作业三

Q1

```sql
SELECT ACCOUNT_ID,
       AVAIL_BALANCE,
       OPEN_DATE,
       LAST_ACTIVITY_DATE
FROM `account`
GROUP BY AVAIL_BALANCE,
         ACCOUNT_ID,
         OPEN_DATE,
         LAST_ACTIVITY_DATE
ORDER BY AVAIL_BALANCE desc
LIMIT 1;
```

Q2

```sql
SELECT T.EMP_ID,
       CONCAT(T.LAST_NAME, T.FIRST_NAME) as name
FROM `employee` as T
WHERE DEPT_ID != (
    SELECT DEPT_ID
    FROM `employee` as Q
    WHERE Q.EMP_ID = T.SUPERIOR_EMP_ID
);
```

Q3

```sql
SELECT branch.BRANCH_ID,
       branch.`NAME`,
       X.AVG_BRANCH
FROM
(
  SELECT P.OPEN_BRANCH_ID as BRANCH_ID,
         P.AVG_BRANCH
  FROM
  (
    SELECT AVG_BRANCH
    FROM
    (
      SELECT OPEN_BRANCH_ID,
             SUM(AVAIL_BALANCE)/count(ACCOUNT_ID) as AVG_BRANCH
      FROM `account`
      GROUP BY OPEN_BRANCH_ID
    ) as T
    LIMIT 1
  ) as Q,
  (
    SELECT OPEN_BRANCH_ID,
           SUM(AVAIL_BALANCE)/count(ACCOUNT_ID) as AVG_BRANCH
    FROM `account`
    GROUP BY OPEN_BRANCH_ID
  ) as P
  WHERE P.AVG_BRANCH = Q.AVG_BRANCH
) as X, branch
WHERE X.BRANCH_ID = branch.BRANCH_ID
```

Q4

```sql
SELECT T.`code`
FROM
(
  SELECT ID_NUMBER as `code`,
         CUST_TYPE_CD
  FROM individual NATURAL JOIN customer
  WHERE ID_NUMBER LIKE "3%"
  UNION
  SELECT CREDIT_CODE as `code`,
         CUST_TYPE_CD
  FROM business NATURAL JOIN customer
  WHERE CREDIT_CODE LIKE "1%"
) as T
ORDER BY `code` ASC
LIMIT 3
```

Q5

```
SELECT CONCAT(LAST_NAME, FIRST_NAME) as `name`,
       TIMESTAMPDIFF(YEAR,BIRTH_DATE, DATE_FORMAT(NOW(),'%Y-%m-%d')) as `age`
FROM individual NATURAL JOIN customer
GROUP BY ID_NUMBER
HAVING COUNT(CUST_ID) > 1;
```

Q6

```
SELECT EMP_ID,
       CONCAT(LAST_NAME, FIRST_NAME) as `name`
FROM
(
  SELECT EMP_ID,
         COUNT(TXN_ID) as TXN_NUM
  FROM acc_transaction,
       employee
  WHERE acc_transaction.TELLER_EMP_ID = employee.EMP_ID
  GROUP BY EMP_ID
  HAVING COUNT(TXN_ID) > 3
) as T NATURAL JOIN employee
WHERE CAST(TIMESTAMPDIFF(YEAR,START_DATE, DATE_FORMAT(NOW(),'%Y-%m-%d')) AS DECIMAL(9,2)) > 5
```

Q7

```
SELECT DISTINCT CUST_ID
FROM account
WHERE PRODUCT_CD in (
    SELECT PRODUCT_CD
    FROM account
    WHERE account.CUST_ID = 3
  )
  AND
  CUST_ID != 3
ORDER BY CUST_ID
```

Q8

```sql
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
)
```

Q9

```sql
SELECT MONTH,
       SUM(AMOUNT) AS AMOUNT
FROM
(
  SELECT MONTH,
         TXN_TYPE_CD, SUM(AMOUNT) as 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 T
  GROUP BY MONTH, TXN_TYPE_CD
) as Q
GROUP BY MONTH
WITH ROLLUP
```

Q10

```sql
SELECT MONTH, TXN_TYPE_CD, 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 T
GROUP BY MONTH,
         TXN_TYPE_CD
WITH ROLLUP

UNION

SELECT MONTH, TXN_TYPE_CD, 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 T
GROUP BY TXN_TYPE_CD,
         MONTH
WITH ROLLUP
```