

## Задача 2

Нужно посчитать количество купленных задач и тестов (опираемся на таблицу Транзакций). Интересно, сколько в сумме купили закрытые задачи и тесты (отдельно - задачи, отдельно - тесты), а также в среднем на 1 пользователя платформы (даже если он вообще не проявлял никакой активности) - также отдельно по задачам и тестам. Также стоит посмотреть, сколько людей купили хотя бы 1 задачу/тест, а сколько решали только бесплатные. Дополнительно выведите, сколько людей вообще не решали ни одной задачи, а также не делали ни одной попытки пройти тест.

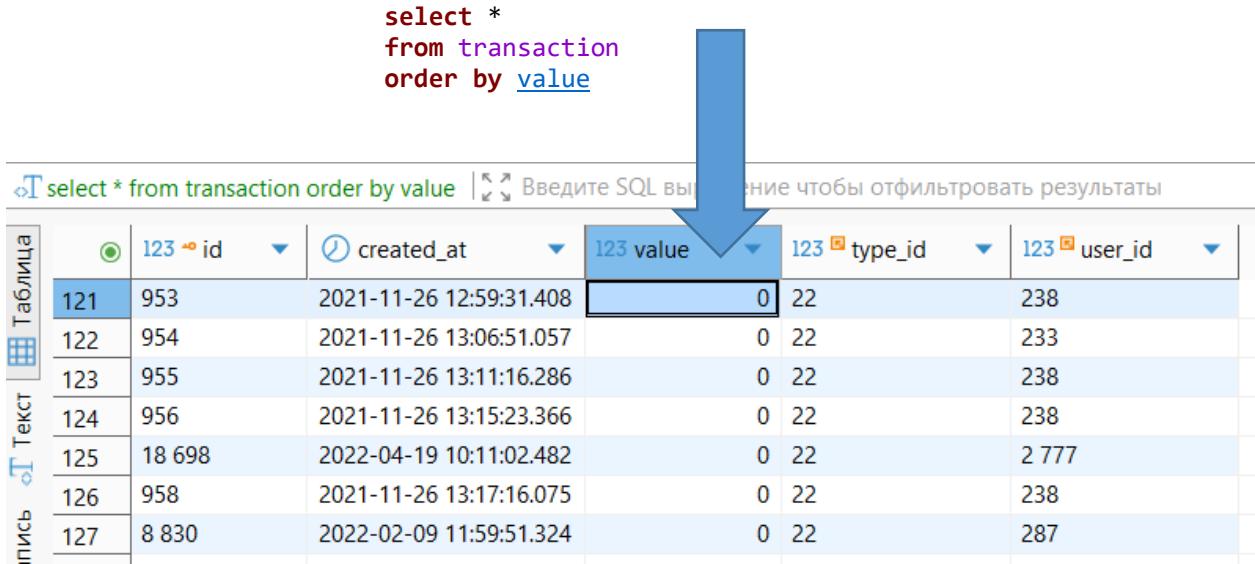
Этой задачи уже нет в симуляторе - она была в старой версии, но мы ее убрали. Поэтому вот небольшие пояснения перед задачей:

За покупку задач в таблице транзакций отвечает type\_id 23, за покупку тестов - 27. Такие задачи считаются закрытыми, и не доступны до покупки. Все остальные задачи (хранящиеся в codesubmit, coderun и problem\_to\_company) - считаем открытыми.

В расчетах исключайте всегда пользователей с id < 94 - это наши внутренние аккаунты. При анализе задач также исключайте из расчета те задачи, которые присутствуют в problem\_to\_company и имеют отношение к компании пользователя, решавшего задачу - это домашние задания для студентов, они тоже искажают статистику.

Например, если у пользователя 10 компания и он решает задачу с номером 300, а в таблице problem\_to\_company эта задача фигурирует как домашняя для компании 10 - ее нужно исключить.

1. У вас в целом очень хороший подход к решению и само решение оформлено с СТЕ внятно и читаемо, но есть один момент. Таблица `transaction` попытке транзакции если она не завершена все равно хранит информацию о пользователе, но при этом значение `value` не NULL, а 0. Следовательно эта запись будет учитываться при использовании  
`count( t.id)`, --кол-во открытых тестов / задач  
`count(distinct t.user_id)`, ---кол-во людей купивших хотя бы 1 тест / задачу



```
select *
from transaction
order by value
```

Введите SQL выражение чтобы отфильтровать результаты

Таблица	id	created_at	value	type_id	user_id
	121	2021-11-26 12:59:31.408	0	22	238
	122	2021-11-26 13:06:51.057	0	22	233
	123	2021-11-26 13:11:16.286	0	22	238
	124	2021-11-26 13:15:23.366	0	22	238
	125	18 698	2022-04-19 10:11:02.482	0	2 777
	126	958	2021-11-26 13:17:16.075	0	22
	127	8 830	2022-02-09 11:59:51.324	0	287

с для таблиц А и В необходимо добавить еще одно условие, оно существенно изменит результат выдачи

```
...
from transaction t
where t.type_id =27 --покупка теста
and t.user_id>=94
and value >0
```

2. Еще одно замечание. По условию необходимо:

«Нужно посчитать 1. количество купленных задач и тестов (опираемся на таблицу Транзакций). Интересно, сколько в сумме купили закрытые задачи и тесты (отдельно - задачи, отдельно - тесты), а также в 2. среднем на 1 пользователя платформы (даже если он вообще не проявлял никакой активности) - также отдельно по задачам и тестам. Также стоит посмотреть, 3. сколько людей купили хотя бы 1 задачу/тест, а 4. сколько решали только бесплатные. Дополнительно выведите, сколько людей 5. вообще не решали ни одной задачи, а также не делали ни одной попытки пройти тест.»

Следовательно вывести информацию необходимо именно в этой последовательности, то есть изменив порядок в `select`

3. Не нашел в Вашем решении ответ на 5 метрику – количество неактивных пользователей. Это можно посчитать используя Ваш же прием – все уникальные пользователи из `transaction` кроме `t.user_id>=94` за минусом тех кто решал хотя бы 1 тест вообще. А так как этот подзапрос уже был, то логично для оптимизации его убрать из подзапроса и сделать отдельным СТЕ

С учетом всего сказанного решение для тестов выглядит следующим образом:

```

with cnt_unique_user as (
    select count(distinct user_id) -- кол-во пользователей вообще (по таблице transaction)
    from transaction -- в teststart, codesubmit, coderun только те, кто запускал и отправлял на проверку
    where user_id>=94 -- их использовать нельзя
),
cnt_work_test_user as ( --кол-во людей решавших хотя бы 1 тест вообще
    select count(distinct user_id)
    from teststart
    where user_id>=94
),
test_statictics as (
    select
        count(*) as count_closed, --количество купленных закрытых тестов
        round(count(t.id)*1.0/count(distinct t.user_id),2) as avg_count_closed, --среднее кол-во купленных тестов на 1 пользователя
        count(distinct t.user_id) as count_user, --кол-во людей купивших хотя бы 1 тест
        (select * from cnt_work_test_user) - count(distinct t.user_id) as count_user, -- кол-во людей, проходившие только бесплатные тесты
        (select * from cnt_unique_user) - (select * from cnt_work_test_user) --кол-во людей решавших хотя бы 1 тест вообще
        from transaction t
        where t.type_id =27 --покупка теста
        and t.user_id>=94
        and value >0
)
select * from test_statictics

```

Уверен, что Вы учтете мои пожелания и статистику по задачам сделаете и оформите по аналогии, кстати !!!

Не забудьте (сейчас этого в Вашем коде нет) что пользователь мог запускать задачу, но не отправлять на проверку или наоборот сразу отправлять на проверку, но при этом не запускать), поэтому логично ссоединить таблицы:

codesubmit, coderun  
через union и уже потом работать с полученной конструкцией.

И последний момент, не забывайте про это условие. Сейчас этого тоже нет !

*«При анализе задач также исключайте из расчета те задачи, которые присутствуют в problem\_to\_company и имеют отношение к компании пользователя, решавшего задачу - это домашние задания для студентов, они тожеискажают статистику.*

*«Например, если у пользователя 10 компаний и он решает задачу с номером 300, а в таблице problem\_to\_company эта задача фигурирует как домашняя для компании 10 - ее нужно исключить.»*