1. 查询用户欠费

create or replace procedure queryCost(clientno in client.id%type, smoney out number) is

basicfee number(7,2);

yearmonth receivables.yearmonth%type;

dtype device.type%type;

sysyear number;

recordyear number;

sysmonth number;

recordmonth number;

cursor temp\_cursor is

select r.basicfee,r.yearmonth,d.type

from client c, device d, receivables r

where c.id = d.clientid

and d.deviceid = r.deviceid

and r.flag = 0*--flag=0表示没交*

and c.id = clientno;

begin

smoney:=0;

open temp\_cursor;

loop

fetch temp\_cursor into basicfee,yearmonth,dtype;

exit when temp\_cursor%notfound;

smoney:=smoney+basicfee;

smoney:=smoney+basicfee\*0.08;

*--附加费用*

if dtype = '01' then

smoney:= smoney + basicfee\*0.1;

else

smoney:=smoney + basicfee\*0.15;

end if;

sysyear:=to\_number(to\_char(sysdate,'yyyy'));

sysmonth:=to\_number(to\_char(sysdate,'mm'));

recordyear:=to\_number(to\_char(to\_date(yearmonth,'yyyymm'),'yyyy'));

recordmonth:=to\_number(to\_char(to\_date(yearmonth,'yyyymm'),'mm'));

if dtype = '02' then

*--当类型为02时存在跨年问题*

if sysyear = recordyear and sysmonth > recordmonth then

smoney:= smoney + round(sysdate - add\_months(to\_date(yearmonth,'yyyymm'),1))\*basicfee\*0.002;

elsif sysyear > recordyear then

*--如果跨年那就分成当年和跨年部分算*

smoney:= smoney + round(to\_date(1231,'yyyymm') - add\_months(to\_date(recordmonth,'yyyymm'),1))\*basicfee\*0.002;

smoney:= smoney + round(sysdate - add\_months(to\_date(recordyear,'yyyy'),12))\*basicfee\*0.003;

end if;

else

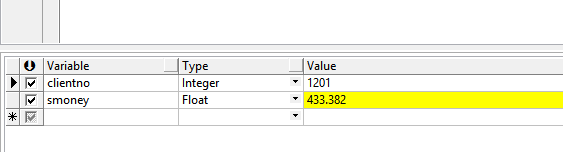
*--类型为01直接计算*

smoney:=smoney + round(sysdate - add\_months(to\_date(yearmonth,'yyyymm'),1))\*basicfee\*0.001;

end if;

end loop;

end queryCost;



1. 给设备缴费

2.1 第一个过程：将缴费信息记录到payfee中

create or replace procedure payrecord(deviceno in device.deviceid%type, fee in number,

ok out char,temp\_bankserial out char) is

paydate payfee.paydate%type;

payid number;

begin

select count(\*)

into payid

from payfee;

paydate := to\_date(to\_char(sysdate-1*--sysdate减1模拟前一天*

,'yyyyMMdd'),'yyyy/MM/dd');

insert into payfee(payfee.type,payfee.id,payfee.deviceid,payfee.paymoney,payfee.paydate,bankcode,bankserial)

values('2000',payid+100,deviceno,fee,paydate,19,payid+100);

commit;

ok:='缴费成功!';

temp\_bankserial:=payid+100;

exception

when no\_data\_found then

dbms\_output.put\_line('改ID不存在');

when others then

dbms\_output.put\_line('其他错误');

end payrecord;

2.2 第二个过程：模拟第二天，对相应设备表、应收费用表进行更新将payfee中的信息设为不可冲正

create or replace procedure inbank is

recorddate receivables.yearmonth%type;

balance device.balance%type;

total device.balance%type;

month\_pay device.balance%type;

currenttype device.type%type;

currentid device.deviceid%type;

basicfee device.balance%type;

flag receivables.flag%type;

sysyear number;

sysmonth number;

recordyear number;

recordmonth number;

*--找到前一天发生变动的设备id*

cursor id\_cursor is

select d.deviceid,p.paymoney

from device d,payfee p

where to\_char(p.paydate,'yyyyMMdd')

=to\_char(sysdate-1,'yyyyMMdd')

and d.deviceid=p.deviceid

and (p.type='2000' or p.type='2002');

*--找到应收费用中的日期与应交的金额*

cursor month\_cursor is

select r.basicfee,r.yearmonth,r.flag

from receivables r

where r.deviceid=currentid

and flag = 0*--找出未交的部分*

order by r.yearmonth desc;

begin

open id\_cursor;

loop

fetch id\_cursor into currentid,total;

exit when id\_cursor%notfound;

select device.balance,device.type

into balance,currenttype

from device

where device.deviceid=currentid;

total:=total+balance;

open month\_cursor;

loop

fetch month\_cursor into basicfee,recorddate,flag;

exit when month\_cursor%notfound;

*--对当前的设备进行每个月的缴费遍历*

*--附加费用*

month\_pay:=basicfee\*1.08;

if currenttype = '01' then

month\_pay:=month\_pay+basicfee\*0.1;

else

month\_pay:=month\_pay+basicfee\*0.15;

end if;

*--违约金*

sysyear:=to\_number(to\_char(sysdate,'yyyy'));

sysmonth:=to\_number(to\_char(sysdate,'mm'));

recordyear:=to\_number(to\_char(to\_date(recorddate,'yyyymm'),'yyyy'));

recordmonth:=to\_number(to\_char(to\_date(recorddate,'yyyymm'),'mm'));

if currenttype = '02' then

if sysyear = recordyear and sysmonth > recordmonth then

month\_pay:= month\_pay + round(sysdate - add\_months(to\_date(recorddate,'yyyymm'),1))\*basicfee\*0.002;

elsif sysyear > recordyear then

*--未跨年部分按照0.002*

month\_pay:= month\_pay + round(to\_date(1231,'yyyymm') - add\_months(to\_date(recordmonth,'yyyymm'),1))\*basicfee\*0.002;

*--跨年部分按照0.003*

month\_pay:= month\_pay + round(sysdate - add\_months(to\_date(recordyear,'yyyy'),12))\*basicfee\*0.003;

end if;

else

month\_pay:=month\_pay + round(sysdate - add\_months(to\_date(recorddate,'yyyymm'),1))\*basicfee\*0.001;

end if;

*--当前的total不够交放入余额并记录下payfee*

if month\_pay>total then

update device set device.balance=device.balance+total where device.deviceid = currentid;

commit;

update payfee set payfee.type='2001' where payfee.deviceid=currentid;

commit;

exit;

else

*--够交则从total扣除应缴费用*

total:=total-month\_pay;

update receivables set flag=1 where receivables.deviceid = currentid

and receivables.yearmonth = recorddate;

commit;

end if;

end loop;

close month\_cursor;

*--如果交完钱还有剩就放回余额*

if total>0 then

update device set device.balance=total where device.deviceid = currentid;

commit;

update payfee set payfee.type='2001' where payfee.deviceid=currentid;

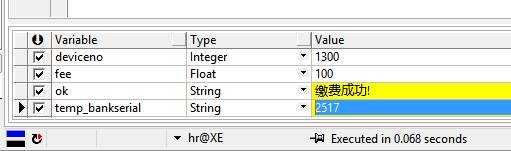
commit;

end if;

end loop;

close id\_cursor;

end inbank;



1. 冲正操作

create or replace procedure reversal(deviceid in device.deviceid%type,

serial in payfee.bankserial%type, reversalresult out char) is

opdate payfee.paydate%type;

payid number;

paymoney payfee.paymoney%type;

begin

select p.paymoney

into paymoney

from payfee p

where p.bankserial=serial

and p.type='2000'

and p.deviceid=deviceid;

*--没有找到就直接*

if paymoney is null then

reversalresult:='冲正失败！';

else

select count(\*)

into payid

from payfee;

opdate := to\_date(to\_char(sysdate-1,'yyyyMMdd'),'yyyy/MM/dd');

insert into payfee(payfee.type,payfee.id,payfee.deviceid,payfee.paymoney,payfee.paydate,bankcode,bankserial)

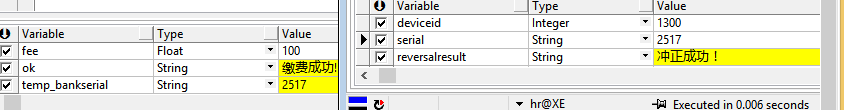
values('2002',payid+100,deviceid,0-paymoney,opdate,19,serial);

commit;

reversalresult:='冲正成功！';

end if ;

end reversal;



1. 对总账

create or replace procedure checkTotal(bankcode IN CHECKRESULT.BANKCODE%type,

totalnum in CHECKRESULT.BANKTOTALCOUNT%type,

totalmoney in CHECKRESULT.BANKTOTALMONEY%type,

checkdate in char,

resultchar out char) is

id checkresult.id%type;

checked\_totalnum checkresult.banktotalcount%type;

checked\_totalmoney checkresult.banktotalmoney%type;

begin

select count(\*),sum(payfee.paymoney)

into checked\_totalnum,checked\_totalmoney

from payfee

where to\_char(payfee.paydate,'yyyymmdd')=

to\_char(to\_date(checkdate,'yyyymmdd'),'yyyymmdd')

and bankcode=payfee.bankcode

and payfee.type='2001';

select to\_char(count(\*))

into id

from checkresult;

if checked\_totalmoney=totalmoney

and checked\_totalnum=totalnum then

insert into checkresult

values(id,checkdate,bankcode,totalnum,totalmoney,checked\_totalnum,checked\_totalmoney);

commit;

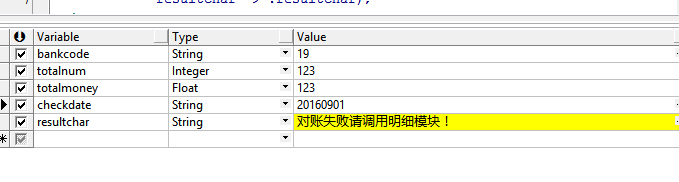
resultchar:='对账成功！';

else

resultchar:='对账失败请调用明细模块！';

end if;

end checkTotal;



1. 对明细

create or replace procedure checkdetail(checkdate in char,bankcode in char) is

ele\_serial payfee.bankserial%type;

ele\_money payfee.paymoney%type;

bank\_money bankrecord.payfee%type;

id number;

cursor ele\_cursor is

select payfee.bankserial, payfee.paymoney

from payfee

where to\_char(payfee.paydate,'yyyymmdd') =

to\_char(to\_date(checkdate,'yyyymmdd'),'yyyymmdd')

and payfee.type='2001'

and payfee.bankcode = bankcode;

begin

*--用payfee中的记录与bankrecord中记录逐一比对*

open ele\_cursor;

loop

fetch ele\_cursor into ele\_serial,ele\_money;

exit when ele\_cursor%notfound;

select bankrecord.payfee

into bank\_money

from bankrecord

where bankrecord.bankserial = ele\_serial;

select count(\*)

into id

from check\_exception;

if bank\_money is null then

*--payfee中有记录，银行没有 类型设置为111*

insert into check\_exception

values(id+1,to\_date(checkdate,'yyyy/mm/dd'),

bankcode,ele\_serial,null,ele\_money,'111');

commit;

else

if bank\_money<>ele\_money then

*--都有记录但是钱不一样 类型设置为222*

insert into check\_exception

values(id+1,to\_date(checkdate,'yyyy/mm/dd'),

bankcode,ele\_serial,bank\_money,ele\_money,'222');

commit;

end if;

end if;

end loop;

close ele\_cursor;

end checkdetail;

