with first\_payments as

(select

user\_id

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

from skyeng\_db.payments

where 1=1

and status\_name = 'success'

and id\_transaction is not null

group by user\_id

) ,

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 '2016-12-31'

),

payments\_by\_dates as

(select

user\_id

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

, sum(classes) as transaction\_balance\_change

from skyeng\_db.payments

where 1=1

and status\_name = 'success'

and id\_transaction is not null

group by

user\_id

, payment\_date

),

all\_dates\_by\_user as

(select

user\_id

, dt

from first\_payments as fp

join all\_dates as ad

on dt >= first\_payments\_date

),

classes\_by\_dates as

(select

user\_id

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

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

from skyeng\_db.classes

where 1=1

and class\_type != 'trial'

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

group by

user\_id

, class\_date

),

payments\_by\_dates\_dates\_cumsum as

(select

adu.user\_id

, dt

, case when transaction\_balance\_change is not null

then transaction\_balance\_change

else '0'

end as transaction\_balance\_change

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

from all\_dates\_by\_user as adu

--join payments\_by\_dates as pbd on adu.user\_id = pbd.user\_id

left join payments\_by\_dates as pbd

on adu.user\_id = pbd.user\_id

and dt = payment\_date

),

classes\_by\_dates\_cumsum as

(select

adu.user\_id

, dt

, case when classes is not null

then classes

else '0'

end as classes

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

from all\_dates\_by\_user as adu

--join classes\_by\_dates as pbd on adu.user\_id = pbd.user\_id

left join classes\_by\_dates as cbd

on adu.user\_id = cbd.user\_id

and dt = class\_date

),

balances as

(select

pbdcs.user\_id

, pbdcs.dt

, transaction\_balance\_change

, transaction\_balance\_change\_cs

, classes

, classes\_cs

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

from payments\_by\_dates\_dates\_cumsum as pbdcs

join classes\_by\_dates\_cumsum as cbdcs

on pbdcs.user\_id = cbdcs.user\_id

and pbdcs.dt = cbdcs.dt

)

--1.2

-- select \*

-- from balances

--1.3 Выберите топ-1000 строк из CTE balances с сортировкой по user\_id и dt. Посмотрите на изменения балансов студентов.

--Какие вопросы стоит задать дата-инженерам и владельцам таблицы payments?

-- select \*

-- from balances

-- order by user\_id, dt

-- limit 1000

--1.4 Посмотрите, как менялось общее количество уроков на балансах студентов.

--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

Какие вопросы стоит задать дата-инженерам и владельцам таблицы payments:

1. Не во всех удачных транзакциях присутствует id;
2. По некоторым неудачным транзакциям все равно начисляются уроки;
3. По таблице из запроса получается, что у некоторых студентов количество уроков на балансе меньше, чем уроков, которые студент прошел.

Общие выводы:

1. Бизнес развивается, люди покупают и проходят уроки;
2. В некоторые дни люди покупают уроки особенно рьяно - возможно причина в маркетинговых акциях, проводимых в это время;
3. К концу года ученики покупают уроков больше, чем проходят.