with first\_payments as

(select user\_id

, min(transaction\_datetime)::date as first\_payment\_date

from skyeng\_db.payments

where status\_name='success'

group by user\_id),

all\_dates as

(select distinct class\_start\_datetime::date as dt

from skyeng\_db.classes

where date\_part('year', class\_start\_datetime) = 2016),

payments\_by\_dates as

(select user\_id

, transaction\_datetime::date as transaction\_day

, sum(classes) as transaction\_balance\_change

from skyeng\_db.payments

where status\_name='success'

group by user\_id, transaction\_datetime

order by user\_id),

all\_dates\_by\_user as

(select first\_payments.user\_id

, all\_dates.dt

from first\_payments

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

),

classes\_by\_dates as

(select user\_id

, class\_start\_datetime::date as class\_date

, count(id\_class)\*-1 as classes

from skyeng\_db.classes

where class\_status in ('success', 'failed\_by\_student') and class\_type!='trial'

group by user\_id, class\_date

),

payments\_by\_dates\_cumsum as

(select all\_dates\_by\_user.\*

, payments\_by\_dates.transaction\_balance\_change

, sum(case when payments\_by\_dates.transaction\_balance\_change is null then 0 else payments\_by\_dates.transaction\_balance\_change end) over (partition by all\_dates\_by\_user.user\_id order by all\_dates\_by\_user.dt) as transaction\_balance\_change\_cs

from all\_dates\_by\_user

left join payments\_by\_dates on payments\_by\_dates.user\_id=all\_dates\_by\_user.user\_id and all\_dates\_by\_user.dt = payments\_by\_dates.transaction\_day

),

classes\_by\_dates\_dates\_cumsum as

(select all\_dates\_by\_user.\*

, classes\_by\_dates.classes

, sum(case when classes\_by\_dates.classes is null then 0 else classes\_by\_dates.classes end) over (partition by all\_dates\_by\_user.user\_id order by all\_dates\_by\_user.dt) as classes\_cs

from all\_dates\_by\_user

left join classes\_by\_dates on all\_dates\_by\_user.user\_id=classes\_by\_dates.user\_id and all\_dates\_by\_user.dt=classes\_by\_dates.class\_date

),

balances as

(select payments\_by\_dates\_cumsum.\*

, classes\_by\_dates\_dates\_cumsum.classes

, classes\_by\_dates\_dates\_cumsum.classes\_cs

,payments\_by\_dates\_cumsum.transaction\_balance\_change\_cs+classes\_by\_dates\_dates\_cumsum.classes\_cs as balance

from payments\_by\_dates\_cumsum

join classes\_by\_dates\_dates\_cumsum on payments\_by\_dates\_cumsum.user\_id=classes\_by\_dates\_dates\_cumsum.user\_id and payments\_by\_dates\_cumsum.dt=classes\_by\_dates\_dates\_cumsum.dt

)

*-- select \**

*-- from balances*

*-- order by user\_id, dt*

*-- limit 1000*

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

**Вопросы дата-инженерам и владельцам таблицы payments:**

Очень много нулевых значений в колонке «classes» вне зависимости от того, какой статус присвоен транзакции (в колонке status\_name).

**Выводы из получившейся визуализации:**

Начиная с марта месяца количество использованных (списанных) уроков увеличивается и, начиная с мая, при незначительных колебаниях (видимо в зависимости от расписания), растет до конца года.

Количество транзакций подвержено значительным колебаниям в течении всего года, скорее всего можно проследить зависимость резких скачков по количеству оплаченных уроков от рекламных акций и скидок.

Соотношение баланса оплаченных и использованных уроков в течении года растет, но без резких скачков.

Кумулятивные показатели оплаченных и использованных уроков растут также равномерно.