# 业务逻辑设计

# 文件预处理

1. 检查合并记录长度是否正确.

E0: 记录长度错误

2.检查各个字段是否正确，错误结果:

E1: bankno错误

E2: IDNO错误

E4: Balance错误

E5: IDTYPE错误

E6: GENDER错误，性别错误

E7: Stime错误

1. 将身份证中小写字母转化为大写

预处理结果

账户信息处理失败应答文件ACCT-$$$$$$$$-MERGEREQ-YYYYMMDD-ERR（中行）

账户关联关系处理失败应答文ACCT-$$$$$$$$-MERGERELAREQ-YYYYMMDD-ERR（中行）

程序伪代码：

If(打开账户读取文件fp\_in\_a == erro|| 打开关联关系读取文件fp\_in\_r== erro)

Return FAIL

If(打开账户正确输出文件fp\_out\_a ==erro || 打开关联关系正确输出文件fp\_out\_r==erro)

Return FAIL

if(打开账户错误输出文件fp\_err\_a==erro || 打开关联关系错误输出文件fp\_err\_r==erro)

return FAIL;

if(打开账户日志文件fp\_log\_a== erro || 打开关联关系日志文件fp\_log\_r== erro)

return FAIL;

while( ret=读取fp\_in\_a != EOF)

{

if(ret == erro)

记录账户错误文件、记录日志文件

comtinue

if( idtypy !=’01’) errcode=’E5’ 记录错误文件fp\_err\_a, continue;

if( idno=NULL || idno.lenth !=15 || idno.lenth !=18 || ...)

errcode=’E2’ 记录错误文件fp\_err\_a, continue;

if(Gender!=’F’ || Gender!=’M’ ) errcode=’E6’ 记录错误文件fp\_err\_a, continue;

if(bankno!=bankid ) errcode=’E1’ 记录错误文件fp\_err\_a, continue;

if(Balance 含有小数点 ) errcode=’E4’ 记录错误文件fp\_err\_a, continue;

if(Stime含有小数点|| Stime<2000 )errcode=’E7’ 记录错误文件fp\_err\_a, continue;

记录该条记录在正确数据文件中

}

while( ret=读取fp\_in\_r != EOF)

{

if(ret == erro)

记录账户错误文件、记录日志文件

comtinue

if( idtypy !=’01’) errcode=’E5’ 记录错误文件fp\_err\_a, continue;

if( idno=NULL || idno.lenth !=15 || idno.lenth !=18 ||...)

errcode=’E2’ 记录错误文件fp\_err\_a, continue;

if(bankno!=issbank ) errcode=’E1’ 记录错误文件fp\_err\_a, continue;

记录该条记录在正确数据文件中

}

# 中国银行数据导入临时表

根据 preload 文件格式建立账户和关联关系两个临时表，账户请求文件的索引为 “文件上送银行 + 身份证号码 + 账户余额 + 开户时间”；关联关系请求文件的索引为“文件上送银行 + 身份证号码 + 医联卡号”

tbl\_ tmp\_mergereq \_$$$$$$$$ 账户信息请求文件预处理后导入临时表

tbl\_tmp\_ mergerelareq \_$$$$$$$$ 账户关联关系请求文件预处理后导入临时表

导入如果出错，请保留错误记录文件，oracle会自动产生

脚本程序如下：

bankid=`expr $fname : 'ACCT-\([0-9]\*\)-M.\*'`

echo $bankid

cd ${MERGE\_HOME}/prefile

relname=`ls \*PRELOAD\*|grep ${bankid}-MERGERELAREQ`

cd ${MERGE\_HOME}/sql/ctl

echo "generate load\_tmp\_acct\_$bankid.ctl file ..."

echo ">>>>>FILE[$fname]<<<<<"

sed "s/TMP\_ACCT/$fname/;s/XXXXXXXX/${bankid}/" load\_tmp\_acctinf.ctl >load\_tmp\_acct\_$bankid.ctl

echo "process load tmp acctinfo[$bankid]..."

sqlldr $DB\_USER\_NAME/$DB\_USER\_PWD@$ORACLE\_SID control=load\_tmp\_acct\_$bankid.ctl log=load\_tmp\_acct\_$bankid.log

echo "generate load\_tmp\_rel\_$bankid.ctl file ..."

echo ">>>>>FILE[$relname]<<<<<"

sed "s/TMP\_REL/$relname/;s/XXXXXXXX/${bankid}/" load\_tmp\_relinf.ctl >load\_tmp\_rel\_$bankid.ctl

echo "process load tmp relinfo[$bankid]..."

sqlldr $DB\_USER\_NAME/$DB\_USER\_PWD@$ORACLE\_SID control=load\_tmp\_rel\_$bankid.ctl log=load\_tmp\_rel\_$bankid.log

echo "load $bankid tmp table complete .."

# 中国银行账户和关联关系检查

交换中心对请求文件进行检查后，对有如下错误的记录统一发原银行，账户记录有误则同步将该账户对应的所有关联关系拒绝（可能有手工检查操作），关联关系有误则将同步将该关联关系对应的所有账户发回。

1. 检查是文件中的账户和关联关系中的账户是否一一对应
2. 账户是否重复

3.医疗卡是否重复

E8： 账户多余（账户无对应关联关系）

E9: 关联关系多余（关联关系无对应账户）

EA: 账户重复（身份证重复）

EB： 医疗卡重复（关联关系重复或一卡多账户）

预处理结果

账户信息处理失败应答文件ACCT-$$$$$$$$-MERGEREQ-YYYYMMDD-ERR（中行）

账户关联关系处理失败应答文ACCT-$$$$$$$$-MERGERELAREQ-YYYYMMDD-ERR（中行）

## 检查是文件中的账户和关联关系中的账户是否一一对应

SQL脚本如下：

检查账户信息没有关联医疗卡

update tbl\_tmp\_mergereq\_${bank} set ErrCode='E8' where IdNo In \

(select distinct Id1 from \

(select a.IdNo Id1, b.IdNo Id2 from \

tbl\_tmp\_mergereq\_${bank} a left join \

tbl\_tmp\_mergerelareq\_${bank} b on a.IdNo = b.IdNo) \

where Id2 is null);

检查医疗卡没有关联账户

update tbl\_tmp\_mergerelareq\_${bank} set ErrCode='E9' where IdNo In \

(select distinct Id2 from (select a.IdNo Id1, b.IdNo Id2 from \

tbl\_tmp\_mergereq\_${bank} a right join \

tbl\_tmp\_mergerelareq\_${bank} b on a.IdNo = b.IdNo) \

where Id1 is null);

## 账户是否重复

SQL脚本如下：

账户信息重复：

update tbl\_tmp\_mergereq\_${bank} set ErrCode='EA' where IdNo In \

(select IdNo from tbl\_tmp\_mergereq\_${bank} \

Where ErrCode is null \

Group By IdNo Having COUNT(\*) > 1);

## 医疗卡是否重复

SQL脚本如下：

医疗卡重复：

update tbl\_tmp\_mergerelareq\_${bank} set ErrCode='EB' where HisCard In \

(select HisCard from tbl\_tmp\_mergerelareq\_${bank} \

Where ErrCode is null \

Group By HisCard Having COUNT(\*) > 1);

## 账户记录有误则同步将该账户对应的所有关联关系拒绝

SQL脚本如下：

update tbl\_tmp\_mergerelareq\_${bank} set ErrCode='EA' where IdNo In \

(select IdNo from tbl\_tmp\_mergereq\_${bank} a, tbl\_tmp\_mergerelareq\_${bank} b\

Where a.ErrCode=‘EA’and b.Error is null and a.IdNo = b.IdNo);

## 关联关系有误则将同步将该关联关系对应的所有账户发回

update tbl\_tmp\_mergereq\_${bank} set ErrCode='EB' where IdNo In \

(select IdNo from tbl\_tmp\_mergerelareq\_${bank} a, tbl\_tmp\_mergereq\_${bank} b\

Where a.ErrCode=‘EB’and b.Error is null and a.IdNo = b.IdNo);

# 逻辑处理

## 账户已在交换中心开户（工行开户)

**账户处理**：医院通知持卡人销户退款，交换中心不作变动。（NULL-中行-A1）

SQL脚本实现如下：

update tbl\_tmp\_mergereq\_${bank} set ErrCode='A1' where IdNo In \

(select distinct Id1 from \

(select a.IdNo Id1,a.errcode errA, b.IdNo Id2, b.bankno bankno \

from tbl\_tmp\_mergereq\_${bank} a left join tbl\_acct\_inf b on a.IdNo = b.IdNo\

where Id2 is not null and (errA is null or errA=' ') and bankno=${bank}\

);

**关联关系处理**：

|  |  |  |  |
| --- | --- | --- | --- |
| **就医卡关联关系状态** | **交换中心处理** | **文件下发** | **ErrorCode** |
| 就医卡在交换中心不存在关联关系 | 在交换中心增加该就医卡与该账户的关联关系 | 无 | R1 |
| 就医卡在交换中心存在关联关系，但为“解绑”状态 | 在交换中心更新该就医卡所绑定的账户信息 | 无 | R2 |
| 就医卡在交换中心存在关联关系，且关联的账户为本账户 | 交换中心关联关系不作变动，该就医卡不作处理 | 无 | 无 |
| 就医卡在交换中心存在关联关系，且关联的账户为其他账户 | 无 | 无 |

## 账户已在交换中心开户(其他银行开户)

**账户处理**：医院通知持卡人销户退款，交换中心不作变动。（NULL-中行-A2）

SQL脚本实现如下：

update tbl\_tmp\_mergereq\_${bank} set ErrCode='A2' where IdNo In \

(select distinct Id1 from \

(select a.IdNo Id1,a.errcode errA, b.IdNo Id2, b.bankno bkNo \

From tbl\_tmp\_mergereq\_${bank} a left join tbl\_acct\_inf b on a.IdNo = b.IdNo)\

where Id2 is not null and (errA is null or errA=' ') and bkNo!=${bank});

**关联关系处理：**

|  |  |  |  |
| --- | --- | --- | --- |
| **就医卡关联关系状态** | **交换中心处理** | **文件下发** | **ErrorCode** |
| 就医卡在交换中心不存在关联关系 | 交换中心关联关系不作变动，该就医卡不作处理 | 无 | 无 |
| 就医卡在交换中心存在关联关系，但为“解绑”状态 | 无 | 无 |
| 就医卡在交换中心存在关联关系，且关联的账户为本账户 | 无 | 无 |
| 就医卡在交换中心存在关联关系，且关联的账户为其他账户 | 无 | 无 |

## 账户未在交换中心开户

**账户处理**：在交换中心开户，开户行为中国工商银行。（开户-工行-A3）

SQL脚本实现如下：

update tbl\_tmp\_mergereq\_${bank} set ErrCode='A3',bankno=‘01022900’where IdNo NOT In\

(select distinct Id1 from \

(select a.IdNo Id1,a.errcode errA, b.IdNo Id2, b.bankno bankno from \

tbl\_tmp\_mergereq\_${bank} a left join tbl\_acct\_inf b on a.IdNo = b.IdNo)\

where Id2 is not null and (errA is null or errA=' ') );

**关联关系处理：**

|  |  |  |  |
| --- | --- | --- | --- |
| **就医卡关联关系状态** | **交换中心处理** | **文件下发** | **ErrorCode** |
| 就医卡在交换中心不存在关联关系 | 在交换中心增加该就医卡与该账户的关联关系 | 工行 | R3 |
| 就医卡在交换中心存在关联关系，但为“解绑”状态 | 在交换中心更新该就医卡所绑定的账户信息 | 工行 | R4 |
| 就医卡在交换中心存在关联关系，且关联的账户为其他账户 | 交换中心关联关系不作变动，该就医卡不作处理 | 无 | 无 |

## 关联关系绑定检查

1.将临时表中所有需要记入交换中心的关联关系进行标记（即存在且为绑定状态的不需要记入）

SQL脚本实现如下：

UPDATE tbl\_tmp\_mergerelareq\_${bank} SET ErrCode='LS' WHERE Hiscard NOT IN \

(SELECT distinct His1 FROM \

(SELECT a.Hiscard His1,a.errcode errA, b.Hiscard His2,b.hiscard\_state state\

FROM tbl\_tmp\_mergerelareq\_${bank} a LEFT JOIN \

tbl\_card\_join\_acct b ON a.Hiscard = b.Hiscard ) \

WHERE His2 IS NOT NULL and (errA is null OR errA=’ ’) AND state=‘绑定’\

);

2.已在交换中心开户的账户对应的就医卡在交换中心不存

UPDATE tbl\_tmp\_mergerelareq\_${bank} SET ErrCode='R1' WHERE IdNo IN \

(select b.IdNo from tbl\_tmp\_mergereq\_${bank} a, tbl\_tmp\_mergerelareq\_${bank} b\

Where a.IdNo=b.IdNo and a.ErrCode=‘A1’\

) and ErrCode=‘LS’;

3.已在交换中心开户的账户对应的就医卡在交换中心已解除关联

UPDATE tbl\_tmp\_mergerelareq\_${bank} SET ErrCode='R2' WHERE IdNo NOT IN \

(select b.IdNo from tbl\_tmp\_mergereq\_${bank} a, tbl\_tmp\_mergerelareq\_${bank} b\

Where a.IdNo=b.IdNo and a.ErrCode=‘A1’\

) and ErrCode=‘LS’;

4.未在交换中心开户的账户对应的就医卡在交换中心不存

UPDATE tbl\_tmp\_mergerelareq\_${bank} SET ErrCode='R3',bankno=‘01022900’ WHERE IdNo IN\

(select b.IdNo from tbl\_tmp\_mergereq\_${bank} a, tbl\_tmp\_mergerelareq\_${bank} b\

Where a.IdNo=b.IdNo and a.ErrCode=‘A3’\

) and ErrCode=‘LS’;

5.未在交换中心开户的账户对应的就医卡在交换中心已解除关联

UPDATE tbl\_tmp\_mergerelareq\_${bank} SET ErrCode='R4',bankno=‘01022900’\

WHERE IdNo NOT IN\

(select b.IdNo from tbl\_tmp\_mergereq\_${bank} a, tbl\_tmp\_mergerelareq\_${bank} b\

Where a.IdNo=b.IdNo and a.ErrCode=‘A3’\

) and ErrCode=‘LS’;

标记码：

A1: 已在交换中心开户的中行账户

A2： 已在交换中心开户的他行账户

A3： 未在交换中心开户的账户

R1: 已在交换中心开户的账户对应的就医卡在交换中心不存

R2: 已在交换中心开户的账户对应的就医卡在交换中心已解除关联

R3: 未在交换中心开户的账户对应的就医卡在交换中心不存

R4: 未在交换中心开户的账户对应的就医卡在交换中心已解除关联

# 结果处理

## 交换中心处理合并：

A3 在交换中心开户，开户行为中国银行

SQL语句：

insert into tbl\_acct\_inf (idtype,idno,bankid,unitno,name,gender,telno,addr,open\_date,last\_mod\_date,state)

(select idtype,idno,bank,hospid,name,gender,telno,addr,stime ,NULL,'A' from

tbl\_tmp\_mergereq\_${bank} and ErrCode=‘A3’ );

R1/R3 账户表增加该就医卡与该账户的关联关系

SQL语句：

insert into tbl\_card\_join\_acct

(select b.IdType, b.IdNo, b.HisCard, b.issbank, a.stime, NULL,'0'

from tbl\_tmp\_mergereq\_${bank} a, tbl\_tmp\_mergerelareq\_${bank} b\

Where a.IdNo=b.IdNo and (b.ErrCode='R1' or b.ErrCode='R3')

);

R2/R4 关联关系表更新该就医卡所绑定的账户信息

SQL语句：

update tbl\_card\_join\_acct set hiscard\_state=‘绑定’，xxx=NULL \

issbank=‘01022900’，stime=SYADATE Where hiscard IN ( \

SELECT hiscard FROM tbl\_tmp\_mergerelareq\_${bank} and ErrCode=‘R2’ \

);

update tbl\_card\_join\_acct set hiscard\_state=‘绑定’，xxx=NULL \

issbank=‘01042900’，stime=SYADATE Where hiscard IN( \

SELECT hiscard FROM tbl\_tmp\_mergerelareq\_${bank} and ErrCode=‘R4’ \

);

## 文件处理结果下发中行

错误应答E1-EB

账户信息处理失败应答文件ACCT-$$$$$$$$-MERGEREQ-YYYYMMDD-ERR

账户关联关系处理失败应答文ACCT-$$$$$$$$-MERGERELAREQ-YYYYMMDD-ERR

正常应答：A1/A2

账户信息处理结果文件ACCT-$$$$$$$$-MERGERRSP-YYYYMMDD

## 文件处理结果下发工行

正常应答：

A3 账户信息处理结果文件ACCT-$$$$$$$$-MERGERRSP-YYYYMMDD

R3/R4 账户关联关系处理结果文件ACCT-$$$$$$$$-MERGERELARSP-YYYYMMDD

# 错误类型：