作业三

**(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。**

SELECT `ACCOUNT\_ID`, `AVAIL\_BALANCE`, `OPEN\_DATE`, `LAST\_ACTIVITY\_DATE`

FROM `account`

WHERE `AVAIL\_BALANCE`>=ALL(

SELECT `AVAIL\_BALANCE`

FROM `account`

)

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

SELECT **CONCAT**(e1.LAST\_NAME,e1.FIRST\_NAME)

FROM employee as e1, employee as e2

WHERE e1.SUPERIOR\_EMP\_ID=e2.EMP\_ID AND e1.DEPT\_ID!=e2.DEPT\_ID

**(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 NAME

FROM(

SELECT NAME,**AVG**(AVAIL\_BALANCE) AS AVERAGE

FROM(

SELECT \*

FROM ACCOUNT, BRANCH

WHERE OPEN\_BRANCH\_ID=BRANCH\_ID

) AS temp1

GROUP BY NAME

) AS temp2

WHERE AVERAGE>=ALL(

SELECT AVERAGE

FROM(

SELECT NAME,**AVG**(AVAIL\_BALANCE) AS AVERAGE

FROM(

SELECT \*

FROM ACCOUNT, BRANCH

WHERE OPEN\_BRANCH\_ID=BRANCH\_ID

) AS temp3

GROUP BY NAME

) AS temp4

)

**(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 code

FROM(

(SELECT ID\_NUMBER as code

FROM individual

WHERE ID\_NUMBER LIKE '3%')

UNION(

SELECT CREDIT\_CODE as code

FROM business

WHERE CREDIT\_CODE LIKE '1%'

)

) as temp

ORDER BY code ASC

LIMIT 3;

**(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 **CONCAT**(LAST\_NAME,FIRST\_NAME), TIMESTAMPDIFF(**YEAR**, BIRTH\_DATE, CURDATE()) AS age

FROM (individual NATURAL JOIN customer) JOIN account USING(CUST\_ID)

GROUP BY CUST\_ID

HAVING **COUNT**(\*)>=2

**(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

FROM(

SELECT \*

FROM employee, acc\_transaction

WHERE employee.EMP\_ID=acc\_transaction.TELLER\_EMP\_ID AND TIMESTAMPDIFF(**YEAR**, START\_DATE, CURDATE())>5

)as temp

GROUP BY EMP\_ID

HAVING **COUNT**(\*)>3

ORDER BY START\_DATE ASC;

**(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 distinct CUST\_ID

FROM account AS a

WHERE(

NOT EXISTS (

SELECT PRODUCT\_CD

FROM account AS a3

WHERE a3.CUST\_ID=3 AND a3.PRODUCT\_CD NOT IN(

SELECT PRODUCT\_CD

FROM account AS a4

WHERE a4.CUST\_ID=a.CUST\_ID)

)

)

**(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 distinct CUST\_ID

FROM account AS a

WHERE(

NOT EXISTS (

SELECT PRODUCT\_CD

FROM account AS a1

WHERE a1.CUST\_ID=a.CUST\_ID AND a1.PRODUCT\_CD NOT IN(

SELECT PRODUCT\_CD

FROM account AS a2

WHERE a2.CUST\_ID=3)

)

AND NOT EXISTS (

SELECT PRODUCT\_CD

FROM account AS a3

WHERE a3.CUST\_ID=3 AND a3.PRODUCT\_CD NOT IN(

SELECT PRODUCT\_CD

FROM account AS a4

WHERE a4.CUST\_ID=a.CUST\_ID)

)

)

**(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 **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

**(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)