# 主表

## 1.full1（帐号、账单日对应的日期，信用额度，证件号码）

df=spark.sql("""select concat(a.XACCOUNT, '\_',a.bill\_date) as id,a.XACCOUNT, a.bill\_date, a.CRED\_LIMIT,a.CUSTR\_NBR

from standard.acct a

""")

spark.sql("use feature")

df.write.saveAsTable("full1")

# 二、标记（具体根据账单日和还款日之间间隔的天数或者可以办理账单分期的具体业务规定时间来定）

df=spark.sql("""select concat(a.XACCOUNT, '\_',a.bill\_date)) as id, 1 as label

from feature.full1 a inner join (

select \*

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) m

on a.XACCOUNT =m.XACCOUNT

where

unix\_timestamp(m.PURCH\_DAY, "yyyy-MM-dd HH:mm:ss") >= unix\_timestamp(a.bill\_date, "yyyy-MM-dd HH:mm:ss")

and

unix\_timestamp(m.PURCH\_DAY, "yyyy-MM-dd HH:mm:ss")< unix\_timestamp(a.bill\_date, "yyyy-MM-dd HH:mm:ss")+20\*24\*60\*60

group by a.XACCOUNT, a.bill\_date """)

spark.sql("use feature ")

df.write.saveAsTable("label")

# 三、基础特征

## 1. custr

df=spark.sql("""select a.id as id, c.CLASS\_CODE,c.DAY\_BIRTH,c.DEPENDENTS,c.EDUCA,c.GENDER,c.HOME\_CODE,c.INCOME\_AN2,c.INCOME\_ANN,c.INCOME\_SR2,c.INCOME\_SRC,c.MAR\_STATUS,c.OCC\_CATGRY,c.RACE\_CODE,c.YR\_FORECOM,c.YR\_IN\_COM2,c.YR\_IN\_COMP,c.YR\_THERE,c.HOME\_LOAN,c.INT\_TAXCOD,cast(substring(a.bill\_date,1,4) as int) - cast(substring(c.DAY\_BIRTH,1,4) as int) as age, (case when cast(c.DAY\_BIRTH as string) like '200%' then 1 when cast(c.DAY\_BIRTH as string) like '199%' then 2 when cast(c.DAY\_BIRTH as string) like '198%' then 3 when (cast(c.DAY\_BIRTH as string) like '196%' or cast(c.DAY\_BIRTH as string) like '196%') then 4 else 5 end ) as age\_bins

from feature.full1 a left join standard.custr c

on a.custr\_nbr=c.custr\_nbr and a.bill\_date = c.bill\_date """)

spark.sql("use feature")

df.write.saveAsTable("custr")

## 2. acct

df=spark.sql("""select a.id as id, CONSUM\_AMT, CASH\_FLG,CASH\_FEES,CASH\_TOTAL,ODUE\_AMT,CLOSE\_CODE,NBR\_PAYMNT,NBR\_PURCH,NBR\_TRANS,LAST\_BILL\_CUM,STM\_BALORI,LAST\_BILL\_CASH,STM\_BALNCE,STM\_BALFLAG,MTHS\_ODUE,MP\_REM\_PPL,STM\_BALMP,STM\_BMFLAG,INT\_NOTION,INT\_CUNOT,INT\_CMPOND,case when b.BANKACCT1 = ' ' then 0 else 1 end as is\_glhkzh, cast(substring(a.bill\_date,1,4) as int) - cast(substring(b.DAY\_OPENED,1,4) as int) as khzl, CATEGORY,MTHS\_ODUE,ODUE\_FLAG,CYCLE\_NBR

from feature.full1 a left join standard.acct b

on a.card\_nbr =b.card\_nbr and a.bill\_date =b.bill\_date """)

spark.sql("use feature ")

df.write.saveAsTable("acct")

## 3. card

df=spark.sql("""select a.id as id, CARD\_STAT,EXPIRY\_DTE

from feature.full1 a left join standard.card b

on a.card\_nbr =b.card\_nbr and a.bill\_date =b.bill\_date """)

spark.sql("use feature")

df.write.saveAsTable("card")

## 4. stmt(根据实际情况多确认)

select a.id as id,b.\*

from feature.full1 a left join(select \* from (selectBAL\_CMPINT,BAL\_FREE,BAL\_INT,BALINTFLAG,BAL\_NOINT,BAL\_ORINT,CASH\_ADFEE,CASH\_ADVCE,CLOSE\_BAL,CLSBAL\_FLAG,INT\_CHDCMP,INT\_CHGD,MIN\_DUE,NBR\_CASHAD,NBR\_FEEDTY,NBR\_OTHERS,ODUE\_FLAG,ODUE\_HELD,PAYMENT,PURCHASES,

row\_number() over (partition by XACCOUNT,bill\_date order by MONTH\_NBR desc) as number

from standard.stmt) where number = 1) b

on a.xaccount = b.xaccount and a.bill\_date = b.bill\_date

# 四、衍生特征（里面涉及到的转换不同银行可能不同）

## 1. 过去X个月的最大消费额

### (1)one\_max\_amt 过去一个月的最大消费额

df=spark.sql("""

select f.id as id,max(e.BILL\_AMT) /100 as one\_max\_amt from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where REV\_IND =" " and BILL\_AMTFLAG="+" and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("one\_max\_amt")

### (2)two\_max\_amt过去2个月的最大消费额

df=spark.sql("""

select f.id as id,max(e.BILL\_AMT/100) as two\_max\_amt from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where REV\_IND =" " and BILL\_AMTFLAG="+" and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-2)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_max\_amt")

### (3)three\_max\_amt过去3个月的最大消费额

df=spark.sql("""

select f.id as id,max(e.BILL\_AMT/100) as three\_max\_amt from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where REV\_IND =" " and BILL\_AMTFLAG="+" and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-3)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_max\_amt")

### (4)six\_max\_amt过去6个月的最大消费额

df=spark.sql("""

select f.id as id,max(e.BILL\_AMT/100) as six\_max\_amt from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where REV\_IND =" " and BILL\_AMTFLAG="+" and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-6)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_max\_amt")

## 2、过去X个月的消费次数

### (1)one\_buy\_count过去1个月的消费次数

df=spark.sql("""

select f.id as id,count(\*) as one\_buy\_count from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where REV\_IND =" " and BILL\_AMTFLAG="+" and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("one\_buy\_count")

### (2)two\_buy\_count过去2个月的消费次数

df=spark.sql("""

select f.id as id,count(\*) as two\_buy\_count from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where REV\_IND =" " and BILL\_AMTFLAG="+" and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-2)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_buy\_count")

### (3)three\_buy\_count过去3个月的消费次数

df=spark.sql("""

select f.id as id,count(\*) as three\_buy\_count from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where REV\_IND =" " and BILL\_AMTFLAG="+" and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-3)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_buy\_count")

### (4)six\_buy\_count过去6个月的消费次数

df=spark.sql("""

select f.id as id,count(\*) as six\_buy\_count from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where REV\_IND =" " and BILL\_AMTFLAG="+" and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-6)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_buy\_count")

## 3、过去X个月的最大消费金额占授信额度的比例

### (1)one\_max\_rate过去1个月的最大消费金额占授信额度的比例

df=spark.sql("""

select f.id as id, round((a.one\_max\_amt/f.cred\_limit),4) as one\_max\_rate

from feature.full1 f inner join one\_max\_amt as a

on f.id=a.id

""")

spark.sql("use feature")

df.write.saveAsTable("one\_max\_rate")

### (2)two\_max\_rate过去2个月的最大消费金额占授信额度的比例

df=spark.sql("""

select f.id as id, round((a.two\_max\_amt/f.cred\_limit),4) as two\_max\_rate

from feature.full1 f inner join two\_max\_amt as a

on f.id=a.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_max\_rate")

### (3)three\_max\_rate过去3个月的最大消费金额占授信额度的比例

df=spark.sql("""

select f.id as id, round((a.three\_max\_amt/f.cred\_limit),4) as three\_max\_rate

from feature.full1 f inner join three\_max\_amt as a

on f.id=a.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_max\_rate")

### (4)six\_max\_rate过去6个月的最大消费金额占授信额度的比例

df=spark.sql("""

select f.id as id, round((a.six\_max\_amt/f.cred\_limit),4) as six\_max\_rate

from feature.full1 f inner join six\_max\_amt as a

on f.id=a.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_max\_rate")

## 4、过去X个月贷记卡存入金额

### (1)one\_sum\_c\_amt过去1个月贷记卡存入金额

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as one\_sum\_c\_amt from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where (REV\_IND =" " or REV\_IND =0) and BILL\_AMTFLAG="-" and trans\_type!="1050" and BRNO != "999999"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("one\_sum\_c\_amt")

### (2)two\_sum\_c\_amt过去2个月贷记卡存入金额

df=spark.sql("""

select f.id as id,(sum(e.BILL\_AMT))/100 as two\_sum\_c\_amt from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where (REV\_IND =" " or REV\_IND =0) and BILL\_AMTFLAG="-" and trans\_type!="1050" and BRNO != "999999"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-2)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_sum\_c\_amt")

### (3)three\_sum\_c\_amt过去3个月贷记卡存入金额

df=spark.sql("""

select f.id as id,(sum(e.BILL\_AMT))/100 as three\_sum\_c\_amt from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where (REV\_IND =" " or REV\_IND =0) and BILL\_AMTFLAG="-" and trans\_type!="1050" and BRNO != "999999"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-3)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_sum\_c\_amt")

### (4)six\_sum\_c\_amt过去6个月贷记卡存入金额

df=spark.sql("""

select f.id as id,(sum(e.BILL\_AMT))/100 as six\_sum\_c\_amt from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where (REV\_IND =" " or REV\_IND =0) and BILL\_AMTFLAG="-" and trans\_type!="1050" and BRNO != "999999"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-6)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_sum\_c\_amt")

## 5、过去X个月消费总额

### (1)one\_sum\_amt过去1个月消费总额

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as one\_sum\_amt from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where REV\_IND =" " and BILL\_AMTFLAG="+" and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("one\_sum\_amt")

### (2)two\_sum\_amt过去2个月消费总额

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as two\_sum\_amt from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where REV\_IND =" " and BILL\_AMTFLAG="+" and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-2)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_sum\_amt")

### (3)three\_sum\_amt过去3个月消费总额

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as three\_sum\_amt from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where REV\_IND =" " and BILL\_AMTFLAG="+" and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-3)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_sum\_amt")

### (4)six\_sum\_amt过去6个月消费总额

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as six\_sum\_amt from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where REV\_IND =" " and BILL\_AMTFLAG="+" and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-6)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_sum\_amt")

## 6、过去X个月消费金额占授信额度的比例

### (1)one\_sum\_rate过去1个月消费金额占授信额度的比例

df=spark.sql("""

select f.id as id, round((a.one\_sum\_amt/f.cred\_limit),4) as one\_sum\_rate

from feature.full1 f inner join one\_sum\_amt as a

on f.id=a.id

""")

spark.sql("use feature")

df.write.saveAsTable("one\_sum\_rate")

### (2)two\_sum\_rate过去2个月消费金额占授信额度的比例

df=spark.sql("""

select f.id as id, round((a.two\_sum\_amt/f.cred\_limit),4) as two\_sum\_rate

from feature.full1 f inner join two\_sum\_amt as a

on f.id=a.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_sum\_rate")

### (3)three\_sum\_rate过去3个月消费金额占授信额度的比例

df=spark.sql("""

select f.id as id, round((a.three\_sum\_amt/f.cred\_limit),4) as three\_sum\_rate

from feature.full1 f inner join three\_sum\_amt as a

on f.id=a.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_sum\_rate")

### (4)six\_sum\_rate过去6个月消费金额占授信额度的比例

df=spark.sql("""

select f.id as id, round((a.six\_sum\_amt/f.cred\_limit),4) as six\_sum\_rate

from feature.full1 f inner join six\_sum\_amt as a

on f.id=a.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_sum\_rate")

## 7、过去X个月消费金额与还款金额的差

### (1)one\_amt\_cha过去一个月消费金额与还款金额的差

df=spark.sql("""

select sa.id as id, round((one\_sum\_amt-one\_sum\_c\_amt),2) as one\_amt\_cha

from feature.one\_sum\_amt sa inner join feature.one\_sum\_c\_amt sca

on sa.id=sca.id

""")

spark.sql("use feature")

df.write.saveAsTable("one\_amt\_cha")

### (2)two\_amt\_cha过去两个月消费金额与还款金额的差

df=spark.sql("""

select sa.id as id, round((two\_sum\_amt-two\_sum\_c\_amt),2) as two\_amt\_cha

from feature.two\_sum\_amt sa inner join feature.two\_sum\_c\_amt sca

on sa.id=sca.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_amt\_cha")

### (3)three\_amt\_cha过去三个月消费金额与还款金额的差

df=spark.sql("""

select sa.id as id, round((three\_sum\_amt-three\_sum\_c\_amt),2) as three\_amt\_cha

from feature.three\_sum\_amt sa inner join feature.three\_sum\_c\_amt sca

on sa.id=sca.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_amt\_cha")

### (4)six\_amt\_cha过去六个月消费金额与还款金额的差

df=spark.sql("""

select sa.id as id, round((six\_sum\_amt-six\_sum\_c\_amt),2) as six\_amt\_cha

from feature.six\_sum\_amt sa inner join feature.six\_sum\_c\_amt sca

on sa.id=sca.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_amt\_cha")

## 8、第X个月单月交易次数

### (1)twom\_count 第二个月单月交易次数

df=spark.sql("""

select bc2.id as id, two\_buy\_count -one\_buy\_count as twom\_count

from feature.two\_buy\_count bc2 inner join feature.one\_buy\_count bc1

on bc2.id= bc1.id

""")

spark.sql("use feature")

df.write.saveAsTable("twom\_count")

### (2)threem\_count 第三个月单月交易次数

df=spark.sql("""

select bc2.id as id, three\_buy\_count -two\_buy\_count as threem\_count

from feature.three\_buy\_count bc2 inner join feature.two\_buy\_count bc1

on bc2.id= bc1.id

""")

spark.sql("use feature")

df.write.saveAsTable("threem\_count")

## 9、过去X个月总消费的均值

### (1)two\_sum\_amt\_mean过去2个月总消费的均值

df=spark.sql("""

select sa2.id as id, round((sa2.two\_sum\_amt/2),2) as two\_sum\_amt\_mean

from feature.two\_sum\_amt sa2

""")

spark.sql("use feature")

df.write.saveAsTable("two\_sum\_amt\_mean")

### (2)three\_sum\_amt\_mean过去3个月总消费的均值

df=spark.sql("""

select sa2.id as id, round((sa2.three\_sum\_amt/3),2) as three\_sum\_amt\_mean

from feature.three\_sum\_amt sa2

""")

spark.sql("use feature")

df.write.saveAsTable("three\_sum\_amt\_mean")

### (3)six\_sum\_amt\_mean过去6个月总消费的均值

df=spark.sql("""

select sa2.id as id, round((sa2.six\_sum\_amt/6),2) as six\_sum\_amt\_mean

from feature.six\_sum\_amt sa2

""")

spark.sql("use feature")

df.write.saveAsTable("six\_sum\_amt\_mean")

## 10、过去X个月的平均交易次数

### (1)twom\_count\_mean过去两个月的平均交易次数

df=spark.sql("""

select b.id, ceiling(b.two\_buy\_count/2) as twom\_count\_mean

from feature.two\_buy\_count as b

""")

spark.sql("use feature")

df.write.saveAsTable("twom\_count\_mean")

### (2)threem\_count\_mean过去3个月的平均交易次数

df=spark.sql("""

select bc2.id as id, ceiling(bc2.three\_buy\_count /3) as threem\_count\_mean

from feature.three\_buy\_count bc2

""")

spark.sql("use feature")

df.write.saveAsTable("threem\_count\_mean")

### (3)6m\_count\_mean过去6个月的平均交易次数

df=spark.sql("""

select bc2.id as id, ceiling (bc2.six\_buy\_count /6) as 6m\_count\_mean

from feature.six\_buy\_count bc2

""")

spark.sql("use feature")

df.write.saveAsTable("6m\_count\_mean")

## 11、过去X个月与过去X个月平均交易次数的差

### (1)one\_to\_two\_count\_mean\_cha过去一个月与过去两个月平均交易次数的差

df=spark.sql("""

select a.id as id, a.one\_buy\_count - b.twom\_count\_mean as one\_to\_two\_count\_mean\_cha

from one\_buy\_count as a, twom\_count\_mean as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("one\_to\_two\_count\_mean\_cha")

### (2)one\_to\_three\_count\_mean\_cha过去一个月与过去三个月平均交易次数的差

df=spark.sql("""

select a.id as id, a.one\_buy\_count - b.threem\_count\_mean as one\_to\_three\_count\_mean\_cha

from one\_buy\_count as a, threem\_count\_mean as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("one\_to\_three\_count\_mean\_cha")

## 12、第X个月与第X个月交易次数的差

### (1)two\_to\_one\_count\_cha第二个月与第一个月交易次数的差

df=spark.sql("""

select a.id as id, a.twom\_count - b.one\_buy\_count as two\_to\_one\_count\_cha

from twom\_count as a, one\_buy\_count as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_to\_one\_count\_cha ")

### (2)three\_to\_two\_count\_cha第三个月与第二个月交易次数的差

df=spark.sql("""

select a.id as id, a.threem\_count - b.twom\_count as three\_to\_two\_count\_cha

from threem\_count as a, twom\_count as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_to\_two\_count\_cha ")

## 13、第X个月至第X个月交易次数的变化率

### (1)two\_to\_one\_count\_rate第二个月至第一个月交易次数的变化率

df=spark.sql("""

select a.id as id, round((a.two\_to\_one\_count\_cha/b.one\_buy\_count),4) as two\_to\_one\_count\_rate

from two\_to\_one\_count\_cha as a, one\_buy\_count as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_to\_one\_count\_rate ")

### (2)three\_to\_two\_count\_rate第三个月至第二个月交易次数的变化率

df=spark.sql("""

select a.id as id, round((a.three\_to\_two\_count\_cha/b.twom\_count),4) as three\_to\_two\_count\_rate

from three\_to\_two\_count\_cha as a, twom\_count as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_to\_two\_count\_rate ")

## 14、过去X个月的平均消费额与过去X个月平均消费额的差

### (1)two\_to\_one\_amt\_cha\_mean过去2个月的平均消费额与过去1个月平均消费额的差

df=spark.sql("""

select a.id as id, round((a.two\_sum\_amt\_mean - b.one\_sum\_amt),2) as two\_to\_one\_amt\_cha\_mean

from two\_sum\_amt\_mean as a,one\_sum\_amt as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_to\_one\_amt\_cha\_mean")

### (2)three\_to\_two\_amt\_cha\_mean过去3个月的平均消费额与过去2个月平均消费额的差

df=spark.sql("""

select a.id as id, round((a.three\_sum\_amt\_mean - b.two\_sum\_amt\_mean),2) as three\_to\_two\_amt\_cha\_mean

from three\_sum\_amt\_mean as a,two\_sum\_amt\_mean as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_to\_two\_amt\_cha\_mean")

### (3)six\_to\_three\_amt\_cha\_mean过去6个月的平均消费额与过去3个月平均消费额的差

df=spark.sql("""

select a.id as id, round((a.six\_sum\_amt\_mean - b.three\_sum\_amt\_mean),2) as six\_to\_three\_amt\_cha\_mean

from six\_sum\_amt\_mean as a,three\_sum\_amt\_mean as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_to\_three\_amt\_cha\_mean")

## 15、第X个月的最大消费额

### (1)twom\_max\_amt第二个月的最大消费额

df=spark.sql("""

select f.id as id, max(e.BILL\_AMT)/100 as twom\_max\_amt from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where REV\_IND =" " and BILL\_AMTFLAG="+" and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-2)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("twom\_max\_amt")

### (2)threem\_max\_amt第三个月的最大消费额

df=spark.sql("""

select f.id as id,max(e.BILL\_AMT)/100 as threem\_max\_amt from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where REV\_IND =" " and BILL\_AMTFLAG="+" and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-2)

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-3)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("threem\_max\_amt")

## 16、第X个月与第X个月最大消费额的差

### (1)twom\_to\_one\_max\_cha第二个月与第一个月最大消费额的差

df=spark.sql("""

select a.id as id, round((a.twom\_max\_amt - b.one\_max\_amt),2) as twom\_to\_one\_max\_cha

from twom\_max\_amt as a,one\_max\_amt as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("twom\_to\_one\_max\_cha")

### (2)threem\_to\_twom\_max\_cha第三个月与第二个月最大消费额的差

df=spark.sql("""

select a.id as id, round((a.threem\_max\_amt - b.twom\_max\_amt),2) as threem\_to\_twom\_max\_cha

from threem\_max\_amt as a,twom\_max\_amt as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("threem\_to\_twom\_max\_cha")

## 17、第X个月至第X个月最大消费额的变化比例

### (1)twom\_to\_one\_max\_rate第二个月至第一个月最大消费额的变化比例

df=spark.sql("""

select a.id as id, round((a.twom\_to\_one\_max\_cha/b.one\_max\_amt),4) as twom\_to\_one\_max\_rate

from twom\_to\_one\_max\_cha as a,one\_max\_amt as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("twom\_to\_one\_max\_rate")

### (2)threem\_to\_twom\_max\_rate第三个月至第二个月最大消费额的变化比例

df=spark.sql("""

select a.id as id, round((a.threem\_to\_twom\_max\_cha/b.twom\_max\_amt),4) as threem\_to\_twom\_max\_rate

from threem\_to\_twom\_max\_cha as a,twom\_max\_amt as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("threem\_to\_twom\_max\_rate")

## 18、过去第X个月的消费总额

### (1)twom\_amt过去第二个月的消费总额

df=spark.sql("""

select a.id as id, round((a.two\_sum\_amt - b.one\_sum\_amt),2) as twom\_amt

from two\_sum\_amt as a,one\_sum\_amt as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("twom\_amt")

### (2)threem\_amt过去第三个月的消费总额

df=spark.sql("""

select a.id as id, round((a.three\_sum\_amt - b.two\_sum\_amt),2) as threem\_amt

from three\_sum\_amt as a,two\_sum\_amt as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("threem\_amt")

## 19、第X个月与第X个月的消费额差

### (1)twom\_to\_one\_amt\_cha第二个月与第一个月的消费额差

df=spark.sql("""

select a.id as id, round((a.twom\_amt - b.one\_sum\_amt),2) as twom\_to\_one\_amt\_cha

from twom\_amt as a,one\_sum\_amt as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("twom\_to\_one\_amt\_cha")

### (2)threem\_to\_twom\_amt\_cha第三个月与第二个月的消费额差

df=spark.sql("""

select a.id as id, round((a.threem\_amt - b.twom\_amt),2) as threem\_to\_twom\_amt\_cha

from threem\_amt as a,twom\_amt as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("threem\_to\_twom\_amt\_cha")

## 20、第X个月与第X个月消费额的变化率

### (1)twom\_to\_one\_amt\_rate第二个月与第一个月消费额的变化率（第二个月与第一月的差/第一个月的消费额）

df=spark.sql("""

select a.id as id, round((a.twom\_to\_one\_amt\_cha/ b.one\_sum\_amt),4) as twom\_to\_one\_amt\_rate

from twom\_to\_one\_amt\_cha as a, one\_sum\_amt as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("twom\_to\_one\_amt\_rate")

### (2)threem\_to\_twom\_amt\_rate第三个月与第二个月消费额的变化率

df=spark.sql("""

select a.id as id, round((a.threem\_to\_twom\_amt\_cha/ b.twom\_amt),4) as threem\_to\_twom\_amt\_rate

from threem\_to\_twom\_amt\_cha as a, twom\_amt as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("threem\_to\_twom\_amt\_rate")

## 21、MCC 过去一个月次数统计

### （1）A

df=spark.sql("""

select f.id as id,count(\*) as A\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5812,5811, 7011,7012,5813,7032,7033,7829,7911,7922,7929,7932,7933,7941,7992,7994,7996,7997,7998,7999,7297,7298,5094,5950,5944,7631,5970,5937,5932,5971) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("A\_a")

### （2）A\_a

df=spark.sql("""

select f.id as id,count(\*) as A\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5812,5811) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("A\_a")

### （3）A\_b

df=spark.sql("""

select f.id as id,count(\*) as A\_b from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (7011,7012,5813) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("A\_b")

### （4）A\_c

df=spark.sql("""

select f.id as id,count(\*) as A\_c from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (7032,7033,7829,7911,7922,7929,7932,7933,7941,7992,7994,7996,7997,7998,7999,7297,7298) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("A\_c")

### （5）A\_d

df=spark.sql("""

select f.id as id,count(\*) as A\_d from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5094,5950,5944,7631) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("A\_d")

### （6）A\_e

df=spark.sql("""

select f.id as id,count(\*) as A\_e from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5970,5937,5932,5971) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("A\_e")

### （7）B

df=spark.sql("""

select f.id as id,count(\*) as B\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (1520,7013,5933, 5271,5511,5521,5551,5561,5571,5592,5598,5599) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("B\_a")

### （8）B\_a

df=spark.sql("""

select f.id as id,count(\*) as B\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (1520,7013,5933) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("B\_a")

### （9）B\_b

df=spark.sql("""

select f.id as id,count(\*) as B\_b from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5271,5511,5521,5551,5561,5571,5592,5598,5599) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("B\_b")

### （10）C

df=spark.sql("""

select f.id as id,count(\*) as C\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5300,5722,5994, 5200,5411,4900,6300,5960,4814,4899,7523,5541,5542,4112,4511,4111,4121,4131,4784,3998,8651,9211,9222,9223,9311,9399,9400,9402) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("C\_a")

### （11）C\_a

df=spark.sql("""

select f.id as id,count(\*) as C\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5300,5722,5994) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("C\_a")

### （12）C\_b

df=spark.sql("""

select f.id as id,count(\*) as C\_b from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5200,5411) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("C\_b")

### （13）C\_c

df=spark.sql("""

select f.id as id,count(\*) as C\_c from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (4900,6300,5960,4814,4899,7523) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("C\_c")

### （14）C\_d

df=spark.sql("""

select f.id as id,count(\*) as C\_d from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5541,5542) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("C\_d")

### （15）C\_e

df=spark.sql("""

select f.id as id,count(\*) as C\_e from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (4112,4511,4111,4121,4131,4784,3998) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("C\_e")

### （16）C\_f

df=spark.sql("""

select f.id as id,count(\*) as C\_f from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (8651,9211,9222,9223,9311,9399,9400,9402) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("C\_f")

### （17）D

df=spark.sql("""

select f.id as id,count(\*) as D\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (8062, 8211,8220,8398) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("D\_a")

### （18）D\_a

df=spark.sql("""

select f.id as id,count(\*) as D\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (8062) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("D\_a")

### （19）D\_b

df=spark.sql("""

select f.id as id,count(\*) as D\_b from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (8211,8220) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("D\_b")

### （20）D\_c

df=spark.sql("""

select f.id as id,count(\*) as D\_c from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (8398) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("D\_c")

### （21）E

df=spark.sql("""

select f.id as id,count(\*) as E\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5532,5533,5814,5921,5931,5964,5965,5983,7538,7832,7841,7991,7993,7995,5211,5231,5251,5261,5309,5310,5311,5331,5399,5422,5441,5451,5462,5499,5611,5621,5631,5641,5651,5655,5661,5681,5691,5697,5698,5699,5712,5713,5714,5718,5719,5732,5733,5734,5735,5940,5941,5942,5943,5945,5946,5947,5948,5949,5973,5975,5977,5978,5992,5993,5995,5999, 8241,8244,8249,8299,8351,763,4011,4225,4457,4812,5963,5966,5967,5969,6513,7535,7542,7549,7622,7623,7629,7641,7692,7699,8050,8099,8111,8911,8912,8931,8999,742,780,4214,4215,4411,4468,4582,4789,4816,4821,5935,5962,5968,5996,5997,7210,7211,7216,7217,7221,7230,7251,7261,7273,7276,7277,7278,7295,7296,7299,7311,7333,7338,7339,7342,7349,7361,7372,7375,7379,7392,7393,7394,7395,7399,7512,7513,7519,7531,7534,5972,4722,4733,7321,8641,8661,8675,8699,8071,8011,8021,8031,8041,8042,8043,8049,5912,5976,4119,6010,6011,6012,6051,6211) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("E\_a")

### （22）E\_a

df=spark.sql("""

select f.id as id,count(\*) as E\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5532,5533,5814,5921,5931,5964,5965,5983,7538,7832,7841,7991,7993,7995,5211,5231,5251,5261,5309,5310,5311,5331,5399,5422,5441,5451,5462,5499,5611,5621,5631,5641,5651,5655,5661,5681,5691,5697,5698,5699,5712,5713,5714,5718,5719,5732,5733,5734,5735,5940,5941,5942,5943,5945,5946,5947,5948,5949,5973,5975,5977,5978,5992,5993,5995,5999) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("E\_a")

### （23）E\_b

df=spark.sql("""

select f.id as id,count(\*) as E\_b from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (8241,8244,8249,8299,8351) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("E\_b")

### （24）E\_c

df=spark.sql("""

select f.id as id,count(\*) as E\_c from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (763,4011,4225,4457,4812,5963,5966,5967,5969,6513,7535,7542,7549,7622,7623,7629,7641,7692,7699,8050,8099,8111,8911,8912,8931,8999,742,780,4214,4215,4411,4468,4582,4789,4816,4821,5935,5962,5968,5996,5997,7210,7211,7216,7217,7221,7230,7251,7261,7273,7276,7277,7278,7295,7296,7299,7311,7333,7338,7339,7342,7349,7361,7372,7375,7379,7392,7393,7394,7395,7399,7512,7513,7519,7531,7534) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("E\_c")

### （25）E\_d

df=spark.sql("""

select f.id as id,count(\*) as E\_d from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5972,4722,4733) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("E\_d")

### （26）E\_e

df=spark.sql("""

select f.id as id,count(\*) as E\_e from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (7321,8641,8661,8675,8699) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("E\_e")

### （27）E\_f

df=spark.sql("""

select f.id as id,count(\*) as E\_f from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (8071,8011,8021,8031,8041,8042,8043,8049,5912,5976,4119) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("E\_f")

### （28）E\_g

df=spark.sql("""

select f.id as id,count(\*) as E\_g from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (6010,6011,6012,6051,6211) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("E\_g")

### （29）F\_a

df=spark.sql("""

select f.id as id,count(\*) as F\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5013,5021,5039,5044,5045,5046,5047,5051,5065,5072,5074,5111,5122,5131,5137,5139,5172,5192,5193,5198,5998,5398,4458) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("F\_a")

### （30）H\_a

df=spark.sql("""

select f.id as id,count(\*) as H\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (9498) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("H\_a")

### （31）Z\_a

df=spark.sql("""

select f.id as id,count(\*) as Z\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD =0000 and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("Z\_a")

## 22、MCC 过去一个月金额统计

### （1）class\_A

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_A\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5812,5811, 7011,7012,5813,7032,7033,7829,7911,7922,7929,7932,7933,7941,7992,7994,7996,7997,7998,7999,7297,7298,5094,5950,5944,7631,5970,5937,5932,5971) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_A\_a ")

### （2）class\_A\_a

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_A\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5812,5811) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_A\_a ")

### （3）class\_A\_b

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_A\_b from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (7011,7012,5813) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_A\_b")

### （4）class\_A\_c

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_A\_c from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (7032,7033,7829,7911,7922,7929,7932,7933,7941,7992,7994,7996,7997,7998,7999,7297,7298) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_A\_c")

### （5）class\_A\_d

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_A\_d from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5094,5950,5944,7631) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_A\_d")

### （6）class\_A\_e

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_A\_e from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5970,5937,5932,5971) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_A\_e")

### （7）class\_B

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_B\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (1520,7013,5933, 5271,5511,5521,5551,5561,5571,5592,5598,5599) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_B\_a")

### （8）class\_B\_a

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_B\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (1520,7013,5933) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_B\_a")

### （9）class\_B\_b

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_B\_b from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5271,5511,5521,5551,5561,5571,5592,5598,5599) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_B\_b")

### （10）class\_C

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_C\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5300,5722,5994, 5200,5411,4900,6300,5960,4814,4899,7523,5541,5542,4112,4511,4111,4121,4131,4784,3998,8651,9211,9222,9223,9311,9399,9400,9402) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_C\_a")

### （11）class\_C\_a

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_C\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5300,5722,5994) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_C\_a")

### （12）class\_C\_b

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_C\_b from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5200,5411) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_C\_b")

### （13）class\_C\_c

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_C\_c from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (4900,6300,5960,4814,4899,7523) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_C\_c")

### （14）class\_C\_d

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_C\_d from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5541,5542) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_C\_d")

### （15）class\_C\_e

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_C\_e from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (4112,4511,4111,4121,4131,4784,3998) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_C\_e")

### （16）class\_C\_f

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_C\_f from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (8651,9211,9222,9223,9311,9399,9400,9402) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_C\_f")

### （17）class\_D

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_D\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (8062, 8211,8220,8398) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_D\_a")

### （18）class\_D\_a

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_D\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (8062) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_D\_a")

### （19）class\_D\_b

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_D\_b from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (8211,8220) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_D\_b")

### （20）class\_D\_c

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_D\_c from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (8398) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_D\_c")

### （21）class\_E

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_E\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5532,5533,5814,5921,5931,5964,5965,5983,7538,7832,7841,7991,7993,7995,5211,5231,5251,5261,5309,5310,5311,5331,5399,5422,5441,5451,5462,5499,5611,5621,5631,5641,5651,5655,5661,5681,5691,5697,5698,5699,5712,5713,5714,5718,5719,5732,5733,5734,5735,5940,5941,5942,5943,5945,5946,5947,5948,5949,5973,5975,5977,5978,5992,5993,5995,5999, 8241,8244,8249,8299,8351,763,4011,4225,4457,4812,5963,5966,5967,5969,6513,7535,7542,7549,7622,7623,7629,7641,7692,7699,8050,8099,8111,8911,8912,8931,8999,742,780,4214,4215,4411,4468,4582,4789,4816,4821,5935,5962,5968,5996,5997,7210,7211,7216,7217,7221,7230,7251,7261,7273,7276,7277,7278,7295,7296,7299,7311,7333,7338,7339,7342,7349,7361,7372,7375,7379,7392,7393,7394,7395,7399,7512,7513,7519,7531,7534,5972,4722,4733,7321,8641,8661,8675,8699,8071,8011,8021,8031,8041,8042,8043,8049,5912,5976,4119,6010,6011,6012,6051,6211) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_E\_a")

### （22）class\_E\_a

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_E\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5532,5533,5814,5921,5931,5964,5965,5983,7538,7832,7841,7991,7993,7995,5211,5231,5251,5261,5309,5310,5311,5331,5399,5422,5441,5451,5462,5499,5611,5621,5631,5641,5651,5655,5661,5681,5691,5697,5698,5699,5712,5713,5714,5718,5719,5732,5733,5734,5735,5940,5941,5942,5943,5945,5946,5947,5948,5949,5973,5975,5977,5978,5992,5993,5995,5999) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_E\_a")

### （23）class\_E\_b

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_E\_b from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (8241,8244,8249,8299,8351) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_E\_b")

### （24）class\_E\_c

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_E\_c from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (763,4011,4225,4457,4812,5963,5966,5967,5969,6513,7535,7542,7549,7622,7623,7629,7641,7692,7699,8050,8099,8111,8911,8912,8931,8999,742,780,4214,4215,4411,4468,4582,4789,4816,4821,5935,5962,5968,5996,5997,7210,7211,7216,7217,7221,7230,7251,7261,7273,7276,7277,7278,7295,7296,7299,7311,7333,7338,7339,7342,7349,7361,7372,7375,7379,7392,7393,7394,7395,7399,7512,7513,7519,7531,7534) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_E\_c")

### （25）class\_E\_d

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_E\_d from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5972,4722,4733) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_E\_d")

### （26）class\_E\_e

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_E\_e from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (7321,8641,8661,8675,8699) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_E\_e")

### （27）class\_E\_f

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_E\_f from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (8071,8011,8021,8031,8041,8042,8043,8049,5912,5976,4119) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_E\_f")

### （28）class\_E\_g

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_E\_g from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (6010,6011,6012,6051,6211) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_E\_g")

### （29）class\_F\_a

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_F\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (5013,5021,5039,5044,5045,5046,5047,5051,5065,5072,5074,5111,5122,5131,5137,5139,5172,5192,5193,5198,5998,5398,4458) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_F\_a")

### （30）class\_H\_a

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_H\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD in (9498) and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_H\_a")

### （31）class\_Z\_a

df=spark.sql("""

select f.id as id, (sum(e.BILL\_AMT))/100 as class\_Z\_a from feature.full1 f inner join (

select ACCTNBR,VAL\_DATE,BILL\_AMT

from standard.event

where MER\_CAT\_CD = 0000 and (REV\_IND =" " or REV\_IND =0) and trans\_type!="1050"

) as e

on f.xaccount=e.ACCTNBR

where

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(e.VAL\_DATE as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("class\_Z\_a")

## 23、过去X个月办理分期的次数

### （1）two\_fq\_count过去2个月的分期次数

df=spark.sql("""

select f.id as id,count(\*) as two\_fq\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-2)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_fq\_count")

### （2）three\_fq\_count过去3个月的分期次数

df=spark.sql("""

select f.id as id,count(\*) as three\_fq\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-3)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_fq\_count")

### （3）six\_fq\_count过去6个月的分期次数

df=spark.sql("""

select f.id as id,count(\*) as six\_fq\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-6)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_fq\_count")

### （4）history\_fq\_count历史的总分期次数

df=spark.sql("""

select f.id as id,count(\*) as history\_fq\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("history\_fq\_count")

## 24、过去X个月办理分期的金额

### （1）two\_fq\_amt过去2个月的分期金额

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_PURCH))/100 as two\_fq\_amt from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH ,ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-2)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_fq\_amt")

### （2）three\_fq\_amt过去3个月的分期金额

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_PURCH))/100 as three\_fq\_amt from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH ,ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-3)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_fq\_amt")

### （3）six\_fq\_amt过去6个月的分期金额

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_PURCH))/100 as six\_fq\_amt from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH ,ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-6)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_fq\_amt")

### （4）history\_fq\_amt历史上的分期金额

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_PURCH))/100 as history\_fq\_amt from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH ,ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("history\_fq\_amt")

## 25、过去X个月办理分期的利息金额

### （1）two\_fq\_interest过去2个月的分期利息金额

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_INT))/100 as two\_fq\_interest from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-2)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_fq\_interest")

### （2）three\_fq\_interest过去3个月的分期利息金额

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_INT))/100 as three\_fq\_interest from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-3)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_fq\_interest")

### （3）six\_fq\_interest过去6个月的分期利息金额

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_INT))/100 as six\_fq\_interest from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-6)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_fq\_interest")

### （4）history\_fq\_interest历史上的分期利息金额

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_INT))/100 as history\_fq\_interest from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("history\_fq\_interest")

## 26、过去X个月办理分期的总费用

### （1）two\_fq\_fee过去2个月的分期总费用

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_FEE))/100 as two\_fq\_fee from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-2)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_fq\_fee")

### （2）three\_fq\_fee过去3个月的分期总费用

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_FEE))/100 as three\_fq\_fee from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-3)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_fq\_fee")

### （3）six\_fq\_fee过去6个月的分期总费用

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_FEE))/100 as six\_fq\_fee from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-6)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_fq\_fee")

### （4）history\_fq\_fee历史上的分期总费用

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_FEE))/100 as history\_fq\_fee from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("history\_fq\_fee")

## 27、history\_max\_fq历史上办理分期的最大分期数

df=spark.sql("""

select f.id as id,max(m.NBR\_MTHS) as history\_max\_fq from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,NBR\_MTHS,REM\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m.XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("history\_max\_fq")

## 28、过去X个月办理账单分期的总月数

### （1）one\_fq\_mths\_count 过去1个月办理分期的总月数

df=spark.sql("""

select f.id as id,sum(m.NBR\_MTHS) as one\_fq\_mths\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,NBR\_MTHS,REM\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("one\_fq\_mths\_count")

### （2）two\_fq\_mths\_count 过去2个月办理账单分期的总月数

df=spark.sql("""

select f.id as id,sum(m.NBR\_MTHS) as two\_fq\_mths\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,NBR\_MTHS,REM\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-2)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_fq\_mths\_count")

### （3）three\_fq\_mths\_count 过去3个月办理分期的总月数

df=spark.sql("""

select f.id as id,sum(m.NBR\_MTHS) as three\_fq\_mths\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,NBR\_MTHS,REM\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-3)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_fq\_mths\_count")

### （4）six\_fq\_mths\_count 过去6个月办理分期的总月数

df=spark.sql("""

select f.id as id,sum(m.NBR\_MTHS) as six\_fq\_mths\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,NBR\_MTHS,REM\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_fq\_mths\_count")

### (5) history\_fq\_mths\_count历史上办理分期的总分期月数

df=spark.sql("""

select f.id as id,sum(m.NBR\_MTHS) as history\_fq\_mths\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,NBR\_MTHS,REM\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("history\_fq\_mths\_count")

## 29、is\_fq该账户是否有办过分期

df=spark.sql("""

select f.id as id, (case when count(\*) >= 1 then '1' else '0' end) as is\_fq from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH,ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N")

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("is\_fq")

## 30、is\_zdfq该账户是否有办过账单分期

df=spark.sql("""

select f.id as id, (case when count(\*) >= 1 then '1' else '0' end) as is\_fq from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH,ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("is\_zdfq")

## 31、过去X个月办理账单分期的次数

### （1）two\_zdfq\_count过去2个月办理账单分期次数

df=spark.sql("""

select f.id as id,count(\*) as two\_zdfq\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-2)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_zdfq\_count")

### （2）three\_zdfq\_count过去3个月办理账单分期分期次数

df=spark.sql("""

select f.id as id,count(\*) as three\_zdfq\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-3)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_zdfq\_count")

### （3）six\_zdfq\_count过去6个月办理账单分期次数

df=spark.sql("""

select f.id as id,count(\*) as six\_zdfq\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-6)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_zdfq\_count")

### （4）history\_zdfq\_count历史上办理账单分期总次数

df=spark.sql("""

select f.id as id,count(\*) as history\_zdfq\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("history\_zdfq\_count")

## 32、过去X个月办理账单分期的金额

### （1）two\_zdfq\_amt过去2个月办理账单分期金额

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_PURCH))/100 as two\_zdfq\_amt from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH ,ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-2)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_zdfq\_amt")

### （2）three\_zdfq\_amt过去3个月办理账单分期金额

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_PURCH))/100 as three\_zdfq\_amt from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH ,ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-3)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_zdfq\_amt")

### （3）six\_zdfq\_amt过去6个月办理账单分期金额

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_PURCH))/100 as six\_zdfq\_amt from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH ,ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-6)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_zdfq\_amt")

### （4）history\_zdfq\_amt历史上办理账单分期总金额

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_PURCH))/100 as history\_zdfq\_amt from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH ,ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("history\_zdfq\_amt")

## 33、过去X个月办理账单分期的利息金额

### （1）two\_zdfq\_interest过去2个月办理账单分期利息金额

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_INT))/100 as two\_zdfq\_interest from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-2)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_zdfq\_interest")

### （2）three\_zdfq\_interest过去3个月办理账单分期利息金额

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_INT))/100 as three\_zdfq\_interest from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-3)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_zdfq\_interest")

### （3）six\_zdfq\_interest过去6个月办理账单分期利息金额

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_INT))/100 as six\_zdfq\_interest from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-6)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_zdfq\_interest")

### （4）history\_zdfq\_interest历史上办理账单分期利息金额

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_INT))/100 as history\_zdfq\_interest from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("history\_zdfq\_interest")

## 34、过去X个月办理账单分期的总费用

### （1）two\_zdfq\_fee过去2个月办理账单分期总费用

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_FEE))/100 as two\_zdfq\_fee from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-2)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_zdfq\_fee")

### （2）three\_zdfq\_fee过去3个月办理账单分期总费用

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_FEE))/100 as three\_zdfq\_fee from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-3)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_zdfq\_fee")

### （3）six\_zdfq\_fee过去6个月办理账单分期总费用

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_FEE))/100 as six\_zdfq\_fee from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-6)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_zdfq\_fee")

### （4）history\_zdfq\_fee历史上办理账单分期总费用

df=spark.sql("""

select f.id as id,(sum(m.ORIG\_FEE))/100 as history\_zdfq\_fee from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,ORIG\_PURCH, ORIG\_INT,ORIG\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("history\_zdfq\_fee")

## 35、history\_max\_zdfq历史上办理账单分期的最大分期数

df=spark.sql("""

select f.id as id,max(m.NBR\_MTHS) as history\_max\_zdfq from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,NBR\_MTHS,REM\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m.XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("history\_max\_zdfq")

## 36、过去X个月办理账单分期的总月数

### （1）one\_zdfq\_mths\_count 过去1个月办理账单分期的总月数

df=spark.sql("""

select f.id as id,sum(m.NBR\_MTHS) as one\_zdfq\_mths\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,NBR\_MTHS,REM\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("one\_zdfq\_mths\_count")

### （2）two\_zdfq\_mths\_count 过去2个月办理账单分期的总月数

df=spark.sql("""

select f.id as id,sum(m.NBR\_MTHS) as two\_zdfq\_mths\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,NBR\_MTHS,REM\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-2)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("two\_zdfq\_mths\_count")

### （3）three\_zdfq\_mths\_count 过去3个月办理账单分期的总月数

df=spark.sql("""

select f.id as id,sum(m.NBR\_MTHS) as three\_zdfq\_mths\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,NBR\_MTHS,REM\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-3)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("three\_zdfq\_mths\_count")

### （4）six\_zdfq\_mths\_count 过去6个月办理账单分期的总月数

df=spark.sql("""

select f.id as id,sum(m.NBR\_MTHS) as six\_zdfq\_mths\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,NBR\_MTHS,REM\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

and

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

>= add\_months(to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp)),-1)

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("six\_zdfq\_mths\_count")

### （5）history\_zdfq\_mths\_count历史上办理账单分期的总分期月数

df=spark.sql("""

select f.id as id,sum(m.NBR\_MTHS) as history\_zdfq\_mths\_count from feature.full1 f inner join (

select XACCOUNT,PURCH\_DAY ,MP\_NUMBER,NBR\_MTHS,REM\_FEE, STATUS

from standard.mpur

where STATUS in ("A","C","P","N") and PROD\_ID like "ZD%"

) as m

on f.xaccount=m. XACCOUNT

where

to\_date(cast(unix\_timestamp(cast(cast(m.PURCH\_DAY as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

< to\_date(cast(unix\_timestamp(cast(cast(f.bill\_date as decimal(8,0)) as varchar(8)),'yyyyMMdd') as timestamp))

group by f.id

""")

spark.sql("use feature")

df.write.saveAsTable("history\_zdfq\_mths\_count")

## 37、过去X个月贷记卡存入金额的均值

### （1）two\_sum\_c\_amt\_mean 过去2个月贷记卡存入金额的均值

df=spark.sql("""

select a.id as id, round((a.two\_sum\_c\_amt/2),2) as two\_sum\_c\_amt\_mean

from feature.two\_sum\_c\_amt a

""")

spark.sql("use feature")

df.write.saveAsTable("two\_sum\_c\_amt\_mean")

### （2）three\_sum\_c\_amt\_mean 过去3个月贷记卡存入金额的均值

df=spark.sql("""

select a.id as id, round((a.three\_sum\_c\_amt/3),2) as three\_sum\_c\_amt\_mean

from feature.three\_sum\_c\_amt a

""")

spark.sql("use feature")

df.write.saveAsTable("three\_sum\_c\_amt\_mean")

### （3）six\_sum\_c\_amt\_mean 过去6个月贷记卡存入金额的均值

df=spark.sql("""

select a.id as id, round((a.six\_sum\_c\_amt/6),2) as six\_sum\_c\_amt\_mean

from feature.six\_sum\_c\_amt a

""")

spark.sql("use feature")

df.write.saveAsTable("six\_sum\_c\_amt\_mean")

## 38、过去第X个月贷记卡存入总额

### （1）twom\_c\_amt 过去第二个月贷记卡存入总额

df=spark.sql("""

select a.id as id, round((a.two\_sum\_c\_amt - b.one\_sum\_c \_amt),2) as twom\_c\_amt

from two\_sum\_c \_amt as a,one\_sum\_c \_amt as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("twom\_c\_amt")

### （2）threem\_c\_amt 过去第三个月贷记卡存入总额

df=spark.sql("""

select a.id as id, round((a.three\_sum\_c\_amt - b.two\_sum\_c \_amt),2) as threem\_c\_amt

from three\_sum\_c \_amt as a,two\_sum\_c \_amt as b

where a.id = b.id

""")

spark.sql("use feature")

df.write.saveAsTable("threem\_c\_amt")

# 五、合并特征

## 1.all1

spark.sql("use feature")

one\_max\_amt =spark.sql("select \* from one\_max\_amt ")

two\_max\_amt =spark.sql("select \* from two\_max\_amt ")

three\_max\_amt =spark.sql("select \* from three\_max\_amt ")

six\_max\_amt =spark.sql("select \* from six\_max\_amt ")

one\_max\_rate =spark.sql("select \* from one\_max\_rate ")

two\_max\_rate =spark.sql("select \* from two\_max\_rate ")

three\_max\_rate =spark.sql("select \* from three\_max\_rate ")

six\_max\_rate =spark.sql("select \* from six\_max\_rate ")

one\_sum\_amt =spark.sql("select \* from one\_sum\_amt ")

two\_sum\_amt =spark.sql("select \* from two\_sum\_amt ")

three\_sum\_amt =spark.sql("select \* from three\_sum\_amt ")

six\_sum\_amt =spark.sql("select \* from six\_sum\_amt ")

one\_sum\_rate =spark.sql("select \* from one\_sum\_rate ")

two\_sum\_rate =spark.sql("select \* from two\_sum\_rate ")

three\_sum\_rate =spark.sql("select \* from three\_sum\_rate ")

six\_sum\_rate =spark.sql("select \* from six\_sum\_rate ")

one\_sum\_c\_amt =spark.sql("select \* from one\_sum\_c\_amt ")

two\_sum\_c\_amt =spark.sql("select \* from two\_sum\_c\_amt ")

three\_sum\_c\_amt =spark.sql("select \* from three\_sum\_c\_amt ")

six\_sum\_c\_amt =spark.sql("select \* from six\_sum\_c\_amt ")

one\_buy\_count =spark.sql("select \* from one\_buy\_count ")

two\_buy\_count =spark.sql("select \* from two\_buy\_count ")

three\_buy\_count =spark.sql("select \* from three\_buy\_count ")

six\_buy\_count =spark.sql("select \* from six\_buy\_count ")

full1=spark.sql("select \* from feature.full1 ")

all1= full1.join(one\_max\_amt,'id','left\_outer').join(two\_max\_amt,'id','left\_outer')\

.join(three\_max\_amt,'id','left\_outer').join(six\_max\_amt,'id','left\_outer')\

.join(one\_max\_rate,'id','left\_outer').join(two\_max\_rate,'id','left\_outer')\

.join(three\_max\_rate,'id','left\_outer').join(six\_max\_rate,'id','left\_outer')\

.join(one\_sum\_amt,'id','left\_outer').join(two\_sum\_amt,'id','left\_outer')\

.join(three\_sum\_amt,'id','left\_outer').join(six\_sum\_amt,'id','left\_outer')\

.join(one\_sum\_rate,'id','left\_outer').join(two\_sum\_rate,'id','left\_outer')\

.join(three\_sum\_rate,'id','left\_outer').join(six\_sum\_rate,'id','left\_outer')\

.join(one\_sum\_c\_amt,'id','left\_outer').join(two\_sum\_c\_amt,'id','left\_outer')\

.join(three\_sum\_c\_amt,'id','left\_outer').join(six\_sum\_c\_amt,'id','left\_outer')\

.join(one\_buy\_count,'id','left\_outer').join(two\_buy\_count,'id','left\_outer')\

.join(three\_buy\_count,'id','left\_outer').join(six\_buy\_count,'id','left\_outer')

all1.write.saveAsTable("all1")

## 2.all2

spark.sql("use feature")

one\_amt\_cha =spark.sql("select \* from one\_amt\_cha ")

two\_amt\_cha =spark.sql("select \* from two\_amt\_cha ")

three\_amt\_cha =spark.sql("select \* from three\_amt\_cha ")

six\_amt\_cha =spark.sql("select \* from six\_amt\_cha ")

twom\_count =spark.sql("select \* from twom\_count ")

threem\_count =spark.sql("select \* from threem\_count ")

two\_sum\_amt\_mean =spark.sql("select \* from two\_sum\_amt\_mean ")

three\_sum\_amt\_mean =spark.sql("select \* from three\_sum\_amt\_mean ")

six\_sum\_amt\_mean =spark.sql("select \* from six\_sum\_amt\_mean ")

twom\_count\_mean =spark.sql("select \* from twom\_count\_mean ")

threem\_count\_mean =spark.sql("select \* from threem\_count\_mean ")

sixm\_count\_mean =spark.sql("select \* from sixm\_count\_mean ")

one\_to\_two\_count\_mean\_cha =spark.sql("select \* from one\_to\_two\_count\_mean\_cha ")

one\_to\_three\_count\_mean\_cha =spark.sql("select \* from one\_to\_three\_count\_mean\_cha ")

two\_to\_one\_count\_cha =spark.sql("select \* from two\_to\_one\_count\_cha ")

three\_to\_two\_count\_cha =spark.sql("select \* from three\_to\_two\_count\_cha ")

two\_to\_one\_count\_rate =spark.sql("select \* from two\_to\_one\_count\_rate ")

three\_to\_two\_count\_rate =spark.sql("select \* from three\_to\_two\_count\_rate ")

two\_to\_one\_amt\_cha\_mean =spark.sql("select \* from two\_to\_one\_amt\_cha\_mean ")

three\_to\_two\_amt\_cha\_mean =spark.sql("select \* from three\_to\_two\_amt\_cha\_mean ")

six\_to\_three\_amt\_cha\_mean =spark.sql("select \* from six\_to\_three\_amt\_cha\_mean ")

twom\_max\_amt =spark.sql("select \* from twom\_max\_amt ")

threem\_max\_amt =spark.sql("select \* from threem\_max\_amt ")

twom\_to\_one\_max\_cha =spark.sql("select \* from twom\_to\_one\_max\_cha ")

threem\_to\_twom\_max\_cha =spark.sql("select \* from threem\_to\_twom\_max\_cha ")

twom\_to\_one\_max\_rate =spark.sql("select \* from twom\_to\_one\_max\_rate ")

threem\_to\_twom\_max\_rate =spark.sql("select \* from threem\_to\_twom\_max\_rate ")

twom\_amt =spark.sql("select \* from twom\_amt ")

threem\_amt =spark.sql("select \* from threem\_amt ")

twom\_to\_one\_amt\_cha =spark.sql("select \* from twom\_to\_one\_amt\_cha ")

threem\_to\_twom\_amt\_cha =spark.sql("select \* from threem\_to\_twom\_amt\_cha ")

twom\_to\_one\_amt\_rate =spark.sql("select \* from twom\_to\_one\_amt\_rate ")

threem\_to\_twom\_amt\_rate =spark.sql("select \* from threem\_to\_twom\_amt\_rate ")

all1=spark.sql("select \* from feature.all1 ")

all2= all1.join(one\_amt\_cha,'id','left\_outer').join(two\_amt\_cha,'id','left\_outer')\

.join(three\_amt\_cha,'id','left\_outer').join(six\_amt\_cha,'id','left\_outer')\

.join(twom\_count,'id','left\_outer').join(threem\_count,'id','left\_outer')\

.join(two\_sum\_amt\_mean,'id','left\_outer').join(three\_sum\_amt\_mean,'id','left\_outer')\

.join(six\_sum\_amt\_mean,'id','left\_outer').join(twom\_count\_mean,'id','left\_outer')\

.join(threem\_count\_mean,'id','left\_outer').join(sixm\_count\_mean,'id','left\_outer')\

.join(one\_to\_two\_count\_mean\_cha,'id','left\_outer')\

.join(one\_to\_three\_count\_mean\_cha,'id','left\_outer')\

.join(two\_to\_one\_count\_cha,'id','left\_outer')\

.join(three\_to\_two\_count\_cha,'id','left\_outer')\

.join(two\_to\_one\_count\_rate,'id','left\_outer')\

.join(three\_to\_two\_count\_rate,'id','left\_outer')\

.join(two\_to\_one\_amt\_cha\_mean,'id','left\_outer')\

.join(three\_to\_two\_amt\_cha\_mean,'id','left\_outer')\

.join(six\_to\_three\_amt\_cha\_mean,'id','left\_outer')\

.join(twom\_max\_amt,'id','left\_outer').join(threem\_max\_amt,'id','left\_outer')\

.join(twom\_to\_one\_max\_cha,'id','left\_outer').join(threem\_to\_twom\_max\_cha,'id','left\_outer')\

.join(twom\_to\_one\_max\_rate,'id','left\_outer').join(threem\_to\_twom\_max\_rate,'id','left\_outer')\

.join(twom\_amt,'id','left\_outer').join(threem\_amt,'id','left\_outer')\

.join(twom\_to\_one\_amt\_cha,'id','left\_outer').join(threem\_to\_twom\_amt\_cha,'id','left\_outer')\

.join(twom\_to\_one\_amt\_rate,'id','left\_outer').join(threem\_to\_twom\_amt\_rate,'id','left\_outer')

all2.write.saveAsTable("all2")

## 3.all3

spark.sql("use feature")

A\_a =spark.sql("select \* from A\_a ")

A\_b =spark.sql("select \* from A\_b ")

A\_c =spark.sql("select \* from A\_c ")

A\_d =spark.sql("select \* from A\_d ")

A\_e =spark.sql("select \* from A\_e ")

B\_a =spark.sql("select \* from B\_a ")

B\_b =spark.sql("select \* from B\_b ")

C\_a =spark.sql("select \* from C\_a ")

C\_b =spark.sql("select \* from C\_b ")

C\_c =spark.sql("select \* from C\_c ")

C\_d =spark.sql("select \* from C\_d ")

C\_e =spark.sql("select \* from C\_e ")

C\_f =spark.sql("select \* from C\_f ")

D\_a =spark.sql("select \* from D\_a ")

D\_b =spark.sql("select \* from D\_b ")

D\_c =spark.sql("select \* from D\_c ")

E\_a =spark.sql("select \* from E\_a ")

E\_b =spark.sql("select \* from E\_b ")

E\_c =spark.sql("select \* from E\_c ")

E\_d =spark.sql("select \* from E\_d ")

E\_e =spark.sql("select \* from E\_e ")

E\_f =spark.sql("select \* from E\_f ")

E\_g =spark.sql("select \* from E\_g ")

F\_a =spark.sql("select \* from F\_a ")

H\_a =spark.sql("select \* from H\_a ")

Z\_a =spark.sql("select \* from Z\_a ")

all2=spark.sql("select \* from all2 ")

all3= all2.join(A\_a,'id','left\_outer').join(A\_b,'id','left\_outer').join(A\_c,'id','left\_outer')\

.join(A\_d,'id','left\_outer').join(A\_e,'id','left\_outer').join(B\_a,'id','left\_outer')\

.join(B\_b,'id','left\_outer').join(C\_a,'id','left\_outer').join(C\_b,'id','left\_outer')\

.join(C\_c,'id','left\_outer').join(C\_d,'id','left\_outer').join(C\_e,'id','left\_outer')\

.join(C\_f,'id','left\_outer').join(D\_a,'id','left\_outer').join(D\_b,'id','left\_outer')\

.join(D\_c,'id','left\_outer').join(E\_a,'id','left\_outer').join(E\_b,'id','left\_outer')\

.join(E\_c,'id','left\_outer').join(E\_d,'id','left\_outer').join(E\_e,'id','left\_outer')\

.join(E\_f,'id','left\_outer').join(E\_g,'id','left\_outer').join(F\_a,'id','left\_outer')\

.join(H\_a,'id','left\_outer').join(Z\_a,'id','left\_outer')

all3.write.saveAsTable("all3")

## 4.all4

spark.sql("use feature")

class\_A\_a =spark.sql("select \* from class\_A\_a ")

class\_A\_b =spark.sql("select \* from class\_A\_b ")

class\_A\_c =spark.sql("select \* from class\_A\_c ")

class\_A\_d =spark.sql("select \* from class\_A\_d ")

class\_A\_e =spark.sql("select \* from class\_A\_e ")

class\_B\_a =spark.sql("select \* from class\_B\_a ")

class\_B\_b =spark.sql("select \* from class\_B\_b ")

class\_C\_a =spark.sql("select \* from class\_C\_a ")

class\_C\_b =spark.sql("select \* from class\_C\_b ")

class\_C\_c =spark.sql("select \* from class\_C\_c ")

class\_C\_d =spark.sql("select \* from class\_C\_d ")

class\_C\_e =spark.sql("select \* from class\_C\_e ")

class\_C\_f =spark.sql("select \* from class\_C\_f ")

class\_D\_a =spark.sql("select \* from class\_D\_a ")

class\_D\_b =spark.sql("select \* from class\_D\_b ")

class\_D\_c =spark.sql("select \* from class\_D\_c ")

class\_E\_a =spark.sql("select \* from class\_E\_a ")

class\_E\_b =spark.sql("select \* from class\_E\_b ")

class\_E\_c =spark.sql("select \* from class\_E\_c ")

class\_E\_d =spark.sql("select \* from class\_E\_d ")

class\_E\_e =spark.sql("select \* from class\_E\_e ")

class\_E\_f =spark.sql("select \* from class\_E\_f ")

class\_E\_g =spark.sql("select \* from class\_E\_g ")

class\_F\_a =spark.sql("select \* from class\_F\_a ")

class\_H\_a =spark.sql("select \* from class\_H\_a ")

class\_Z\_a =spark.sql("select \* from class\_Z\_a ")

all3=spark.sql("select \* from all3 ")

all4= all3.join(class\_A\_a,'id','left\_outer').join(class\_A\_b,'id','left\_outer')\

.join(class\_A\_c,'id','left\_outer').join(class\_A\_d,'id','left\_outer')\

.join(class\_A\_e,'id','left\_outer').join(class\_B\_a,'id','left\_outer')\

.join(class\_B\_b,'id','left\_outer').join(class\_C\_a,'id','left\_outer')\

.join(class\_C\_b,'id','left\_outer').join(class\_C\_c,'id','left\_outer')\

.join(class\_C\_d,'id','left\_outer').join(class\_C\_e,'id','left\_outer')\

.join(class\_C\_f,'id','left\_outer').join(class\_D\_a,'id','left\_outer')\

.join(class\_D\_b,'id','left\_outer').join(class\_D\_c,'id','left\_outer')\

.join(class\_E\_a,'id','left\_outer').join(class\_E\_b,'id','left\_outer')\

.join(class\_E\_c,'id','left\_outer').join(class\_E\_d,'id','left\_outer')\

.join(class\_E\_e,'id','left\_outer').join(class\_E\_f,'id','left\_outer')\

.join(class\_E\_g,'id','left\_outer').join(class\_F\_a,'id','left\_outer')\

.join(class\_H\_a,'id','left\_outer').join(class\_Z\_a,'id','left\_outer')

all4.write.saveAsTable("all4")

## 5.alll5

spark.sql("use feature")

two\_fq\_count =spark.sql("select \* from two\_fq\_count ")

three\_fq\_count =spark.sql("select \* from three\_fq\_count ")

six\_fq\_count =spark.sql("select \* from six\_fq\_count ")

history\_fq\_count =spark.sql("select \* from history\_fq\_count ")

two\_fq\_amt =spark.sql("select \* from two\_fq\_amt ")

three\_fq\_amt =spark.sql("select \* from three\_fq\_amt ")

six\_fq\_amt =spark.sql("select \* from six\_fq\_amt ")

history\_fq\_amt =spark.sql("select \* from history\_fq\_amt ")

two\_fq\_interest =spark.sql("select \* from two\_fq\_interest ")

three\_fq\_interest =spark.sql("select \* from three\_fq\_interest ")

six\_fq\_interest =spark.sql("select \* from six\_fq\_interest ")

history\_fq\_interest =spark.sql("select \* from history\_fq\_interest ")

two\_fq\_fee =spark.sql("select \* from two\_fq\_fee ")

three\_fq\_fee =spark.sql("select \* from three\_fq\_fee ")

six\_fq\_fee =spark.sql("select \* from six\_fq\_fee ")

history\_fq\_fee =spark.sql("select \* from history\_fq\_fee ")

history\_max\_fq =spark.sql("select \* from history\_max\_fq ")

one\_fq\_mths\_count =spark.sql("select \* from one\_fq\_mths\_count ")

two\_fq\_mths\_count =spark.sql("select \* from two\_fq\_mths\_count ")

three\_fq\_mths\_count =spark.sql("select \* from three\_fq\_mths\_count ")

six\_fq\_mths\_count =spark.sql("select \* from six\_fq\_mths\_count ")

history\_fq\_mths\_count =spark.sql("select \* from history\_fq\_mths\_count ")

is\_fq =spark.sql("select \* from is\_fq ")

is\_zdfq =spark.sql("select \* from is\_zdfq ")

all4 =spark.sql("select \* from all4 ")

all5= all4.join(two\_fq\_count,'id','left\_outer').join(three\_fq\_count,'id','left\_outer')\

.join(six\_fq\_count,'id','left\_outer').join(history\_fq\_count,'id','left\_outer')\

.join(two\_fq\_amt,'id','left\_outer').join(three\_fq\_amt,'id','left\_outer')\

.join(six\_fq\_amt,'id','left\_outer').join(history\_fq\_amt,'id','left\_outer')\

.join(two\_fq\_interest,'id','left\_outer').join(three\_fq\_interest,'id','left\_outer')\

.join(six\_fq\_interest,'id','left\_outer').join(history\_fq\_interest,'id','left\_outer')\

.join(two\_fq\_fee,'id','left\_outer').join(three\_fq\_fee,'id','left\_outer')\

.join(six\_fq\_fee,'id','left\_outer').join(history\_fq\_fee,'id','left\_outer')\

.join(history\_max\_fq,'id','left\_outer').join(one\_fq\_mths\_count,'id','left\_outer')\

.join(two\_fq\_mths\_count,'id','left\_outer').join(three\_fq\_mths\_count,'id','left\_outer')\

.join(six\_fq\_mths\_count,'id','left\_outer').join(history\_fq\_mths\_count,'id','left\_outer')\

.join(is\_fq,'id','left\_outer').join(is\_zdfq,'id','left\_outer')

all5.write.saveAsTable("all5")

## 6.all6

spark.sql("use feature")

two\_zdfq\_count =spark.sql("select \* from two\_zdfq\_count ")

three\_zdfq\_count =spark.sql("select \* from three\_zdfq\_count ")

six\_zdfq\_count =spark.sql("select \* from six\_zdfq\_count ")

history\_zdfq\_count =spark.sql("select \* from history\_zdfq\_count ")

two\_zdfq\_amt =spark.sql("select \* from two\_zdfq\_amt ")

three\_zdfq\_amt =spark.sql("select \* from three\_zdfq\_amt ")

six\_zdfq\_amt =spark.sql("select \* from six\_zdfq\_amt ")

history\_zdfq\_amt =spark.sql("select \* from history\_zdfq\_amt ")

two\_zdfq\_interest =spark.sql("select \* from two\_zdfq\_interest ")

three\_zdfq\_interest =spark.sql("select \* from three\_zdfq\_interest ")

six\_zdfq\_interest =spark.sql("select \* from six\_zdfq\_interest ")

history\_zdfq\_interest =spark.sql("select \* from history\_zdfq\_interest ")

two\_zdfq\_fee =spark.sql("select \* from two\_zdfq\_fee ")

three\_zdfq\_fee =spark.sql("select \* from three\_zdfq\_fee ")

six\_zdfq\_fee =spark.sql("select \* from six\_zdfq\_fee ")

history\_zdfq\_fee =spark.sql("select \* from history\_zdfq\_fee ")

history\_max\_zdfq =spark.sql("select \* from history\_max\_zdfq ")

one\_zdfq\_mths\_count =spark.sql("select \* from one\_zdfq\_mths\_count ")

two\_zdfq\_mths\_count =spark.sql("select \* from two\_zdfq\_mths\_count ")

three\_zdfq\_mths\_count =spark.sql("select \* from three\_zdfq\_mths\_count ")

six\_zdfq\_mths\_count =spark.sql("select \* from six\_zdfq\_mths\_count ")

history\_zdfq\_mths\_count =spark.sql("select \* from history\_zdfq\_mths\_count ")

two\_sum\_c\_amt\_mean =spark.sql("select \* from two\_sum\_c\_amt\_mean ")

three\_sum\_c\_amt\_mean =spark.sql("select \* from three\_sum\_c\_amt\_mean ")

six\_sum\_c\_amt\_mean =spark.sql("select \* from six\_sum\_c\_amt\_mean ")

twom\_c\_amt = spark.sql("select \* from twom\_c\_amt ")

threem\_c\_amt = spark.sql("select \* from threem\_c\_amt ")

all5 =spark.sql("select \* from all5 ")

all6= all5.join(two\_zdfq\_count,'id','left\_outer').join(three\_zdfq\_count,'id','left\_outer')\

.join(six\_zdfq\_count,'id','left\_outer').join(history\_zdfq\_count,'id','left\_outer')\

.join(two\_zdfq\_amt,'id','left\_outer').join(three\_zdfq\_amt,'id','left\_outer')\

.join(six\_zdfq\_amt,'id','left\_outer').join(history\_zdfq\_amt,'id','left\_outer')\

.join(two\_zdfq\_interest,'id','left\_outer').join(three\_zdfq\_interest,'id','left\_outer')\

.join(six\_zdfq\_interest,'id','left\_outer').join(history\_zdfq\_interest,'id','left\_outer')\

.join(two\_zdfq\_fee,'id','left\_outer').join(three\_zdfq\_fee,'id','left\_outer')\

.join(six\_zdfq\_fee,'id','left\_outer').join(history\_zdfq\_fee,'id','left\_outer')\

.join(history\_max\_zdfq,'id','left\_outer') .join(one\_zdfq\_mths\_count,'id','left\_outer')\

.join(two\_zdfq\_mths\_count,'id','left\_outer') .join(three\_zdfq\_mths\_count,'id','left\_outer')\

.join(six\_zdfq\_mths\_count,'id','left\_outer').join(history\_zdfq\_mths\_count,'id','left\_outer')\

.join(two\_sum\_c\_amt\_mean,'id','left\_outer') .join(three\_sum\_c\_amt\_mean,'id','left\_outer')\

.join(six\_sum\_c\_amt\_mean,'id','left\_outer') .join(twom\_c\_amt,'id','left\_outer')\

.join(threem\_c\_amt,'id','left\_outer')

all6.write.saveAsTable("all6")

## 7.all7

spark.sql("use feature")

custr =spark.sql("select \* from custr ")

acct =spark.sql("select \* from acct ")

card =spark.sql("select \* from card ")

stmt =spark.sql("select \* from stmt ")

label =spark.sql("select \* from label ")

all6 =spark.sql("select \* from all6 ")

all7=all6.join(custr,'id','left\_outer').join(acct,'id','left\_outer').join(card,'id','left\_outer') \

.join(stmt,'id','left\_outer').join(label,'id','left\_outer')

all7.write.saveAsTable("all7")

# 六、筛选样本的几种方法(具体是否这样筛选需和客户沟通确认)

## 1.根据账户状态（根据实际情况定取值）

df = spark.sql("""select \*

from feature.all7

where CLOSE\_CODE = ' ' """)

df.write.saveAsTable("all8")

## 2.根据账户类别（根据实际情况定取值）

df = spark.sql("""select \*

from feature.all7

where CATEGORY = 0020 """)

df.write.saveAsTable("all8")

## 3.根据账单金额（根据实际情况定取值）

df = spark.sql("""select \*

from feature.all7

where STM\_BALNCE > 200 """)

df.write.saveAsTable("all8")

## 4.根据当前逾期期数（根据实际情况定取值）

df = spark.sql("""select \*

from feature.all7

where MTHS\_ODUE < 2 """)

df.write.saveAsTable("all8")

## 5.根据账单日（根据实际情况定取值）

df = spark.sql("""select \*

from feature.all7

where CYCLE\_NBR = 1 or CYCLE\_NBR = 10 """)

df.write.saveAsTable("all8")

## 6.根据账户逾期状态（根据实际情况定取值）

df = spark.sql("""select \*

from feature.all7

where ODUE\_FLAG = 0 """)

df.write.saveAsTable("all8")