with first\_payments as

(

select user\_id

, date\_trunc('day', transaction\_datetime) as first\_payment\_date

from

(select user\_id

, transaction\_datetime

, row\_number() over(partition by user\_id order by transaction\_datetime) rw\_payment

from skyeng\_db.payments as p

where status\_name = 'success'

group by user\_id, transaction\_datetime

order by date\_trunc('day', transaction\_datetime) desc) as row\_st\_lessons

where rw\_payment = 1

),

all\_dates as

(

select distinct date\_trunc('day', class\_start\_datetime) as dt

from skyeng\_db.classes

where class\_start\_datetime between '2016-01-01' and ' 2017-01-01'

),

all\_dates\_by\_user as

(

select user\_id

, dt

from all\_dates

join first\_payments

on all\_dates.dt >= first\_payments.first\_payment\_date

order by user\_id, dt

),

payments\_by\_dates as

(

select user\_id

, date\_trunc('day', transaction\_datetime) as pay\_day

, sum(classes) as transaction\_balance\_change

from skyeng\_db.payments

where status\_name = 'success'

group by user\_id, pay\_day

order by user\_id, pay\_day

),

payments\_by\_dates\_cumsum as

(

select a.user\_id

, dt

, transaction\_balance\_change

, case when transaction\_balance\_change is null then 0

else transaction\_balance\_change

end as transaction\_balance\_change\_no\_zero

, sum (transaction\_balance\_change) over (partition by a.user\_id order by dt) as transaction\_balance\_change\_cs

, case when sum (transaction\_balance\_change) over (partition by a.user\_id order by dt) is null then 0

else sum (transaction\_balance\_change) over (partition by a.user\_id order by dt)

end as transaction\_balance\_change\_cs\_no\_zero

from all\_dates\_by\_user a

left join payments\_by\_dates b

on a.user\_id = b.user\_id

and a.dt = b.pay\_day

),

classes\_by\_dates as

(

select user\_id

, date\_trunc('day', class\_start\_datetime::date) as class\_date

, count(class\_start\_datetime) \* -1 as classes

from skyeng\_db.classes

where class\_type != 'trial'

and class\_status in ('success', 'failed\_by\_student')

group by user\_id, class\_date

),

classes\_by\_dates\_cumsum as

(

select a.user\_id

, dt

, classes

, case when classes is null then 0

else classes

end as classes\_no\_zero

, sum (classes) over (partition by a.user\_id order by dt) as classes\_cs

, case when sum (classes) over (partition by a.user\_id order by dt) is null then 0

else sum (classes) over (partition by a.user\_id order by dt)

end as classes\_cs\_no\_zero

from all\_dates\_by\_user a

left join classes\_by\_dates b

on a.user\_id = b.user\_id

and a.dt = b.class\_date

),

balances as

(

select a.user\_id

, a.dt

, transaction\_balance\_change\_no\_zero

, transaction\_balance\_change\_cs\_no\_zero

, classes\_no\_zero

, classes\_cs\_no\_zero

, (classes\_cs\_no\_zero + transaction\_balance\_change\_cs\_no\_zero) as balance

from payments\_by\_dates\_cumsum a

left join classes\_by\_dates\_cumsum b

on a.user\_id = b.user\_id

and a.dt = b.dt

)

select dt

, sum (transaction\_balance\_change\_no\_zero) tr\_bal\_change\_sum

, sum (transaction\_balance\_change\_cs\_no\_zero) tr\_bal\_change\_cs\_sum

, sum (classes\_no\_zero) cl\_sum

, sum (classes\_cs\_no\_zero) cl\_cs\_sum

, sum (balance) balance\_sum

from balances

group by dt

order by dt