with first\_payments as

(select user\_id

, min(transaction\_datetime) ::date as first\_payment\_date ---первая дата транзакции без времени

from skyeng\_db.payments

where status\_name='success'

group by user\_id

order by user\_id

),

-------------------конец 1 шага

all\_dates as

(select distinct(class\_start\_datetime ::date) as dt ---уникальные даты уроков без времени

from skyeng\_db.classes

where date\_trunc('year',class\_start\_datetime)='2016-01-01'

group by dt

order by dt

),

-------------------конец 2 шага

all\_dates\_by\_user as

(select user\_id

, dt

from first\_payments a

join all\_dates b

on a.first\_payment\_date <= b.dt ---даты жизни студента после первой транзакции

),

-------------------конец 3 шага

payments\_by\_dates as

(select user\_id

, transaction\_datetime ::date as payment\_date ---дата транзакции без времени

, sum(classes) as transaction\_balance\_change ---сколько уроков списано в этот день

from skyeng\_db.payments

where status\_name='success'

group by user\_id, payment\_date

order by user\_id

),

-------------------конец 4 шага

payments\_by\_dates\_cumsum as

(select c.user\_id

, dt

, transaction\_balance\_change

, sum(coalesce(transaction\_balance\_change, 0)) over (partition by c.user\_id order by dt rows between unbounded preceding and current row) as transaction\_balance\_change\_cs ---кумулятивная сумма по полю transaction\_balance\_change для всех строк до текущ вкл, с разбивкой по user\_id и сортировкой по dt

from all\_dates\_by\_user c

left join payments\_by\_dates d

on c.user\_id=d.user\_id

and c.dt=d.payment\_date

-------------------конец 5 шага

),

classes\_by\_dates as

(select user\_id

, class\_start\_datetime ::date as class\_date ---день урока

, count(id\_class)\*(-1) as classes ---количество пройденных в этот день уроков(минус урок, списание)

from skyeng\_db.classes

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

and class\_type !='trial'

group by user\_id, class\_date

),

-------------------конец 6 шага

classes\_by\_dates\_cumsum as

(select f.user\_id

, dt

, classes

, sum(coalesce(classes, 0)) over (partition by f.user\_id order by dt rows between unbounded preceding and current row) as classes\_cs ---кумулятивная сумма по полю classes для всех строк до текущ вкл, с разбивкой по user\_id и сортировкой по dt

from all\_dates\_by\_user f

left join classes\_by\_dates g

on f.user\_id=g.user\_id

and f.dt=g.class\_date

),

-------------------конец 7 шага

balances as

(select h.user\_id

, h.dt

, transaction\_balance\_change

, transaction\_balance\_change\_cs

, classes

, classes\_cs

, classes\_cs + transaction\_balance\_change\_cs as balance

from payments\_by\_dates\_cumsum h

left join classes\_by\_dates\_cumsum j

on h.user\_id=j.user\_id

and h.dt=j.dt

)

-------------------конец 8 шага

-- select \*

-- from balances

-- order by user\_id, dt

-- limit 1000

-- -----------------конец запроса для Задания 1

select dt

, sum(transaction\_balance\_change) as sum\_transaction\_balance\_change

, sum(transaction\_balance\_change\_cs) as sum\_transaction\_balance\_change\_cs

, sum(classes) as sum\_classes

, sum(classes\_cs) as sum\_classes\_cs

, sum(balance) as sum\_balance

from balances

group by dt

order by dt

-------------------конец 9 шага