* **Описание базы данных**

**SKYENG\_DB**

**classes**

***Витрина с уроками***

* + **user\_id** - уникальный идентификатор юзера
  + **id\_class** - уникальный идентификатор урока
  + **class\_start\_datetime** - время начала урока
  + **class\_end\_datetime** - время конца урока
  + **class\_removed\_datetime** - время удаления записи о данном уроке
  + **id\_teacher** - уникальный идентификатор учителя
  + **class\_status** - статус урока (успешно проведен / отменен и тд)
  + **class\_status\_datetime -** время проставления статуса по уроку

**payments**

***Витрина с платежами по урокам***

* + **user\_id** - уникальный идентификатор юзера
  + **id\_transaction** - уникальный идентификатор транзакции
  + **operation\_name** - название проведенной операции
  + **status\_name** - статус проведенной операции (исполнена / не исполнена и тд)
  + **classes** - количество оплаченных уроков
  + **payment\_amount** - выплаченная сумма
  + **transaction\_datetime** - время проведения операции

**students**

***Витрина со списком студентов***

* + **user\_id** - уникальный идентификатор юзера
  + **student\_sex** - пол юзера
  + **geo\_cluster** - географическая агрегация
  + **country\_name** - короткое название страны
  + **region\_name** - название региона
  + **email\_domain** - домен электронной почты

**teachers**

***Витрина со списком учителей***

* + **id\_teacher** - уникальный идентификатор учителя
  + **age** - возраст
  + **city** - город проживания учителя
  + **department** - направление, в котором работает учитель
  + **max\_teaching\_level** - название уровня языка у преподавателя
  + **id\_teaching\_level** - уникальный идентификатор уровня языка у преподавателя
  + **language\_group** - основной язык преподавателя

**Задача.**

Смоделировать изменение балансов студентов - количество уроков, которое есть у каждого студента.

Чтобы проверить, всё ли в порядке с нашими данными и составить список гипотез и вопросов, нам важно понимать:

* Сколько всего уроков было на балансе **всех учеников** за каждый календарный день
* Как это количество менялось под влиянием транзакций (оплат, начислений, корректирующих списаний) и уроков (списаний с баланса по мере прохождения уроков)

Также нам нужно создать таблицу, где будут балансы **каждого студента** за каждый день.

В результате должен получиться запрос, который собирает данные о балансах студентов за каждый прожитый ими день.

with -- 1.1. подзапрос: для каждого студента дата успешной (со статусом success) транзакции.

first\_payments as

(

select user\_id

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

from skyeng\_db.payments

where status\_name = 'success'

group by 1

)

,

all\_dates as -- 1.2. собираем календарь, когда были уроки.

(

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

from skyeng\_db.classes

where class\_status = 'success'

and date\_trunc('year', class\_status\_datetime) = '2016-01-01'

)

,

payments\_by\_dates as -- 1.3. собираем инфо по транзакциям и начисленным/списанным классам по каждому студенту.

(

select user\_id

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

, sum (classes) as transaction\_balance\_change

from skyeng\_db.payments

where status\_name = 'success'

and date\_trunc ('year', transaction\_datetime) = '2016-01-01'

group by 1, 2

order by 1

)

,

all\_dates\_by\_user as -- 2.2 объедининяем `all\_dates` и `first\_payments`, здесь будут храниться все даты жизни студента после того,

( -- как произошла его первая транзакция.

select user\_id

, dt

from all\_dates as a

join first\_payments as b on a.dt >= b.first\_payment\_date

order by 1, 2

)

,

classes\_by\_dates as -- 2.3 количество уроков за каждый день для каждого ученика, только уроки со статусом success и failed\_by\_student

( -- classes умножаю на -1, чтобы отразить, что - — это списания с баланса.

select user\_id

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

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

from skyeng\_db.classes

where (class\_status = 'success' or class\_status ='failed\_by\_student')

and class\_type != 'trial'

group by 1, 2

)

,

balances as -- 3.1 Баланс студента, который сформирован транзакциями и количествами пройденных уроков

(

select c.user\_id

, dt

, row\_number () over (partition by c.user\_id order by dt desc) as rn\_date\_desc

, row\_number () over (partition by c.user\_id order by dt asc) as rn\_date\_asc

, transaction\_balance\_change

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

, classes

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

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

then 0

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

+ (case when sum (classes) over (partition by c.user\_id order by dt asc) is null

then 0

else sum (classes) over (partition by c.user\_id order by dt asc) end) as balance

from all\_dates\_by\_user as c

left join payments\_by\_dates as d on c.user\_id = d.user\_id and c.dt = d.payment\_day

left join classes\_by\_dates as e on c.user\_id= e.user\_id and e.class\_date=c.dt

group by 1, 2, 5, 7

order by 1 asc, 2 desc

)

select dt -- изменения балансов по датам по всем студентам

, sum (transaction\_balance\_change) as payments\_by\_dates

, sum (transaction\_balance\_change\_cs) as payments\_cumulative

, sum (classes) as cnt\_classes

, sum (classes\_cs) as cumulative\_classes

, sum (balance) as total\_daily\_balance

from balances

group by dt

order by dt desc