



Моделирование посуточного баланса уроков студентов онлайн школы “Skyeng”

Верещагин Роман

Формулировка задания

Наша задача — смоделировать изменение балансов студентов.

Баланс — это количество уроков, которое есть у каждого студента.

Чтобы проверить, всё ли в порядке с нашими данными, составьте 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 мы хотим создать таблицу, где будут балансы **каждого студента** за каждый день.



План решения

Шаг 1

Узнаем, когда была первая транзакция для каждого студента. Начиная с этой даты, мы будем собирать его баланс уроков.

Создадим CTE **first_payments** с двумя полями: **user_id** и **first_payment_date** (дата первой успешной транзакции).

Шаг 2

Соберем таблицу с датами за каждый календарный день 2016 года. Есть разные способы это сделать, но мы воспользуемся тем, который уже знаем. Выберем все даты из таблицы **classes**, создадим CTE **all_dates** с полем **dt**, где будут храниться уникальные даты (без времени) уроков.

Шаг 3

Узнаем, за какие даты имеет смысл собирать баланс для каждого студента. Для этого объединим таблицы и создадим CTE **all_dates_by_user**, где будут храниться все даты жизни студента после того, как произошла его первая транзакция. В таблице должны быть такие поля: **user_id**, **dt**.

Шаг 4

Найдем все изменения балансов, связанные с успешными транзакциями. Выберем все транзакции из таблицы `payments`, сгруппируем их по `user_id` и дате транзакции (без времени) и найдем сумму по полю `classes`. В результате получим CTE `payments_by_dates` с полями: `user_id`, `payment_date`, `transaction_balance_change` (сколько уроков было начислено или списано в этот день).

Шаг 5

Найдем баланс студентов, который сформирован только транзакциями. Для этого объединим `all_dates_by_user` и `payments_by_dates` так, чтобы совпадали даты и `user_id`. Используем оконные выражения (функцию sum`), чтобы найти кумулятивную сумму по полю `transaction_balance_change` для всех строк до текущей включительно с разбивкой по `user_id` и сортировкой по `dt`.`

В результате получим CTE `payments_by_dates_cumsum` с полями: `user_id`, `dt`, `transaction_balance_change` — `transaction_balance_change_cs` (кумулятивная сумма по `transaction_balance_change`). При подсчете кумулятивной суммы заменяем пустые значения нулями.`

Шаг 6

Найдем изменения балансов из-за прохождения уроков. Создадим CTE `classes_by_dates`, посчитав в таблице `classes` количество уроков за каждый день для каждого ученика. Нас не интересуют вводные уроки и уроки со статусом, отличным от `success` и `failed_by_student`. Получим результат с такими полями: `user_id`, `class_date`, `classes` (количество пройденных в этот день уроков). Причем `classes` мы умножим на `-1`, чтобы отразить, что `-` — это списания с баланса.

Шаг 7

По аналогии с уже проделанным шагом для оплат создадим CTE для хранения кумулятивной суммы количества пройденных уроков. Для этого объединим таблицы `all_dates_by_user` и `classes_by_dates` так, чтобы совпадали даты и `user_id`. Используем оконные выражения (функцию sum`), чтобы найти кумулятивную сумму по полю `classes` для всех строк до текущей включительно с разбивкой по `user_id` и сортировкой по `dt`. В результате получим CTE classes_by_dates_dates_cumsum` с полями: `user_id`, `dt`, classes — classes_cs (кумулятивная сумма по classes). При подсчете кумулятивной суммы заменяем пустые значения нулями.`

Шаг 8

Создадим CTE `balances` с вычисленными балансами каждого студента. Для этого объединим таблицы `payments_by_dates_cumsum` ****и `classes_by_dates_dates_cumsum` так, чтобы совпадали даты и `user_id`.

Получим такие поля: ** `user_id`, `dt`, `transaction_balance_change`, `transaction_balance_change_cs`, `classes`, `classes_cs`, `balance` (`classes_cs` + `transaction_balance_change_cs`).

Шаг 9

Посмотрим, как менялось общее количество уроков на балансах студентов.

Для этого просуммируем поля `transaction_balance_change`, `transaction_balance_change_cs`, `classes`, `classes_cs`, `balance` из CTE `balances` с группировкой и сортировкой по `dt`

Шаг 10

Выбираем топ-1000 строк из CTE `balances` с сортировкой по `user_id` и `dt`. Смотрим на изменения балансов студентов.

Изучаем полученные данные.

Шаг 11

Создаем визуализацию (линейную диаграмму) итогового результата.

Формулируем выводы из получившейся визуализации?

Создаем визуализацию (линейную диаграмму) итогового результата.

Формулируем выводы из получившейся визуализации?

Итоговый запрос выглядит так

```
with first_payments as
    (select user_id
     ,min(transaction_datetime)::date first_payment_date
    from skyeng_db.payments
    where status_name = 'success'
    group by 1
    ),
all_dates as
    (select distinct date (class_start_datetime) as dt
    from skyeng_db.classes
    where class_start_datetime<'2017-01-01'
    ),
all_dates_by_user as
    (select fp.user_id
     ,ad.dt
    from first_payments as fp
    join all_dates as ad on ad.dt>=fp.first_payment_date
    order by 2
    ),
payments_by_dates as
    (select user_id
     , date (transaction_datetime) as payment_date
     , sum (classes) as transaction_balance_change
    from skyeng_db.payments
    where status_name = 'success'
    group by 1,2
    order by 1,2
    ),
payments_by_dates_cumsum as
    (select abdu.user_id
     ,abdu.dt
     ,pbd.transaction_balance_change
     ,sum (case when transaction_balance_change is not null then transaction_balance_change else 0 end)
         over (partition by abdu.user_id order by abdu.dt) as transaction_balance_change_cs
    from all_dates_by_user as abdu
    left join payments_by_dates as pbd on pbd.user_id=abdu.user_id and pbd.payment_date=abdu.dt
    ),
classes_by_dates as
    (select user_id
     ,date (class_start_datetime) 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 1,2
    order by 1
    ),
classes_by_dates_cumsum as
    (select abdu.user_id
     ,abdu.dt
     ,cbt.classes
     ,sum (case when classes is not null then classes else 0 end) over (partition by abdu.user_id order by abdu.dt) as classes_cs
    from all_dates_by_user as abdu
    left join classes_by_dates as cbt on abdu.dt=cbt.class_date and abdu.user_id=cbt.user_id
    ),
balances as
    (select pcumsum.user_id
     ,pcumsum.dt
     ,pcumsum.transaction_balance_change
     ,pcumsum.transaction_balance_change_cs
     ,ccumsum.classes
     ,ccumsum.classes_cs
     ,ccumsum.classes_cs + pcumsum.transaction_balance_change_cs as balance
    from payments_by_dates_cumsum as pcumsum
    join classes_by_dates_cumsum as ccumsum on pcumsum.user_id=ccumsum.user_id and pcumsum.dt=ccumsum.dt
    -- order by 2
    -- limit 1000
    )
select distinct 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 1
order by 1
```


В результате запроса получается таблица с посуточным балансом отражающая посуточный приход, расход и итоговый баланс уроков школы.

дт	Покупка уроков	Суммарное количество купленных уроков	Использование уроков	Суммарное количество использованных уроков	Итоговый баланс
06.01.2016	11	11		0	11
07.01.2016		11		0	11
08.01.2016	80	91		0	91
09.01.2016	16	107	-1	-1	106
10.01.2016	8	115	-1	-2	113
11.01.2016	32	147	-1	-3	144
12.01.2016	4	151	-6	-9	142
13.01.2016	14	165	-2	-11	154
14.01.2016		165	-4	-15	150
15.01.2016	80	245	-3	-18	227
16.01.2016	9	254		-18	236
17.01.2016	74	328	-4	-22	306
18.01.2016	6	334	-5	-27	307
19.01.2016	38	372	-6	-33	339
20.01.2016	9	381	-8	-41	340
21.01.2016	16	397	-8	-49	348
22.01.2016	9	406	-3	-52	354
23.01.2016	4	410	-3	-55	355
24.01.2016		410	-7	-62	348
25.01.2016		410	-5	-67	343
26.01.2016	20	430	-7	-74	356
27.01.2016	27	457	-9	-83	374
28.01.2016	4	461	-13	-96	365
29.01.2016		461	-6	-102	359
30.01.2016		461	-4	-106	355
31.01.2016	36	497	-4	-110	387
01.02.2016	16	513	-8	-118	395
02.02.2016	18	531	-11	-129	402
03.02.2016	65	596	-13	-142	454
04.02.2016	2	598	-19	-161	437
05.02.2016	12	610	-4	-165	445
06.02.2016	20	630	-8	-173	457
07.02.2016	78	708	-8	-181	527
08.02.2016	32	740	-10	-191	549
09.02.2016	77	817	-14	-205	612

Анализ баланса уроков с группировкой по студентам и дате

Вывод

Изучив данные по таблице (шаг 10) можно увидеть, что у некоторых пользователей встречается отрицательный баланс. Это подразумевает, что занятие со студентом было проведено, но оплаты не было, что невозможно. Отсюда можно сделать вывод, что данные по оплате либо не были внесены в таблицу payments, либо были внесены в неверном формате и при подсчете не учитывались. Эту информацию мы передадим дата-инженерам и владельцам таблицы.

ser_id	dt	transaction_balance_change	transaction_balance_change_cs	classes	classes_cs	balance
1984910	02.06.2016		16	-1	-17	-1
1984910	03.06.2016		16		-17	-1
1984910	04.06.2016		16		-17	-1
1984910	05.06.2016		16		-17	-1
1984910	06.06.2016		16		-17	-1
1984910	07.06.2016		16	-1	-18	-2
1984910	08.06.2016		16		-18	-2
1984910	09.06.2016		16	-1	-19	-3
1984910	10.06.2016		16		-19	-3
1984910	11.06.2016		16		-19	-3
1984910	12.06.2016		16		-19	-3
1984910	13.06.2016		16		-19	-3
1984910	07.07.2016		24	-1	-25	-1
1984910	08.07.2016		24		-25	-1
1984910	09.07.2016		24		-25	-1
1984910	10.07.2016		24		-25	-1
1984910	11.07.2016		24		-25	-1
1984910	12.07.2016		24		-25	-1
1984910	13.07.2016		24		-25	-1
1984910	14.07.2016		24	-1	-26	-2
1984910	15.07.2016		24		-26	-2
1984910	16.07.2016		24		-26	-2
1984910	17.07.2016		24		-26	-2
1984910	18.07.2016		24		-26	-2
1984910	19.07.2016		24	-1	-27	-3
1984910	20.07.2016		24		-27	-3
1984910	21.07.2016		24		-27	-3
1984910	22.07.2016		24		-27	-3
1984910	23.07.2016		24		-27	-3
1984910	24.07.2016		24		-27	-3

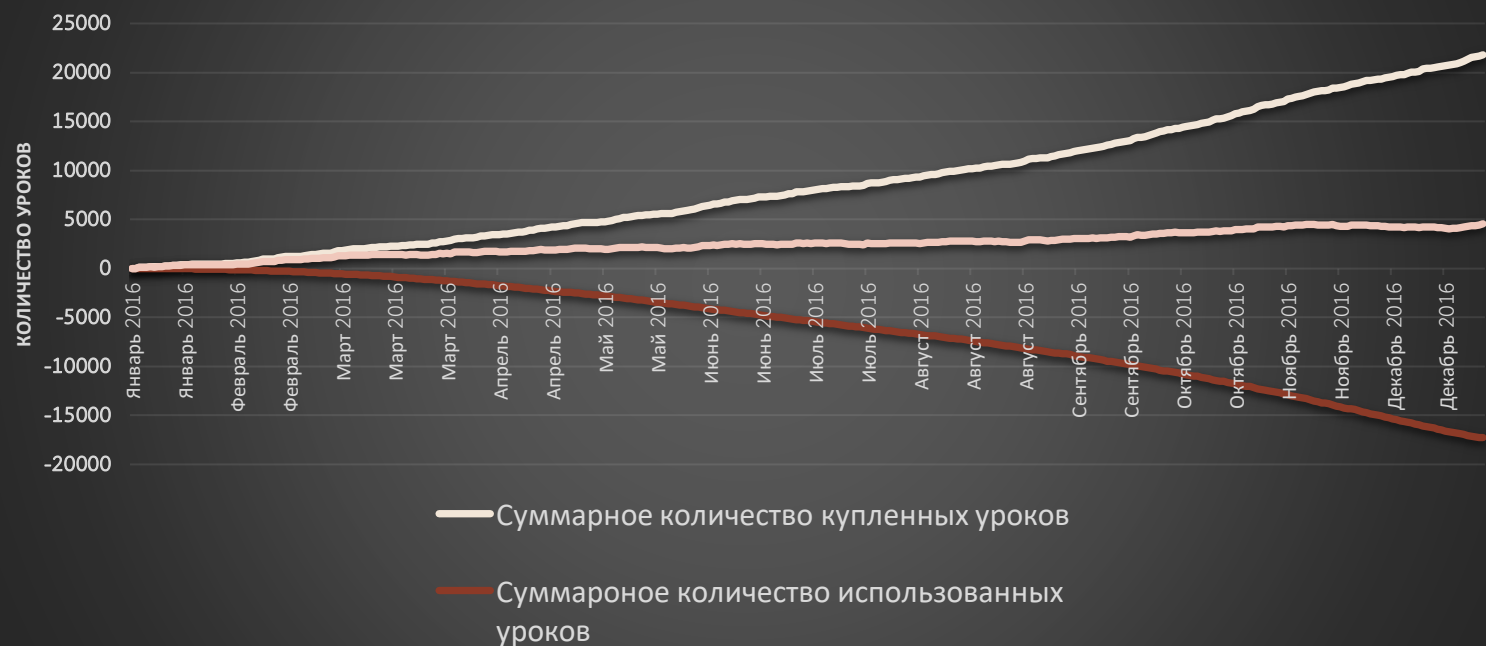
Изучив график по покупке и использованию купленных уроков видно, что наибольшая активность начинается после сентября, что скорее всего связано с началом "учебного года". Эту информацию необходимо использовать для планирования наличия нужного количества преподавателей.

Активность студентов по покупке и использованию уроков



Изучив график суммарного количества купленных и использованных уроков видно, что общее количество приобретенных уроков превышает количество использованных. На конец года у студентов осталось 4534 неиспользованных урока. Эту информацию необходимо передать наставникам и менеджерам для того, что бы они “будили” студентов.

График посуточного баланса уроков.



Спасибо

Верещагин Роман

8917-986-53-65

vereschagin_roman@mail.ru