

Задача 1

Сколько монет в среднем списывает пользователь за весь срок жизни? Сколько монет ему начисляется? Какая в среднем по пользователям разница между начислениями и списаниями?

Примечание: В расчетах “среднего” учитывайте только тех пользователей, у которых в целом были какие-то транзакции.

В расчетах исключайте всегда пользователей с $id < 94$ - это наши внутренние аккаунты. Когда будете работать с таблицей Транзакций, не берите в расчет транзакции больше или равные 500 монетам - это начисления бета-тестерам, они будут сильно мешать.

1. В коде присутствуют пустые строки, это не позволяет на первом шаге проверить код приходится их удалять в ручном режиме. Будьте внимательнее в будущем, чтобы не тратить время друг друга

```
with A as (select sum(t.value) as sum_value,
                t.user_id
            from transaction t
            where t.type_id in(2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,29) --id начислений
            and t.value<=500
            and t.user_id>=94
            group by t.user_id
        ), --начисление по каждому юзеру

A1 as (select round (sum(A.sum_value)/count(A.user_id), 2) as среднее_начисление
        from A), /*вывод среднего начисления */

B as (select round(sum(t.value)*1.0/count(A.user_id ), 2) as среднее_списание
        from A
        left join transaction t on t.user_id =A.user_id
        where t.type_id in(1,23,24,25,26,27,28) --id списаний
        and t.value<=500
        and t.user_id>=94
    ) /*вывод среднего списания */

select A1.среднее_начисление,
       B.среднее_списание,
       A1.среднее_начисление-B.среднее_списание as дельта
from A1, B
```

2. Это мелочи, но при написании комментариев в одной строке нецелесообразно использовать конструкцию `/* */`, это как минимум 4 символа, а `--` всего два и на одной клавише, что существенно экономит время.

3. Точки роста при использовании where

```
...
where t.type_id in(2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,29) --id
начислений
...
where t.type_id in(1,23,24,25,26,27,28) --id списаний
```

удобнее использовать конструкцию between ... and

```
where (t.type_id = 29 or t.type_id between 2 and 22)
...
where (t.type_id = 1 or t.type_id between 23 and 28)
```

4. Использование названий переменных кириллицей не запрещено, но как бы не принято, и в разных системах может вызывать разночтение, лучше использовать только английский алфавит

5. В этом СТЕ вместо громоздкого вычисления можно использовать агрегатную функцию avg, результат тот же, но это компактнее, а главное быстрее.

```
select round (sum(A.sum_value)/count(A.user_id), 2) as среднее_начисление,
            avg(sum_value )
        from A)
```

6. Не сильно понял зачем нужно было «городить» конструкцию с left join. Более того там закралась ошибка суммируются все списания без группировки по пользователям, а деление происходит на всех пользователей. Гораздо проще, а главное правильное по аналогии с СТЕ А посчитать так же всех пользователей и их списания

```
with A as (select sum(t.value) as sum_value,
               t.user_id
            from transaction t
            where t.type_id in(2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,29) --id начислений
            and t.value<=500
            and t.user_id>=94
            group by t.user_id
        ), --начисление по каждому юзеру
A1 as (select round (sum(A.sum_value)/count(A.user_id), 2) as среднее_начисление
        from A), /*вывод среднего начисления */
B as (select round(sum(t.value)*1.0/count(A.user_id ), 2) as среднее_списание
        from A
        ➔ left join transaction t on t.user_id =A.user_id
        where t.type_id in(1,23,24,25,26,27,28) --id списаний
        and t.value<=500
        and t.user_id>=94
    ) /*вывод среднего списания */
```

Вот мой вариант этого фрагмента.

```
with user_write_off as ( -- списания
    select t.user_id, sum(t.value ) as debet_write_off
    from transaction t
    where (t.type_id = 1 or t.type_id between 23 and 28)
        and t.value<=500
        and t.user_id>=94
    group by t.user_id
)
select avg(debet_write_off)
from user_write_off
```

7. Интересное решение со скалярным произведением таблиц, но во-первых это очень нагружает процессор базы данных и лучше использовать такой ход в действительно необходимых ситуациях, во-вторых так как на предыдущем шаге появилась ошибка она попала и в финальную выдачу

```
select A1.среднее_начисление,
       B.среднее_списание,
       A1.среднее_начисление-B.среднее_списание as дельта
from A1, B
```

С учетом указанных моментов, предлагаю вот такой код, кстати обратите внимание на единство подхода с проблемами и отступами

```
with user_accrual as (-- начисления
    select t.user_id,      sum(t.value ) as accrual_sum
    from transaction t
    where (t.type_id = 29 or t.type_id between 2 and 22)
           and t.value<=500
           and t.user_id>=94
    group by t.user_id
),
user_write_off as ( -- списания
    select t.user_id,      sum(t.value ) as write_off_sum
    from transaction t
    where (t.type_id = 1 or t.type_id between 23 and 28)
           and t.value<=500
           and t.user_id>=94
    group by t.user_id
)
select round (avg (accrual_sum), 2) as accrual, -- среднее зачисление
       round (avg(write_off_sum), 2) as write_off, -- среднее списания
       round (avg (accrual_sum), 2) - round (avg(write_off_sum), 2) as diff --разница
from user_accrual ua -- полное объединение таблиц по user id
full outer join user_write_off uw on ua.user_id = uw.user_id
```

Результат

Таблица	123 accrual	123 write_off	123 diff
1	119,69	65,36	54,33

Ваш результат

Таблица	123 среднее_начисление	123 среднее_списание	123 дельта
1	119,69	23,9	95,79