Условия:

1) Условится на том, что SUBS\_ID - это id клиента

2) Нам нужны записи только тех клиентов, которые хотя бы раз его изменяли, то есть имеют равно или более 2 записей (в ином случае нужно использовать RIGHT JOIN вместо INNER JOIN, что менее эффективно)

SELECT t1.subs\_id AS SUBS\_ID, MAX(t1.balance\_date) AS BALANCE\_DATE\_CURR,t1.balance\_amt AS BALANCE\_AMT\_CURR, MAX(t2.max) AS BALANCE\_DATE\_1, t2.balance\_amt AS BALANCE\_AMT\_1 FROM test AS t1 INNER JOIN(

SELECT subs\_id,MAX(balance\_date),balance\_amt FROM test GROUP BY user\_id, balance\_amt

) AS t2 ON t1.subs\_id = t2.subs\_id AND t1.balance\_date != t2.max GROUP BY t1.subs\_id, t1.balance\_amt, t2.balance\_amt

Написал запрос именно таким образом, так как MAX() отбрасывает все NULL, а те данные которые находятся без даты в базе уже некорректные(CONSTRAINT NOT NULL для даты не объявлен )

Использовал INNER JOIN, так как меня попросили собрать статистику по предыдущей и последней операции, то есть все данные с одной операцией отбрасываются с помощью него

Для ускорения работы запроса я бы создал индекс для column balance\_date, так как он используется в предложении INNER JOIN, из-за чего база обращается к индексу.

Также хочу заметить, что баланс клиента может менятся лишь раз в день, что не дает большой нагрузки при добавлении данных из-за индекса.

Также протестировал разные кейсы в табличке test с 2 миллионами записей в ней, по плану выполнения запроса увидел, что индекс при его создании используется и отрабатывает с гораздо большей эффективностью. (запрос отрабатывает за 2 секунды)