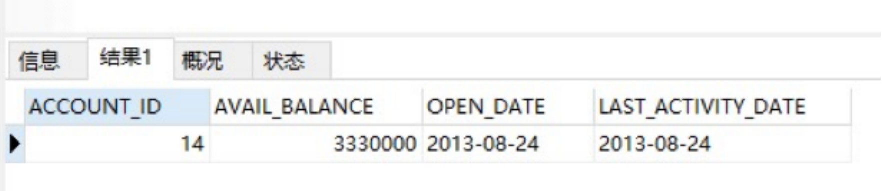
作业3-10185102142-李泽浩

（1）找出余额最多的账户信息，包括账户编号、可用余额、开户日期和最后活跃日期。输出属性包括ACCOUNT_ID, AVAIL_BALANCE, OPEN_DATE, LAST_ACTIVITY_DATE。

```sql
SELECT ACCOUNT_ID, AVAIL_BALANCE, OPEN_DATE, LAST_ACTIVITY_DATE
from account
ORDER BY AVAIL_BALANCE
DESC LIMIT 1;

SELECT ACCOUNT_ID, AVAIL_BALANCE, OPEN_DATE, LAST_ACTIVITY_DATE
from account
where AVAIL_BALANCE = (
  SELECT max(AVAIL_BALANCE)
  FROM account as a
  where ACCOUNT_ID = a.ACCOUNT_ID
);
```

结果如下：

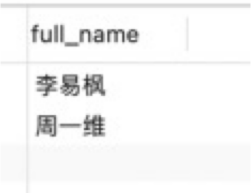


ACCOUNT_ID	AVAIL_BALANCE	OPEN_DATE	LAST_ACTIVITY_DATE
14	3330000	2013-08-24	2013-08-24

（2）找出所属部门与其上级领导所属部门不同的员工姓名，输出一个字段，属性命名为name（用CONCAT函数）。输出属性包括name。

```sql
SELECT concat(e1.LAST_NAME ,'',e1.FIRST_NAME) as full_name
FROM employee as e1, employee as e2
where e1.SUPERIOR_EMP_ID = e2.EMP_ID
      AND
  e1.DEPT_ID != e2.DEPT_ID;
```

结果如下：

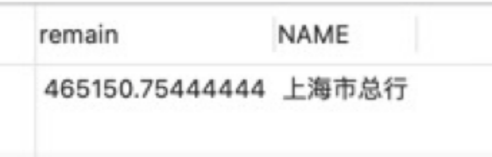


full_name
李易枫
周一维

（3）找出平均余额最多的支行名称（注：可能存在并列最多的情况）。输出属性包括NANE。

```
SELECT (sum(a.AVAIL_BALANCE) / count(a.OPEN_BRANCH_ID)) as res, b.NAME
FROM account as a, branch as b
where a.OPEN_BRANCH_ID = b.BRANCH_ID
GROUP BY b.`NAME`
ORDER BY res
desc LIMIT 1;
```

结果如下：

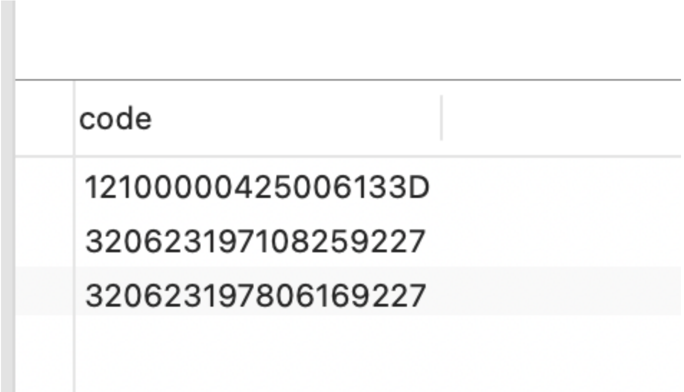


remain	NAME
465150.75444444	上海市总行

（4）找出身份证号以“3”开头的个人的（对私账户）和社会信用代码以“1”开头的客户（对公账户），将此属性命名为code；随后将其按照字符串从小到大排序，筛选出前3个。输出属性包括code。

```
SELECT t.code
FROM(
  SELECT i.ID_NUMBER as code, c1.CUST_TYPE_CD
  FROM individual as i JOIN customer as c1
    ON i.CUST_ID = c1.CUST_ID
  WHERE i.ID_NUMBER LIKE "3%"
  UNION
  SELECT b.CREDIT_CODE as code, c2.CUST_TYPE_CD
  FROM business as b JOIN customer as c2
    on b.CUST_ID = c2.CUST_ID
  WHERE b.CREDIT_CODE LIKE "1%"
) as t
ORDER BY code
ASC
LIMIT 3;
```

结果如下：

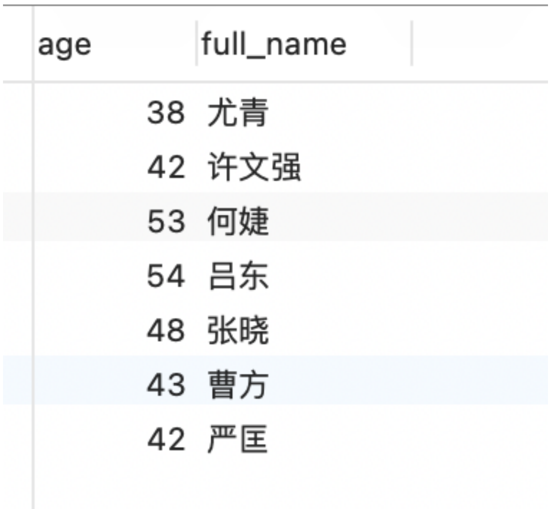


code
12100000425006133D
320623197108259227
320623197806169227

（5）找出至少拥有两个账户的个人客户（individual表）的姓名（命名为name）和年龄（命名为age）。（注：可使用FROM_DAYS、TO_DAYS和NOW函数计算年龄）。输出属性包括name和age。

```
SELECT FLOOR(DATEDIFF(CURRENT_DATE,i.BIRTH_DATE)/365) as age,
        concat(i.LAST_NAME ,'', i.FIRST_NAME)as full_name
FROM individual as i, account as a
where i.CUST_ID = a.CUST_ID
GROUP BY i.CUST_ID
HAVING  count(a.ACCOUNT_ID)>= 2;
```

结果如下：

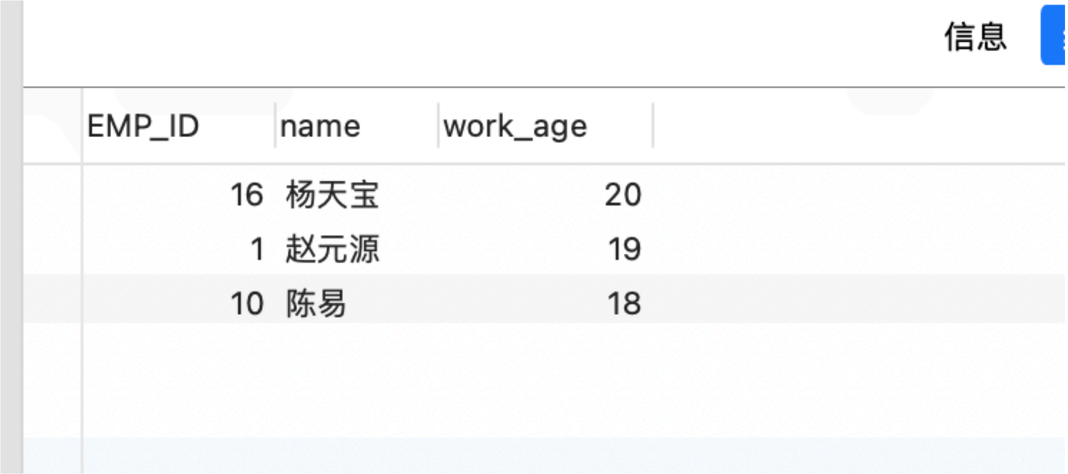


age	full_name
38	尤青
42	许文强
53	何婕
54	吕东
48	张晓
43	曹方
42	严匡

（6）找出工龄大于5年，且办理的执行交易数大于3次的员工信息，按其入职时间从先到后顺序输出。（注：可使用FROM_DAYS、TO_DAYS和NOW函数计算工龄）。输出属性包括EMP_ID。

```
SELECT EMP_ID, CONCAT(LAST_NAME, FIRST_NAME) as `name`,
      FLOOR(DATEDIFF(CURRENT_DATE, employee.START_DATE)/365) as work_age
FROM(
  SELECT EMP_ID, count(TXN_ID) as TXN_c
  FROM acc_transaction join employee
      on acc_transaction.TELLER_EMP_ID = employee.EMP_ID
  GROUP BY EMP_ID
  HAVING COUNT(TXN_ID) > 3
) as t NATURAL JOIN employee
WHERE FLOOR(DATEDIFF(CURRENT_DATE, employee.START_DATE)/365) > 5
ORDER BY work_age
DESC;
```

结果如下：

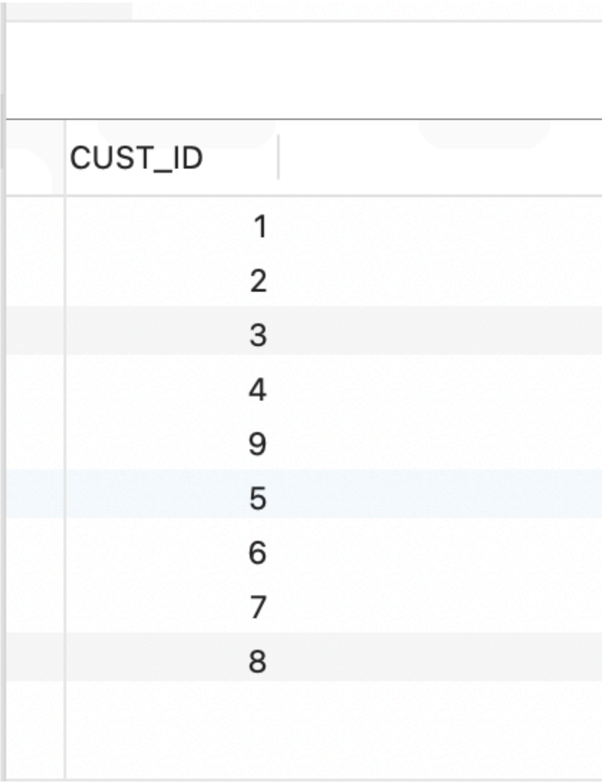


EMP_ID	name	work_age
16	杨天宝	20
1	赵元源	19
10	陈易	18

（7）查询至少购买了编号为“3”的客户所购买的所有产品的客户编号。输出属性包括CUST_ID。

```
SELECT CUST_ID
FROM account
WHERE PRODUCT_CD IN
(
  SELECT a.PRODUCT_CD
  FROM account as a
  WHERE a.CUST_ID = 3
)
GROUP BY cust_id;
```

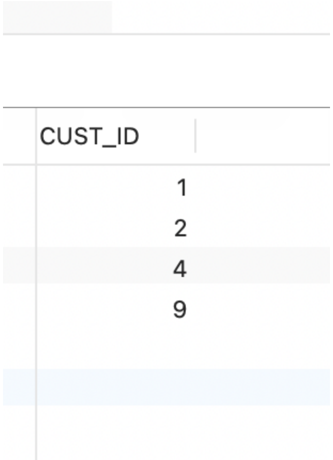
结果如下：



CUST_ID
1
2
3
4
9
5
6
7
8

（8）查询购买了编号为“3”的客户购买的产品完全相同的客户编号。输出属性包括CUST_ID。

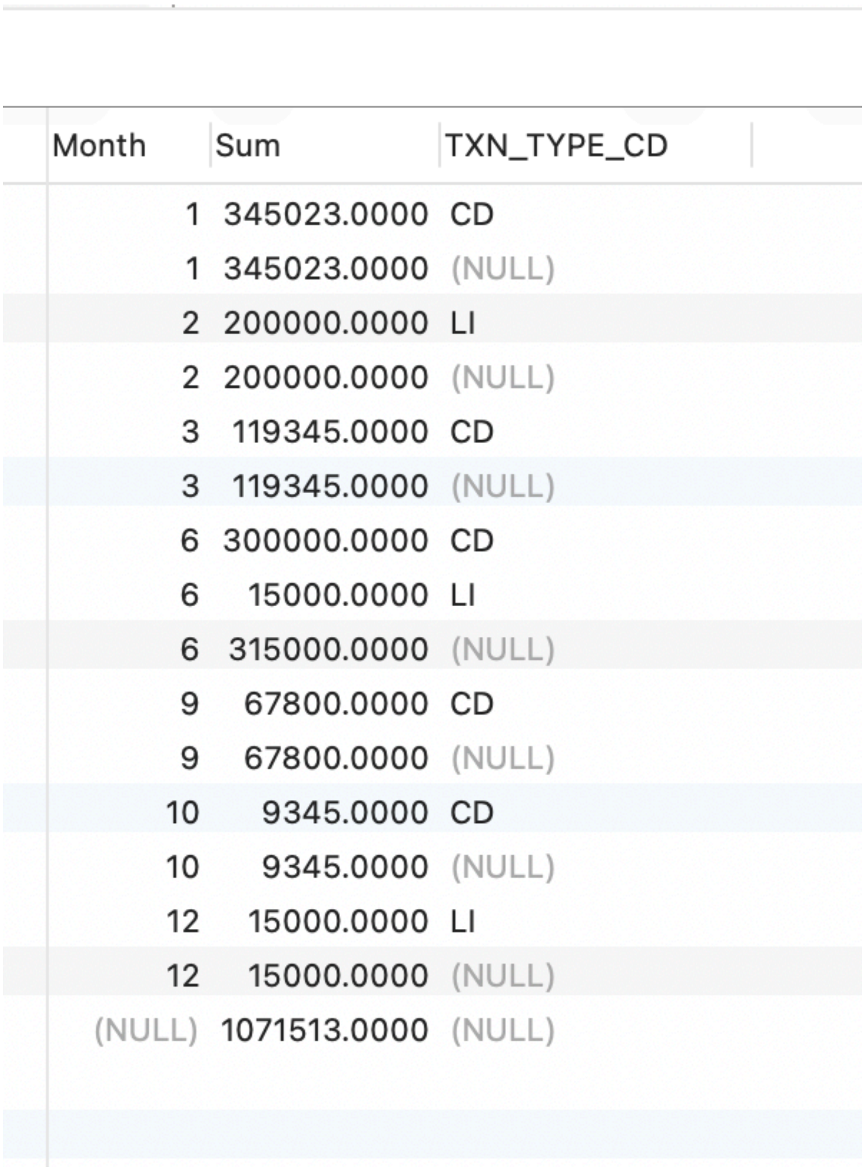
```
SELECT ID as CUST_ID
FROM
(
  SELECT T.ID1 as ID,
         COUNT(T.ID1) as NUM_ID
FROM
  (
    SELECT B.CUST_ID AS ID1,
           A.CUST_ID AS ID2,
           A.PRODUCT_CD AS PID
    FROM
    (
      SELECT CUST_ID,
             PRODUCT_CD
      FROM account
      WHERE CUST_ID = 3
    ) AS A
JOIN
    (
      SELECT CUST_ID,
             PRODUCT_CD
      FROM account
      WHERE CUST_ID != 3
    ) AS B
      ON A.PRODUCT_CD = B.PRODUCT_CD
      ORDER BY B.CUST_ID
  ) AS T
    GROUP BY T.ID1
  ORDER BY T.ID1
) AS Q
WHERE NUM_ID = (
  SELECT count(CUST_ID)
  FROM account
  WHERE CUST_ID = 3
)
```



CUST_ID
1
2
4
9

（9）请对2015年的交易历史进行报表汇总，具体查询输出要求为：首先对交易月份（命名为month）和交易类型编码进行分组，接着对交易月份进行分组，最后输出2015年销售总额。输出属性包括month，TXN_TYPE_CD，sum（销售总额）。

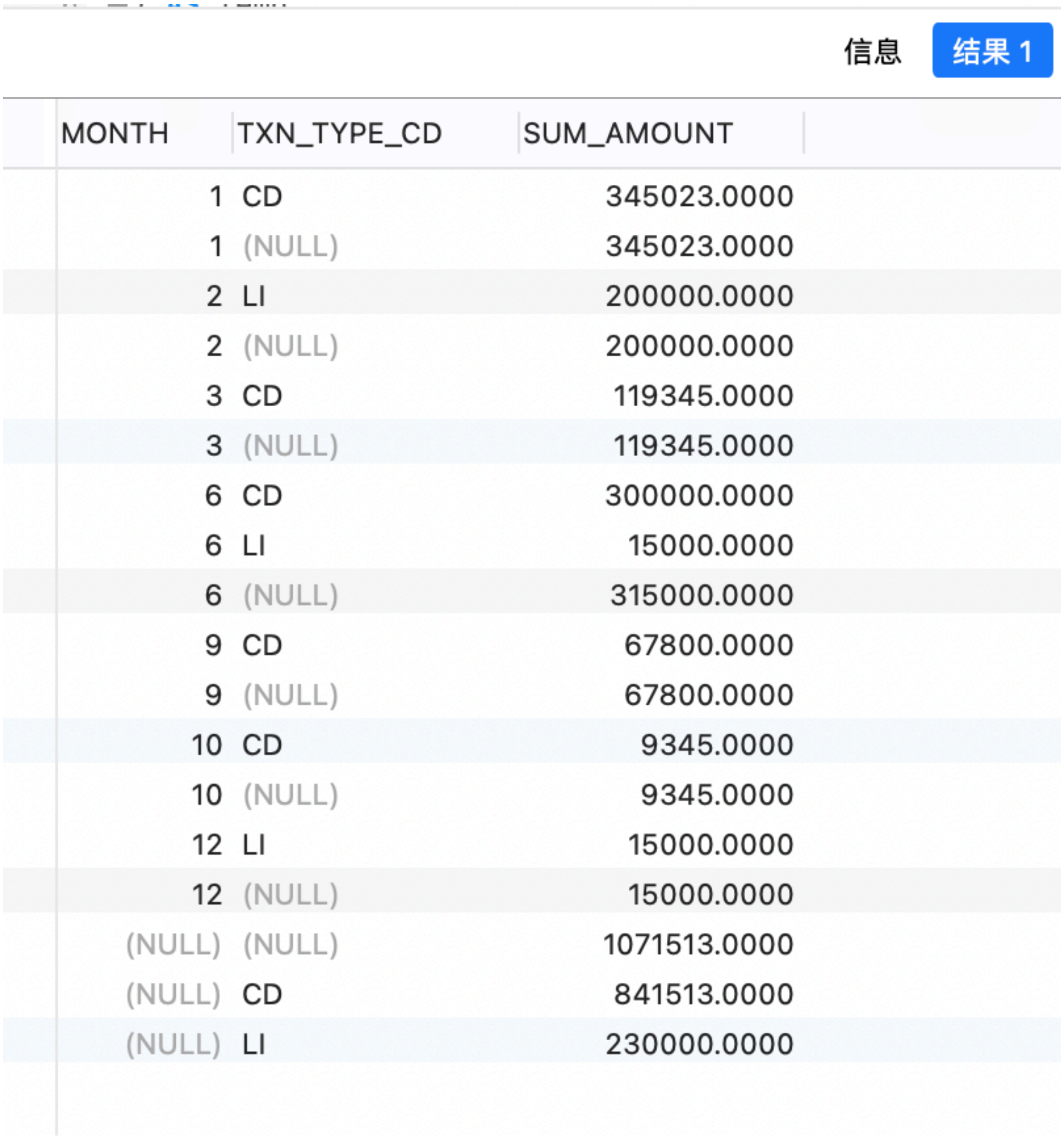
```
SELECT M,sum(S)as Sum,TXN_TYPE_CD
FROM(
  SELECT M,SUM(AMOUNT) as S,TXN_TYPE_CD
    FROM(
      SELECT MONTH(TXN_DATE) as M,TXN_TYPE_CD,AMOUNT
      FROM acc_transaction
      WHERE TXN_DATE LIKE "2015%"
    ) as T
    GROUP BY TXN_TYPE_CD,M
)as T2
GROUP BY M,TXN_TYPE_CD
WITH ROLLUP;
```



Month	Sum	TXN_TYPE_CD
1	345023.0000	CD
1	345023.0000	(NULL)
2	200000.0000	LI
2	200000.0000	(NULL)
3	119345.0000	CD
3	119345.0000	(NULL)
6	300000.0000	CD
6	15000.0000	LI
6	315000.0000	(NULL)
9	67800.0000	CD
9	67800.0000	(NULL)
10	9345.0000	CD
10	9345.0000	(NULL)
12	15000.0000	LI
12	15000.0000	(NULL)
(NULL)	1071513.0000	(NULL)

（10）请对2015年的交易历史进行报表汇总，使用union集合操作实现cube汇总查询。输出属性包括month，TXN_TYPE_CD，sum（销售总额）。

```
(SELECT MONTH, TXN_TYPE_CD, SUM(AMOUNT) AS SUM_AMOUNT
FROM(
    SELECT CAST(DATE_FORMAT(TXN_DATE, "%m") AS SIGNED) AS
MONTH,TXN_TYPE_CD,AMOUNT
    FROM acc_transaction
    WHERE TXN_DATE LIKE "2015%"
) AS temp
GROUP BY MONTH,TXN_TYPE_CD
WITH ROLLUP)
UNION
(SELECT MONTH, TXN_TYPE_CD, SUM(AMOUNT) AS SUM_AMOUNT
FROM(
    SELECT CAST(DATE_FORMAT(TXN_DATE, "%m") AS SIGNED) AS
MONTH,TXN_TYPE_CD,AMOUNT
    FROM acc_transaction
    WHERE TXN_DATE LIKE "2015%"
) AS temp
GROUP BY TXN_TYPE_CD,MONTH
WITH ROLLUP)
```



MONTH	TXN_TYPE_CD	SUM_AMOUNT
1	CD	345023.0000
1	(NULL)	345023.0000
2	LI	200000.0000
2	(NULL)	200000.0000
3	CD	119345.0000
3	(NULL)	119345.0000
6	CD	300000.0000
6	LI	15000.0000
6	(NULL)	315000.0000
9	CD	67800.0000
9	(NULL)	67800.0000
10	CD	9345.0000
10	(NULL)	9345.0000
12	LI	15000.0000
12	(NULL)	15000.0000
(NULL)	(NULL)	1071513.0000
(NULL)	CD	841513.0000
(NULL)	LI	230000.0000