meter\_mode='02' AND NOT EXISTS(SELECT meterVk.METER\_ID FROM a\_equip\_meter\_vending meterVk WHERE meterVk.METER\_ID=a.METER\_ID);

## 8、<脚本5.1：计量点.txt>

# 1-插入计量点档案 INSERT INTO a\_usagepoint

① 先复制这1张表

② BCompare对比或者sql验证：

select \*

FROM a\_consumer a

INNER JOIN tmp\_bj b ON a.cons\_no = CONCAT('yh\_', b.customer\_id)

LEFT JOIN a\_grid\_line c ON CONCAT('zxb\_',b.line\_id) = c.line\_no

LEFT JOIN a\_grid\_transformer d ON CONCAT('zxb\_',b.suburb\_id) = d.tf\_no

LEFT JOIN VD\_E\_BILL\_Package e ON b.tariffname = e.pkg\_name

WHERE NOT EXISTS(SELECT f.mp\_no FROM a\_usagepoint f WHERE f.mp\_no = CONCAT('mp\_',a.cons\_id,'\_',b.mt\_comm\_addr));

select COUNT(1) from a\_usagepoint; -- 261481 -- 261701 == 220

## 9、<脚本5.2：计量点设备与用户联系人.txt>

# 1-插入计量点-设备关系 INSERT INTO a\_mp\_equipment\_rela

# 2-保存用户联系方式 INSERT INTO a\_consumer\_contacts

① 先复制这2张表

② BCompare对比或者sql验证：

select \*

FROM a\_usagepoint a

INNER JOIN a\_equip\_meter b ON SUBSTRING\_INDEX(a.mp\_no,'\_',-1) = b.assetno AND b.meter\_mode='02'

WHERE NOT EXISTS(SELECT rela.MP\_ID FROM a\_mp\_equipment\_rela rela WHERE rela.MP\_ID = a.mp\_id

AND rela.equipmentid = b.meter\_id AND rela.equipmenttype='02');

SELECT

\*

FROM a\_consumer a

INNER JOIN tmp\_yh b ON (b.LINKMAN\_PHONE <> '' OR b.US\_EMAIL <> '') AND a.cons\_no = CONCAT('yh\_', b.CUSTOMER\_ID)

WHERE NOT EXISTS(SELECT c.CUSTOMER\_ID FROM tmp\_yh1 c WHERE c.CUSTOMER\_ID = b.CUSTOMER\_ID);

select COUNT(1) from a\_mp\_equipment\_rela; -- 261481 -- 261701 == 220

select COUNT(1) from a\_consumer\_contacts; -- 14510 -- 14529 == 19

## 10、<脚本6：债务.txt>

# 1-更新用户债务update VD\_A\_ACCOUNT、update vd\_a\_user\_debt

# 2-插入用户债务表 INSERT INTO vd\_a\_user\_debt

# 3-插入用户债务配置表 INSERT INTO vd\_a\_user\_debt\_set

## 11、<脚本7：累计值-数据融合.txt>

#查询新系统对应收费日期周期用户免费值

#查询用户最大免费累计日期

#免费值放入下月用户免费累计值

#购电时间周期内没有用户免费额度数据，则增加

select CUMU\_OBJ\_ID, COUNT(1) from vd\_c\_cumu\_value where CUMU\_OBJ = '01' GROUP BY CUMU\_OBJ\_ID HAVING COUNT(1) > 1; -- 30458

select \* from tmp\_ljz where isfree = 1; -- 534

SELECT cons.cons\_id as consId,ljz.mt\_comm\_addr as meterNo, ljz.isfree as isFree, ljz.energy as freeValue, ljz.lastvenddate as vendDate

from tmp\_ljz ljz

inner join a\_equip\_meter meter on meter.assetno=ljz.mt\_comm\_addr

inner join a\_mp\_equipment\_rela rela on meter.METER\_ID = rela.EQUIPMENTID AND rela.EQUIPMENTTYPE='02'

inner join a\_usagepoint point on rela.mp\_id=point.mp\_id

inner join a\_consumer cons on cons.cons\_id=point.cons\_id

where ljz.isfree=1; -- 534

-- sexecute process

select \* from vd\_c\_cumu\_value where CUMU\_OBJ\_ID='781456'; -- 30458 + 534 = 30992

select COUNT(1) from vd\_c\_cumu\_value; -- 30458 + 534 = 30992

select \*

from vd\_c\_cumu\_value cumu

where cumu.cumu\_obj='01' and cumu.CUMU\_ITEM='0101'

and cumu.CUMU\_OBJ\_ID=

(

select cons.cons\_id

from a\_equip\_meter meter

inner join a\_mp\_equipment\_rela rela on meter.METER\_ID = rela.EQUIPMENTID AND rela.EQUIPMENTTYPE='02'

inner join a\_usagepoint point on rela.mp\_id=point.mp\_id

inner join a\_consumer cons on cons.cons\_id=point.cons\_id

where meter.assetno = '06410260001'

);

## 12、临时添加<脚本8：step迁移>

月底半夜过了零点

① 先导入tmp\_stepcumu

② 执行脚本8

# 七、迁移验证（按迁移关联度排序）

## 1、操作员登录验证

**Uap管理员账号：**admin/admin2019.

**MDC管理员账号：**MDCAdmin/MDCAdmin2019-

## 2、迁移数据页面验证

**预置条件**：为登录操作员授予单位访问权限，以MDCAdmin为例；

**预置操作**： MDCAdmin登录主站>>组织体系管理>>管理单位配置>>点根单位，选择用户>>添加单位访问权限 （UAP\_ USER\_ORG \_MANAGE）>>添加全部单位；

**验证操作**：

组织体系管理>>组织管理>>组织单位，抽查单位档案、从属关系等。

档案>>电网管理>>站线变，抽查站线变档案、从属关系等。

档案>>用户管理>>用户档案管理，抽查用户档案、状态等。

档案>>用户管理>>计量点档案管理，抽查计量点档案、状态等。

设备>>设备信息维护>>电表管理，抽查电表档案、状态等。

核算管理>>债务管理，抽查用户债务、明细等。

## 3、营业厅售电场景验证

**预置条件**：表计有批次，计量点费率方案下有生效版本；

**预置操作**：无；

**验证操作**：

①销售管理>>售电>>选择用户/表计。

②数据库中查询并记录该表计累计值。

③（杭州盒子测试需要修改VK对应的SGC号为测试盒子内的SGC）

④随后进行售电操作（充值方式分别为100金额，40资源测试），生成Token核对小票。

⑤确保基于累计值、费率方案算费正确。

## 4、开户场景验证

**预置条件**：新开户。

**预置操作**：正常录入表计、用户档案并关联，准备好Token模板。

**验证操作**：

①token管理>>token申请，业务类型：开户装表。

②token管理>>token审核，通过。

③token管理>>token执行，检查生成Token的正确性，充值Token/附加费用应产生相应应收。

## 5、正常收费场景验证

**预置条件**：对有债务的用户进行收费；

**预置操作**：

**验证操作**：

①销售管理>>收费>>选择用户/表计。

②对所欠债务进行部分收取。

③确保收费成功，计算结果正确。

## 6、收费撤销场景验证（略）

**预置条件**：对有已收费的用户进行收费撤销；

**预置操作**：

**验证操作**：

①

## 7、充值纠正场景验证（略）

**预置条件**：对有已充值的用户进行充值纠正；

**预置操作**：

**验证操作**：

①

## 8、换表场景验证（略）

**预置条件**：对有已装表的用户进行换表；

**预置操作**：

**验证操作**：

①

## 9、注销场景验证（略）

**预置条件**：对有已装表的用户进行注销；

**预置操作**：

**验证操作**：

①

## 10、注销结算场景验证（略）

**预置条件**：

**预置操作**：

**验证操作**：

①

## 11、日结场景验证（略）

**预置条件**：

**预置操作**：

**验证操作**：

①

## 12、解款场景验证（略）

**预置条件**：

**预置操作**：

**验证操作**：

①