## 7.8 情景题---》做题

### 7.8.1 蚂蚁森林植物申领统计

#### 7.8.1.1 背景说明

下表记录了用户每天的蚂蚁森林低碳生活领取的记录流水。

table\_name：user\_low\_carbon

user\_id  data\_dt  low\_carbon

用户     日期      减少碳排放（g）

蚂蚁森林植物换购表，用于记录申领环保植物所需要减少的碳排放量table\_name:  plant\_carbon

plant\_id  plant\_name  low\_carbon

植物编号 植物名 换购植物所需要的碳

#### 7.8.1.2 原始数据样例

user\_low\_carbon：

user\_id date\_dt low\_carbon

u\_001 2017/1/1 10

u\_001 2017/1/2 150

u\_001 2017/1/2 110

u\_001 2017/1/2 10

u\_001 2017/1/4 50

u\_001 2017/1/4 10

u\_001 2017/1/6 45

u\_001 2017/1/6 90

u\_002 2017/1/1 10

u\_002 2017/1/2 150

u\_002 2017/1/2 70

u\_002 2017/1/3 30

u\_002 2017/1/3 80

u\_002 2017/1/4 150

u\_002 2017/1/5 101

u\_002 2017/1/6 68

…

plant\_carbon：

plant\_id plant\_name plant\_carbon

p001 梭梭树 17

p002 沙柳 19

p003 樟子树 146

p004 胡杨 215

…

1.创建表

create table user\_low\_carbon(user\_id String,data\_dt String,low\_carbon int) row format delimited fields terminated by '\t';

create table plant\_carbon(plant\_id string,plant\_name string,low\_carbon int) row format delimited fields terminated by '\t';

2.加载数据

load data local inpath "/opt/module/data/low\_carbon.txt" into table user\_low\_carbon;

load data local inpath "/opt/module/data/plant\_carbon.txt" into table plant\_carbon;

3.设置本地模式

set hive.exec.mode.local.auto=true;

#### 7.8.1.2 题目一

蚂蚁森林植物申领统计

问题：假设2017年1月1日开始记录低碳数据（user\_low\_carbon），假设2017年10月1日之前满足申领条件的用户都申领了一颗p004-胡杨，

剩余的能量全部用来领取“p002-沙柳” 。

统计在10月1日累计申领“p002-沙柳” 排名前10的用户信息；以及他比后一名多领了几颗沙柳。

得到的统计结果如下表样式：

user\_id  plant\_count less\_count(比后一名多领了几颗沙柳)

u\_101    1000         100

u\_088    900          400

u\_103    500          …

**SQL流程**

（1）先获取在10月1日前low\_carbon总和最大的11个人

select user\_id,sum(low\_carbon) low\_carbon\_sum

from user\_low\_carbon

where datediff(regexp\_replace(data\_dt, "/", "-"),regexp\_replace('2017/10/1', "/", "-")) < 0

group by user\_id

order by low\_carbon\_sum

desc limit 11;t1

（2）查询出"胡杨"所需的低碳量

select low\_carbon from plant\_carbon where plant\_id='p004';t2

（3）查询出"沙柳"所需的低碳量

select low\_carbon from plant\_carbon where plant\_id='p002';t3

（4）计算出在申领一颗"胡杨"后可申领的"沙柳"棵数

select user\_id,round((t1.low\_carbon\_sum-t2.low\_carbon)/t3.low\_carbon) plant\_count,

from t1,t2,t3;t4

（5）将每一行的下一个申领棵数放在当前行

select user\_id,plant\_count,lead(plant\_count,1,0) over(sort by plant\_count desc) as leadCount from t4;t5

（6）计算最终的差集（前十名比下一名多多少棵）

select user\_id,(plant\_count-leadCount) from t5 limit 10;

（7）最终Sql

SELECT

user\_id,

plant\_count,

(plant\_count - leadCount)

FROM

(

SELECT

user\_id,

plant\_count,

lead (plant\_count, 1, 0) over (sort BY plant\_count DESC) AS leadCount

FROM

(

SELECT

user\_id,

round(

(

t1.low\_carbon\_sum - t2.low\_carbon

) / t3.low\_carbon

) plant\_count

FROM

(

SELECT

user\_id,

sum(low\_carbon) low\_carbon\_sum

FROM

user\_low\_carbon

WHERE

datediff(

regexp\_replace (data\_dt, "/", "-"),

regexp\_replace ('2017/10/1', "/", "-")

) < 0

GROUP BY

user\_id

ORDER BY

low\_carbon\_sum DESC

LIMIT 11

) t1,

(

SELECT

low\_carbon

FROM

plant\_carbon

WHERE

plant\_id = 'p004'

) t2,

(

SELECT

low\_carbon

FROM

plant\_carbon

WHERE

plant\_id = 'p002'

) t3

) t4

) t5

LIMIT 10;

（8）结果展示

+----------+--------------+-------+--+

| user\_id | plant\_count | \_c2 |

+----------+--------------+-------+--+

| u\_007 | 66.0 | 2.0 |

| u\_013 | 64.0 | 10.0 |

| u\_008 | 54.0 | 7.0 |

| u\_005 | 47.0 | 1.0 |

| u\_010 | 46.0 | 2.0 |

| u\_014 | 44.0 | 5.0 |

| u\_011 | 39.0 | 1.0 |

| u\_009 | 38.0 | 6.0 |

| u\_006 | 32.0 | 9.0 |

| u\_002 | 23.0 | 1.0 |

+----------+--------------+-------+--+

#### 7.8.1.3 题目二

蚂蚁森林低碳用户排名分析

问题：查询user\_low\_carbon表中每日流水记录，条件为：

用户在2017年，连续三天（或以上）的天数里，

每天减少碳排放（low\_carbon）都超过100g的用户低碳流水。

需要查询返回满足以上条件的user\_low\_carbon表中的记录流水。

例如用户u\_002符合条件的记录如下，因为2017/1/2~2017/1/5连续四天的碳排放量之和都大于等于100g：

seq（key） user\_id data\_dt  low\_carbon

xxxxx10    u\_002  2017/1/2  150

xxxxx11    u\_002  2017/1/2  70

xxxxx12    u\_002  2017/1/3  30

xxxxx13    u\_002  2017/1/3  80

xxxxx14    u\_002  2017/1/4  150

xxxxx14    u\_002  2017/1/5  101

备注：统计方法不限于sql、procedure、python,java等

**第一种解法：**

**SQL流程**

（1）按照用户及时间聚合，计算每个人每天的低碳量（2017年）

select user\_id,data\_dt,sum(low\_carbon) low\_carbon\_sum from user\_low\_carbon

where substring(data\_dt,1,4)='2017'

group BY user\_id,data\_dt

having low\_carbon\_sum>100;t1

（2）将每一条数据的前后各两条数据的时间放置在一行，默认值为（1970/7/1）

select user\_id,

data\_dt,

lag(data\_dt,2,"1970/7/1") over(partition by user\_id) as lag2Date,

lag(data\_dt,1,"1970/7/1") over(partition by user\_id) as lag1Date,

lead(data\_dt,1,"1970/7/1") over(partition by user\_id) as lead1Date,

lead(data\_dt,2,"1970/7/1") over(partition by user\_id) as lead2Date

from t1;t2

（3）计算每一天数据时间与前后两条数据之间的差值

select user\_id,

data\_dt,

datediff(regexp\_replace(data\_dt, "/", "-"),regexp\_replace(lag2Date, "/", "-")) lag2,

datediff(regexp\_replace(data\_dt, "/", "-"),regexp\_replace(lag1Date, "/", "-")) lag1,

datediff(regexp\_replace(data\_dt, "/", "-"),regexp\_replace(lead1Date, "/", "-")) lead1,

datediff(regexp\_replace(data\_dt, "/", "-"),regexp\_replace(lead2Date, "/", "-")) lead2

from (select user\_id,

data\_dt,

lag(data\_dt,2,"1970/7/1") over(partition by user\_id) as lag2Date,

lag(data\_dt,1,"1970/7/1") over(partition by user\_id) as lag1Date,

lead(data\_dt,1,"1970/7/1") over(partition by user\_id) as lead1Date,

lead(data\_dt,2,"1970/7/1") over(partition by user\_id) as lead2Date

from t2;t3

（4）取出最终需要的值，连续3天的（user\_id,data\_dt）

select user\_id,data\_dt

from t3

where (lag2=2 and lag1 =1) or (lag1 =1 and lead1 = -1) or(lead1=-1 and lead2 = -2);t4

（5）与原表Join得到最终需要的结果

select t5.user\_id,t5.data\_dt,t5.low\_carbon

from user\_low\_carbon t5

join t4

where t4.user\_id = t5.user\_id and t4.data\_dt = t5.data\_dt;

（6）最终Sql

select t5.user\_id,t5.data\_dt,t5.low\_carbon

from user\_low\_carbon t5

join (select user\_id,data\_dt

from (select user\_id,

data\_dt,

datediff(regexp\_replace(data\_dt, "/", "-"),regexp\_replace(lag2Date, "/", "-")) lag2,

datediff(regexp\_replace(data\_dt, "/", "-"),regexp\_replace(lag1Date, "/", "-")) lag1,

datediff(regexp\_replace(data\_dt, "/", "-"),regexp\_replace(lead1Date, "/", "-")) lead1,

datediff(regexp\_replace(data\_dt, "/", "-"),regexp\_replace(lead2Date, "/", "-")) lead2

from (select user\_id,

data\_dt,

lag(data\_dt,2,"1970/7/1") over(partition by user\_id) as lag2Date,

lag(data\_dt,1,"1970/7/1") over(partition by user\_id) as lag1Date,

lead(data\_dt,1,"1970/7/1") over(partition by user\_id) as lead1Date,

lead(data\_dt,2,"1970/7/1") over(partition by user\_id) as lead2Date

from (select user\_id,data\_dt,sum(low\_carbon) low\_carbon\_sum from user\_low\_carbon

where substring(data\_dt,1,4)='2017'

group BY user\_id,data\_dt

having low\_carbon\_sum>100)t1)t2)t3

where (lag2=2 and lag1 =1) or (lag1 =1 and lead1 = -1) or(lead1=-1 and lead2 = -2))t4

where t4.user\_id = t5.user\_id and t4.data\_dt = t5.data\_dt;

SELECT

t5.user\_id,

t5.data\_dt,

t5.low\_carbon

FROM

user\_low\_carbon t5

JOIN (

SELECT

user\_id,

data\_dt

FROM

(

SELECT

user\_id,

data\_dt,

datediff(

regexp\_replace (data\_dt, "/", "-"),

regexp\_replace (lag2Date, "/", "-")

) lag2,

datediff(

regexp\_replace (data\_dt, "/", "-"),

regexp\_replace (lag1Date, "/", "-")

) lag1,

datediff(

regexp\_replace (data\_dt, "/", "-"),

regexp\_replace (lead1Date, "/", "-")

) lead1,

datediff(

regexp\_replace (data\_dt, "/", "-"),

regexp\_replace (lead2Date, "/", "-")

) lead2

FROM

(

SELECT

user\_id,

data\_dt,

lag (data\_dt, 2, "1970/7/1") over (PARTITION BY user\_id) AS lag2Date,

lag (data\_dt, 1, "1970/7/1") over (PARTITION BY user\_id) AS lag1Date,

lead (data\_dt, 1, "1970/7/1") over (PARTITION BY user\_id) AS lead1Date,

lead (data\_dt, 2, "1970/7/1") over (PARTITION BY user\_id) AS lead2Date

FROM

(

SELECT

user\_id,

data\_dt,

sum(low\_carbon) low\_carbon\_sum

FROM

user\_low\_carbon

WHERE

substring(data\_dt, 1, 4) = '2017'

GROUP BY

user\_id,

data\_dt

HAVING

low\_carbon\_sum > 100

) t1

) t2

) t3

WHERE

(lag2 = 2 AND lag1 = 1)

OR (lag1 = 1 AND lead1 = - 1)

OR (lead1 =- 1 AND lead2 = - 2)

) t4

WHERE

t4.user\_id = t5.user\_id

AND t4.data\_dt = t5.data\_dt;

（7）结果展示

+-------------+-------------+----------------+--+

| t5.user\_id | t5.data\_dt | t5.low\_carbon |

+-------------+-------------+----------------+--+

| u\_002 | 2017/1/2 | 150 |

| u\_002 | 2017/1/2 | 70 |

| u\_002 | 2017/1/3 | 30 |

| u\_002 | 2017/1/3 | 80 |

| u\_002 | 2017/1/4 | 150 |

| u\_002 | 2017/1/5 | 101 |

| u\_005 | 2017/1/2 | 50 |

| u\_005 | 2017/1/2 | 80 |

| u\_005 | 2017/1/3 | 180 |

| u\_005 | 2017/1/4 | 180 |

| u\_005 | 2017/1/4 | 10 |

| u\_008 | 2017/1/4 | 260 |

| u\_008 | 2017/1/5 | 360 |

| u\_008 | 2017/1/6 | 160 |

| u\_008 | 2017/1/7 | 60 |

| u\_008 | 2017/1/7 | 60 |

| u\_009 | 2017/1/2 | 70 |

| u\_009 | 2017/1/2 | 70 |

| u\_009 | 2017/1/3 | 170 |

| u\_009 | 2017/1/4 | 270 |

| u\_010 | 2017/1/4 | 90 |

| u\_010 | 2017/1/4 | 80 |

| u\_010 | 2017/1/5 | 90 |

| u\_010 | 2017/1/5 | 90 |

| u\_010 | 2017/1/6 | 190 |

| u\_010 | 2017/1/7 | 90 |

| u\_010 | 2017/1/7 | 90 |

| u\_011 | 2017/1/1 | 110 |

| u\_011 | 2017/1/2 | 100 |

| u\_011 | 2017/1/2 | 100 |

| u\_011 | 2017/1/3 | 120 |

| u\_013 | 2017/1/2 | 150 |

| u\_013 | 2017/1/2 | 50 |

| u\_013 | 2017/1/3 | 150 |

| u\_013 | 2017/1/4 | 550 |

| u\_013 | 2017/1/5 | 350 |

| u\_014 | 2017/1/5 | 250 |

| u\_014 | 2017/1/6 | 120 |

| u\_014 | 2017/1/7 | 270 |

| u\_014 | 2017/1/7 | 20 |

+-------------+-------------+----------------+--+

**第二种解法：**

**SQL流程**

（1）按照用户及时间聚合，计算每个人每天的低碳量（2017年）并给每一条数据打标签（同一个用户不同时间排序）

select user\_id,data\_dt,

sum(low\_carbon) low\_carbon\_sum,

row\_number() over(partition by user\_id order by data\_dt) as rn

from user\_low\_carbon

where substring(data\_dt,1,4)='2017'

group BY user\_id,data\_dt

having low\_carbon\_sum>100;t1

（2）获取每一条数据时间跟标签之间的差值

select user\_id,data\_dt,date\_sub(to\_date(regexp\_replace(data\_dt,"/", "-")),rn) diffDate from t1;

（3）按照所获得的差值聚合，得到同一个用户下相同差值的个数

select user\_id,data\_dt,count(\*) over(partition by user\_id,diffDate) diffDateCount from t2;t3

（4）过滤出相同差值个数在3及以上的数据

select user\_id,data\_dt from t3 where diffDateCount>=3;

（5）与原表Join得到最终需要的结果

select t5.user\_id,t5.data\_dt,t5.low\_carbon

from user\_low\_carbon t5

join t4

where t4.user\_id = t5.user\_id and t4.data\_dt = t5.data\_dt

order by t5.user\_id,t5.data\_dt;

（6）最终Sql

SELECT

t5.user\_id,

t5.data\_dt,

t5.low\_carbon

FROM

user\_low\_carbon t5

JOIN (

SELECT

user\_id,

data\_dt

FROM

(

SELECT

user\_id,

data\_dt,

count(\*) over (

PARTITION BY user\_id,

diffDate

) diffDateCount

FROM

(

SELECT

user\_id,

data\_dt,

date\_sub(

to\_date (

regexp\_replace (data\_dt, "/", "-")

),

rn

) diffDate

FROM

(

SELECT

user\_id,

data\_dt,

sum(low\_carbon) low\_carbon\_sum,

row\_number () over (

PARTITION BY user\_id

ORDER BY

data\_dt

) AS rn

FROM

user\_low\_carbon

WHERE

substring(data\_dt, 1, 4) = '2017'

GROUP BY

user\_id,

data\_dt

HAVING

low\_carbon\_sum > 100

) t1

) t2

) t3

WHERE

diffDateCount >= 3

) t4

WHERE

t4.user\_id = t5.user\_id

AND t4.data\_dt = t5.data\_dt

ORDER BY

t5.user\_id,

t5.data\_dt;

（7）结果展示

+-------------+-------------+----------------+--+

| t5.user\_id | t5.data\_dt | t5.low\_carbon |

+-------------+-------------+----------------+--+

| u\_002 | 2017/1/2 | 150 |

| u\_002 | 2017/1/2 | 70 |

| u\_002 | 2017/1/3 | 30 |

| u\_002 | 2017/1/3 | 80 |

| u\_002 | 2017/1/4 | 150 |

| u\_002 | 2017/1/5 | 101 |

| u\_005 | 2017/1/2 | 50 |

| u\_005 | 2017/1/2 | 80 |

| u\_005 | 2017/1/3 | 180 |

| u\_005 | 2017/1/4 | 180 |

| u\_005 | 2017/1/4 | 10 |

| u\_008 | 2017/1/4 | 260 |

| u\_008 | 2017/1/5 | 360 |

| u\_008 | 2017/1/6 | 160 |

| u\_008 | 2017/1/7 | 60 |

| u\_008 | 2017/1/7 | 60 |

| u\_009 | 2017/1/2 | 70 |

| u\_009 | 2017/1/2 | 70 |

| u\_009 | 2017/1/3 | 170 |

| u\_009 | 2017/1/4 | 270 |

| u\_010 | 2017/1/4 | 90 |

| u\_010 | 2017/1/4 | 80 |

| u\_010 | 2017/1/5 | 90 |

| u\_010 | 2017/1/5 | 90 |

| u\_010 | 2017/1/6 | 190 |

| u\_010 | 2017/1/7 | 90 |

| u\_010 | 2017/1/7 | 90 |

| u\_011 | 2017/1/1 | 110 |

| u\_011 | 2017/1/2 | 100 |

| u\_011 | 2017/1/2 | 100 |

| u\_011 | 2017/1/3 | 120 |

| u\_013 | 2017/1/2 | 150 |

| u\_013 | 2017/1/2 | 50 |

| u\_013 | 2017/1/3 | 150 |

| u\_013 | 2017/1/4 | 550 |

| u\_013 | 2017/1/5 | 350 |

| u\_014 | 2017/1/5 | 250 |

| u\_014 | 2017/1/6 | 120 |

| u\_014 | 2017/1/7 | 270 |

| u\_014 | 2017/1/7 | 20 |

+-------------+-------------+----------------+--+

### 7.8.2 求出连续三天有销售记录的店铺

1） 原始数据

A,2017-10-11,300

A,2017-10-12,200

A,2017-10-13,100

A,2017-10-15,100

A,2017-10-16,300

A,2017-10-17,150

A,2017-10-18,340

A,2017-10-19,360

B,2017-10-11,400

B,2017-10-12,200

B,2017-10-15,600

C,2017-10-11,350

C,2017-10-13,250

C,2017-10-14,300

C,2017-10-15,400

C,2017-10-16,200

D,2017-10-13,500

E,2017-10-14,600

E,2017-10-15,500

D,2017-10-14,600

2） 分析 : 给每个用户一个编号,用日期减去编号,如果是同一天,那么就是连续的

A,2017-10-11,300,1,2017-10-10

A,2017-10-12,200,2,2017-10-10

A,2017-10-13,100,3,2017-10-10

A,2017-10-15,100,4,2017-10-11

A,2017-10-16,300,5,2017-10-11

A,2017-10-17,150,6,2017-10-11

A,2017-10-18,340,7,2017-10-11

A,2017-10-19,360,8,2017-10-11

B,2017-10-11,400

B,2017-10-12,200

B,2017-10-15,600

C,2017-10-11,350

C,2017-10-13,250

C,2017-10-14,300

C,2017-10-15,400

C,2017-10-16,200

D,2017-10-13,500

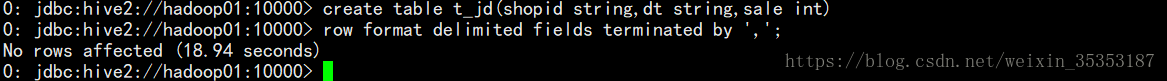
E,2017-10-14,600

E,2017-10-15,500

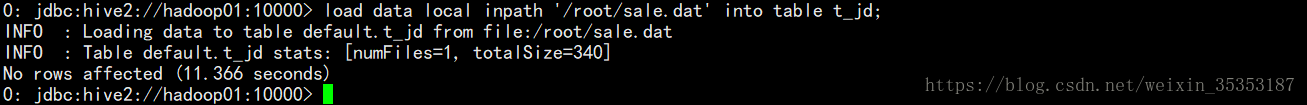
D,2017-10-14,600

**1:建表，加载数据**

create table t\_jd(shopid string,dt string,sale int)  
row format delimited fields terminated by ',';

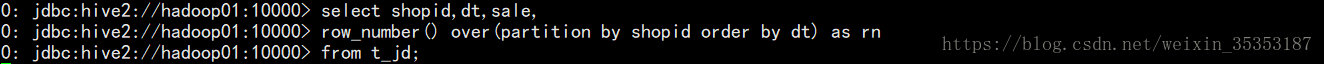


load data local inpath '/root/sale.dat' into table t\_jd;

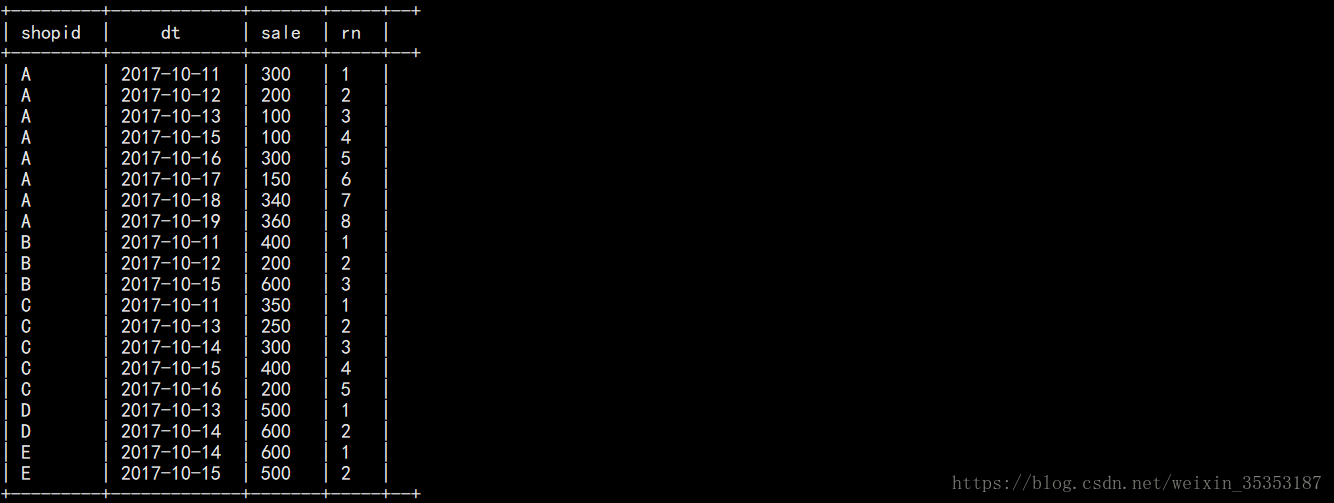


**2:打编号**

select shopid,dt,sale,  
row\_number() over(partition by shopid order by dt) as rn   
from t\_jd;



**结果 :**



**3 根据编号，生成连续日期**

select shopid,dt,sale,rn,

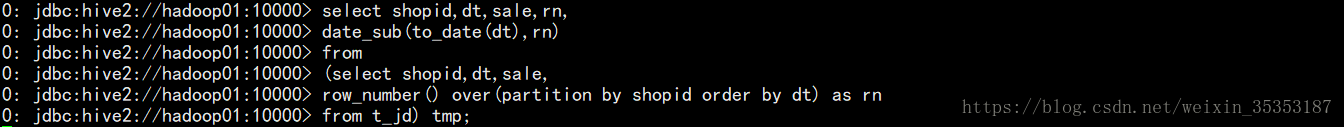
date\_sub(to\_date(dt),rn)

from

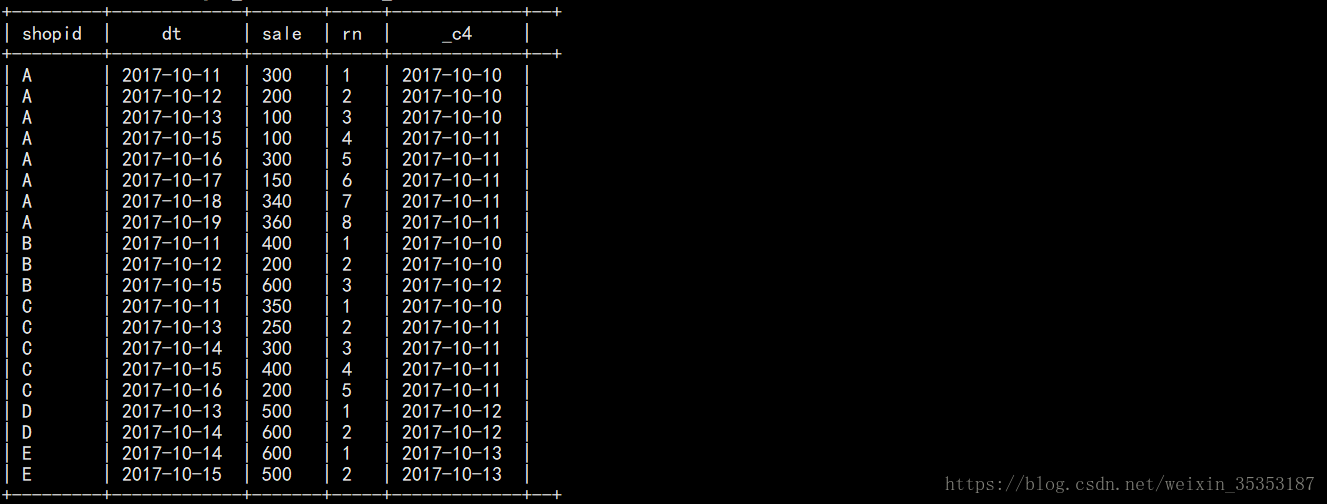
(select shopid,dt,sale,

row\_number() over(partition by shopid order by dt) as rn

from t\_jd) tmp;



**结果 :**



**4 分组，求count**

select shopid,count(1) as cnt

from

(select shopid,dt,sale,rn,

date\_sub(to\_date(dt),rn) as flag

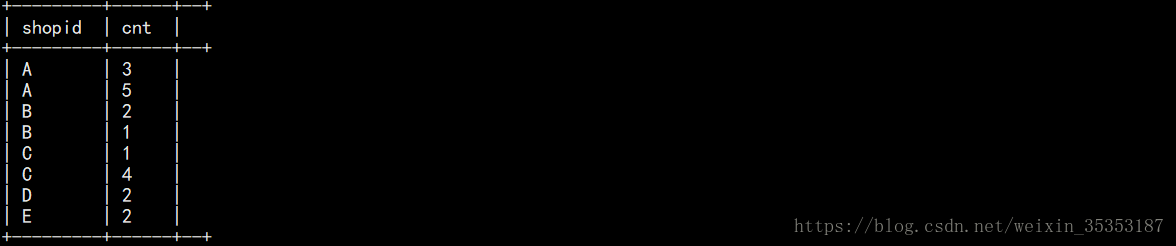
from

(select shopid,dt,sale,row\_number() over(partition by shopid order by dt) as rn

from t\_jd) tmp) tmp2

group by shopid,flag;

**结果 :**



**5 筛选出连续天数大于等于3的**

select shopid from

(select shopid,count(1) as cnt

from

(select shopid,dt,sale,rn,

date\_sub(to\_date(dt),rn) as flag

from

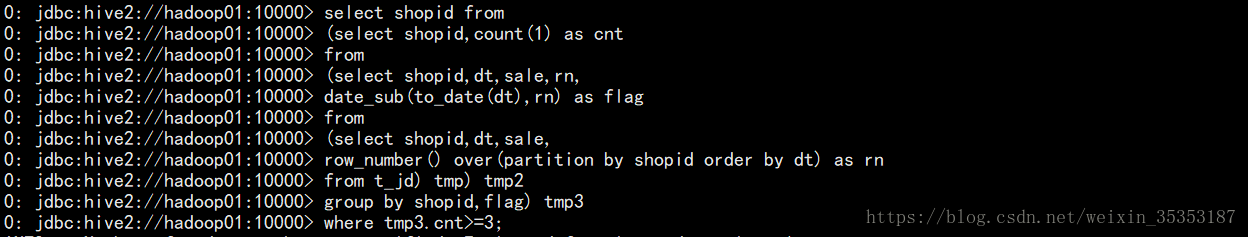
(select shopid,dt,sale,

row\_number() over(partition by shopid order by dt) as rn

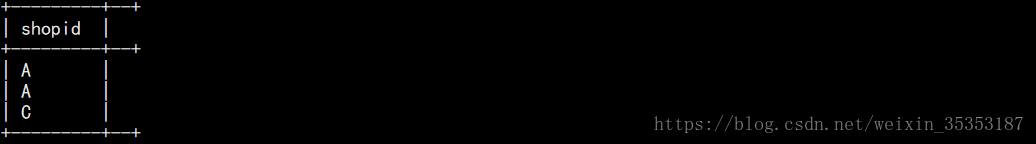
from t\_jd) tmp) tmp2

group by shopid,flag) tmp3

where tmp3.cnt>=3;



**结果 :**



**6 去重**

select distinct shopid from

(select shopid,count(1) as cnt

from

(select shopid,dt,sale,rn,

date\_sub(to\_date(dt),rn) as flag

from

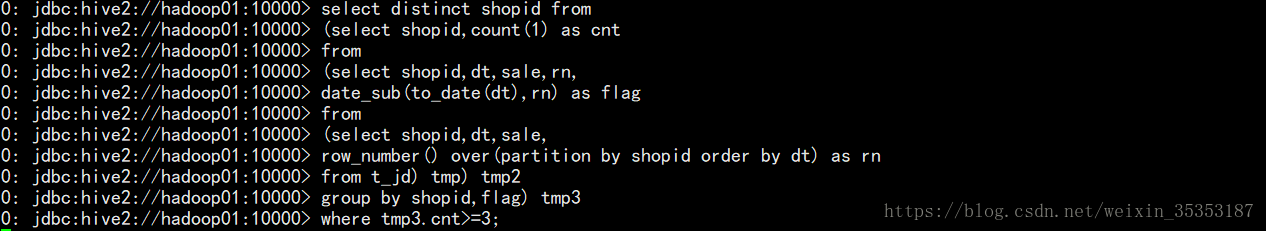
(select shopid,dt,sale,

row\_number() over(partition by shopid order by dt) as rn

from t\_jd) tmp) tmp2

group by shopid,flag) tmp3

where tmp3.cnt>=3;



**结果 :**

