（1）查询

涉及到的表：

    客户信息表、计费设备表、应收费用表

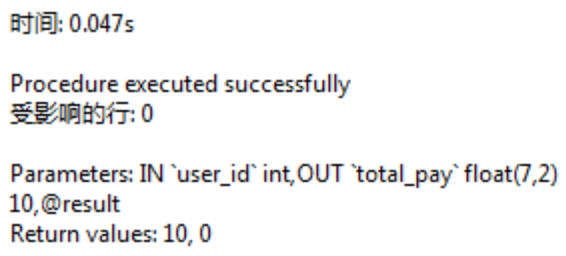
输入：

    客户编号

输出：

    应缴费金额

```
SET FOREIGN_KEY_CHECKS=0;

-- ----------------------------
-- Procedure structure for bank1
-- ----------------------------
DROP PROCEDURE IF EXISTS `bank1`;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `bank1`(IN `user_id`
int,OUT `total_pay` float(7,2))
BEGIN
     #Routine body goes here...
DECLARE user_info VARCHAR(20);
DECLARE done int default -1;  -- to control the loop
DECLARE tmp0 int;
DECLARE tmp1 char(2);
DECLARE tmp2 FLOAT(7,2);


DECLARE user_device CURSOR for SELECT device.deviceid,
device.type, receivables.basicfee

FROM (receivables JOIN device ON
device.deviceid=receivables.deviceid)

WHERE device.clientid=user_id and receivables.flag=1;

DECLARE continue handler for not found set done=1;

-- SELECT client.`name` INTO user_info FROM client, device WHERE
device.clientid=client.id and client.id = 11;
-- SELECT * FROM (receivables LEFT JOIN device ON
device.deviceid=receivables.deviceid) WHERE
device.clientid=user_id;
-- SET total_pay=2;
OPEN user_device;
set total_pay = 0;
myloop:LOOP


        fetch user_device into tmp0,tmp1,tmp2;
                     -- exit
        if done = 1 then
        leave myLoop;
        end if;

        /* do something */
                      set total_pay = total_pay + tmp2*1.08;

                      if tmp1 = "01" then
                      set total_pay = total_pay + tmp2*0.1;
                      end if;
                      if tmp1 = "02" then
                      set total_pay = total_pay + tmp2*0.15;
                      end if;

        -- output
                      -- select tmp0,tmp1 ;

END LOOP myloop;

CLOSE user_device;
END
;;
DELIMITER ;
```



```
时间: 0.047s

Procedure executed successfully
受影响的行: 0

Parameters: IN `user_id` int,OUT `total_pay` float(7,2)
10,@result
Return values: 10, 0
```

（2）缴费

涉及到的表：

　　计费设备表、应收费用表、用户交费表

输入：

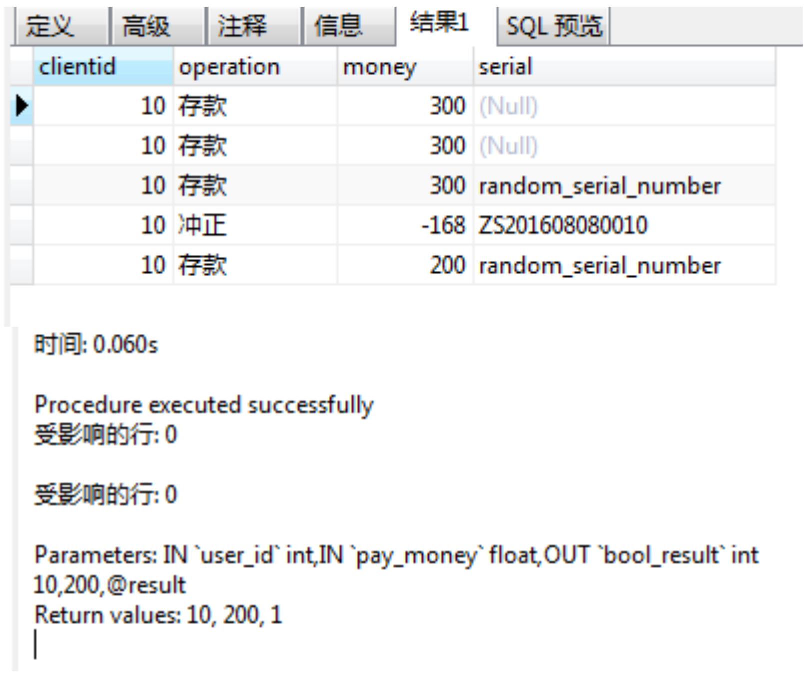
　　客户号，缴费金额

输出：

　　成功，失败

```
-- ----------------------------
-- Procedure structure for bank2
-- ----------------------------
DROP PROCEDURE IF EXISTS `bank2`;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `bank2`(IN `user_id`
int,IN `pay_money` float,OUT `bool_result` int)
```

```
BEGIN
	#Routine body goes here...
DECLARE money_should_pay int DEFAULT 0;


CALL bank1(user_id, money_should_pay);

if pay_money > money_should_pay THEN
set bool_result = 1;
else
set bool_result = 0;
END IF;
INSERT INTO user_pay VALUES(user_id, "瀛樻T", pay_money,
"random_serial_number");
SELECT * FROM user_pay;
END
;;
DELIMITER ;
```



定义 | 高级 | 注释 | 信息 | 结果1 | SQL 预览

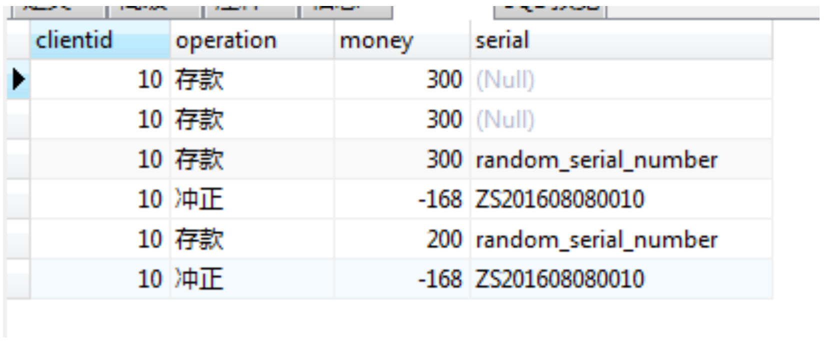
clientid	operation	money	serial
10	存款	300	(Null)
10	存款	300	(Null)
10	存款	300	random_serial_number
10	冲正	-168	ZS201608080010
10	存款	200	random_serial_number

```
时间: 0.060s

Procedure executed successfully
受影响的行: 0

受影响的行: 0

Parameters: IN `user_id` int,IN `pay_money` float,OUT `bool_result` int
10,200,@result
Return values: 10, 200, 1
```

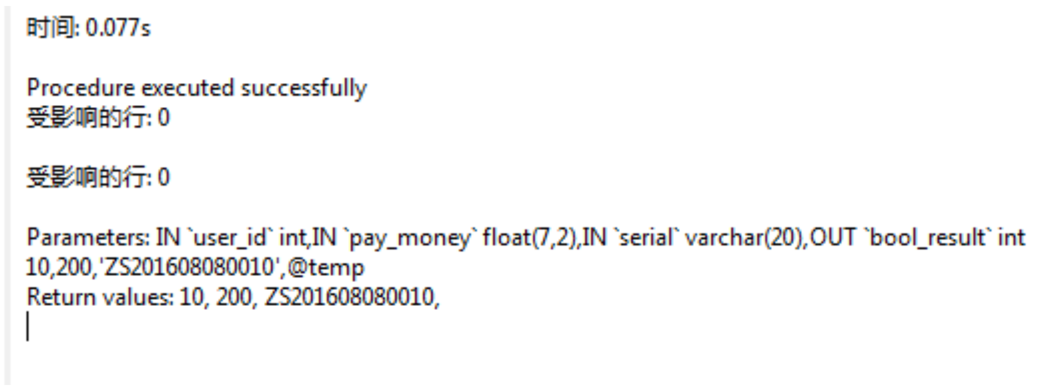
（3）冲正
涉及到的表：
　　计费设备表、应收费用表、用户交费表
输入：
　　客户号，冲正金额, 原流水号
输出：
　　成功，失败

-- ----------------------------

```
-- Procedure structure for bank3
-- ----------------------------
DROP PROCEDURE IF EXISTS `bank3`;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `bank3`(IN `user_id`
int,IN `pay_money` float(7,2),IN `serial` varchar(20),OUT `bool_result` int)
BEGIN
	#Routine body goes here...
DECLARE serial_number VARCHAR(20);
DECLARE payfee_paymoney int DEFAULT 0;
DECLARE device_deviceid int DEFAULT 0;

SELECT payfee.bankserial, payfee.paymoney, device.deviceid
	FROM (payfee JOIN device ON device.deviceid=payfee.deviceid)
  WHERE payfee.bankserial = serial INTO serial_number, payfee_paymoney,
device_deviceid;

set payfee_paymoney = payfee_paymoney*(-1);

INSERT INTO user_pay VALUES(user_id, "鍐叉",  payfee_paymoney,
serial_number);

UPDATE receivables set receivables.flag=0 WHERE
receivables.deviceid=device_deviceid;

SELECT * FROM user_pay;

-- 10,200,'ZS201608080010',@temp
END
;;
DELIMITER ;
```



clientid	operation	money	serial
10	存款	300	(Null)
10	存款	300	(Null)
10	存款	300	random_serial_number
10	冲正	-168	ZS201608080010
10	存款	200	random_serial_number
10	冲正	-168	ZS201608080010



```
时间: 0.077s

Procedure executed successfully
受影响的行: 0

受影响的行: 0

Parameters: IN `user_id` int,IN `pay_money` float(7,2),IN `serial` varchar(20),OUT `bool_result` int
10,200,'ZS201608080010',@temp
Return values: 10, 200, ZS201608080010,
```

（4）对总账

涉及到的表：

用户交费表、对总账表

输入：

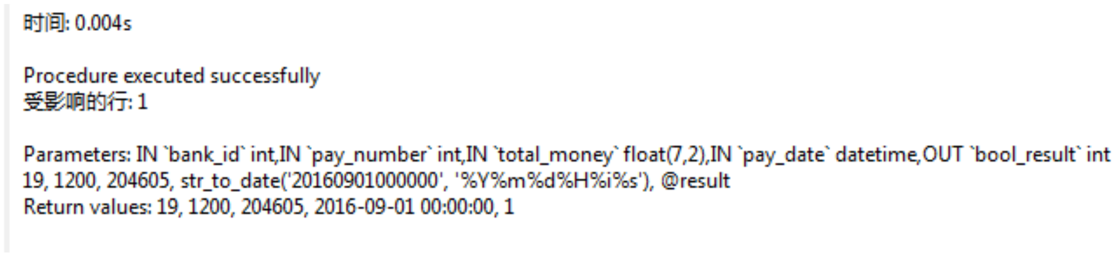
银行代码，总笔数，总金额，对帐日期

输出：

无

```
-- ----------------------------
-- Procedure structure for bank4
-- ----------------------------
DROP PROCEDURE IF EXISTS `bank4`;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `bank4`(IN `bank_id`
int,IN `pay_number` int,IN `total_money` float(7,2),IN `pay_date`
datetime,OUT `bool_result` int)
BEGIN
	#Routine body goes here...
-- DECLARE pay_num int DEFAULT 0;
DECLARE tmp0 int;
DECLARE tmp1 FLOAT(7,2);

SELECT COUNT(*), SUM(payfee.paymoney)

FROM payfee

WHERE payfee.bankcode=bank_id and payfee.paydate=pay_date
INTO tmp0,tmp1;

set bool_result = 0;
if tmp0=pay_number and tmp1=total_money THEN
set bool_result = 1;
end IF;

-- 19, 1200, 204605, str_to_date('20160901000000', '%Y%m%d%H%i%s'),
@result
```

```
END
;;
DELIMITER ;
```



```
时间: 0.004s

Procedure executed successfully
受影响的行: 1

Parameters: IN `bank_id` int,IN `pay_number` int,IN `total_money` float(7,2),IN `pay_date` datetime,OUT `bool_result` int
19, 1200, 204605, str_to_date('20160901000000', '%Y%m%d%H%i%s'), @result
Return values: 19, 1200, 204605, 2016-09-01 00:00:00, 1
```

（5）对明细账

涉及到的表：

　　用户缴费表、对账明细、对账异常表

输入：

　　银行对账明细记录集合

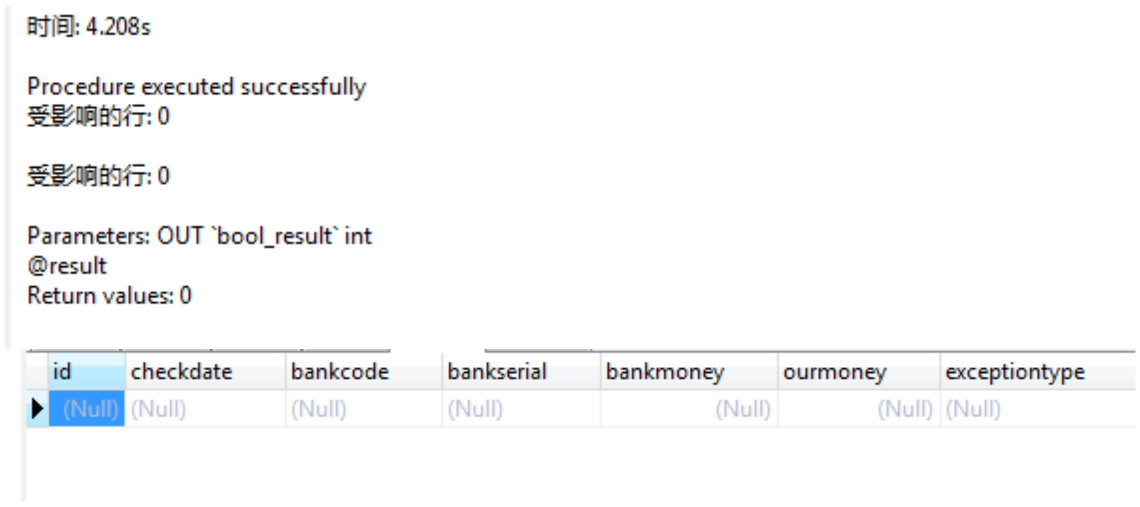
输出：

　　无

```
-- ----------------------------
-- Procedure structure for bank5
-- ----------------------------
DROP PROCEDURE IF EXISTS `bank5`;
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `bank5`(OUT
`bool_result` int)
BEGIN
	#Routine body goes here...
DECLARE line_of_except int default 0;

DECLARE done int default -1;

DECLARE tmp0 datetime;
DECLARE tmp1 char(2);
DECLARE tmp2 VARCHAR(20);
DECLARE tmp3 FLOAT(7,2);
DECLARE tmp4 VARCHAR(20);

DECLARE left_nul CURSOR for
	SELECT payfee.paydate, payfee.bankcode, payfee.bankserial,
payfee.paymoney FROM payfee LEFT JOIN bankrecord ON
payfee.bankserial=bankrecord.bankserial
	WHERE bankrecord.bankserial is NULL or payfee.bankserial is NULL;

DECLARE right_nul CURSOR for
```

```
         SELECT payfee.paydate, bankrecord.bankcode,
bankrecord.bankserial, bankrecord.payfee FROM payfee RIGHT JOIN
bankrecord ON payfee.bankserial=bankrecord.bankserial
         WHERE bankrecord.bankserial is NULL or payfee.bankserial is NULL;

DECLARE inner_nul CURSOR for
         SELECT payfee.paydate, payfee.bankcode, payfee.bankserial,
payfee.paymoney, bankrecord.payfee
         FROM payfee JOIN bankrecord ON
payfee.bankserial=bankrecord.bankserial;
--       WHERE (CAST(payfee.paymoney AS FLOAT))>10;


DECLARE continue handler for not found set done=1;

SELECT COUNT(*) FROM check_exception INTO line_of_except;


OPEN left_nul;
myloop:LOOP


       fetch left_nul into tmp0,tmp1,tmp2,tmp4;
                                    -- exit
       if done = 1 then
       leave myLoop;
       end if;
       set line_of_except = line_of_except + 1;
       INSERT INTO check_exception VALUES(line_of_except, tmp0, tmp1,
tmp2, tmp4, NULL, "bnk");
       -- output
                                    -- select tmp0,tmp1 ;

END LOOP myloop;
CLOSE left_nul;
SET done = -1;
SELECT COUNT(*) FROM check_exception INTO line_of_except;

OPEN right_nul;
myloop:LOOP


       fetch right_nul into tmp0,tmp1,tmp2,tmp3;
                                    -- exit
       if done = 1 then
       leave myLoop;
       end if;
       set line_of_except = line_of_except + 1;
       INSERT INTO check_exception VALUES(line_of_except, tmp0, tmp1,
tmp2, tmp3, NULL, "pay");
       -- output
```

```
                                    -- select tmp0,tmp1 ;

END LOOP myloop;
CLOSE right_nul;
SET done = -1;
SELECT COUNT(*) FROM check_exception INTO line_of_except;

OPEN inner_nul;
myloop:LOOP


       fetch inner_nul into tmp0,tmp1,tmp2,tmp3,tmp4;
                                    -- exit
       if done = 1 then
       leave myLoop;
       end if;
                                    if tmp3 <> tmp4 then
       set line_of_except = line_of_except + 1;
       INSERT INTO check_exception VALUES(line_of_except, tmp0, tmp1,
tmp2, tmp3, NULL, "USD");
                                    end if;
       -- output
                                    -- select tmp0,tmp1 ;

END LOOP myloop;
CLOSE inner_nul;

set bool_result = 0;
IF line_of_except > 0 THEN
set bool_result = 1;
end if;
SELECT * FROM check_exception;

END
;;
DELIMITER ;
```



```
时间: 4.208s

Procedure executed successfully
受影响的行: 0

受影响的行: 0

Parameters: OUT `bool_result` int
@result
Return values: 0
```

id	checkdate	bankcode	bankserial	bankmoney	ourmoney	exceptiontype
(Null)	(Null)	(Null)	(Null)	(Null)	(Null)	(Null)