Maksimilian Golovach J4133c

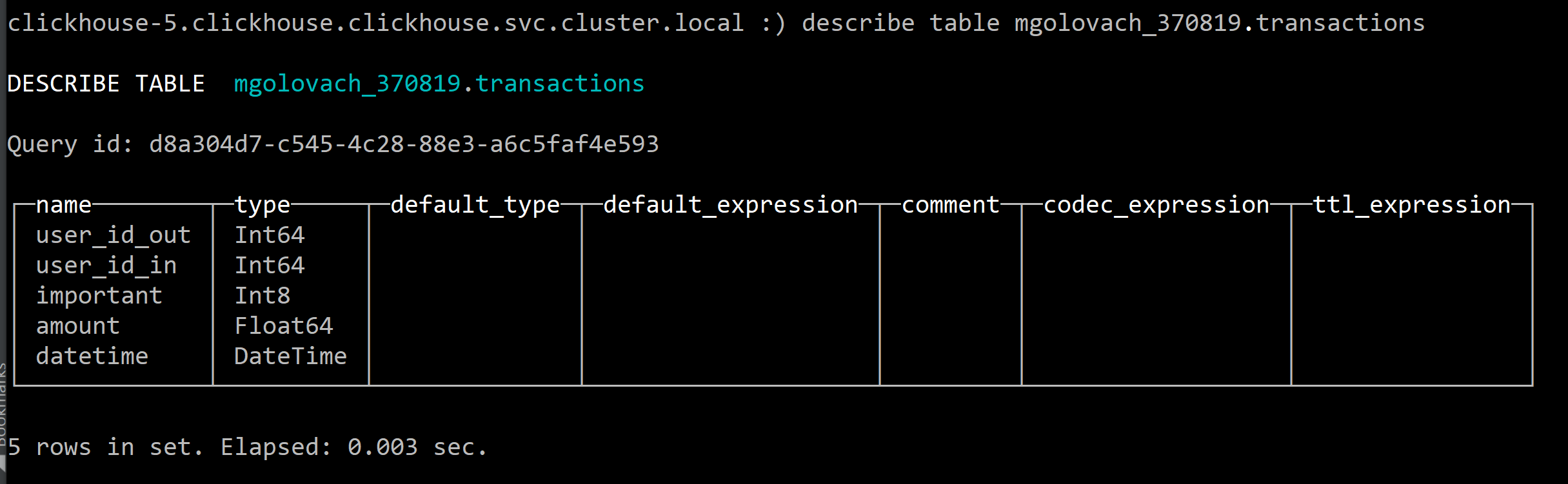
**Schemas**

**Transactions**:

mgolovach\_370819.transactions

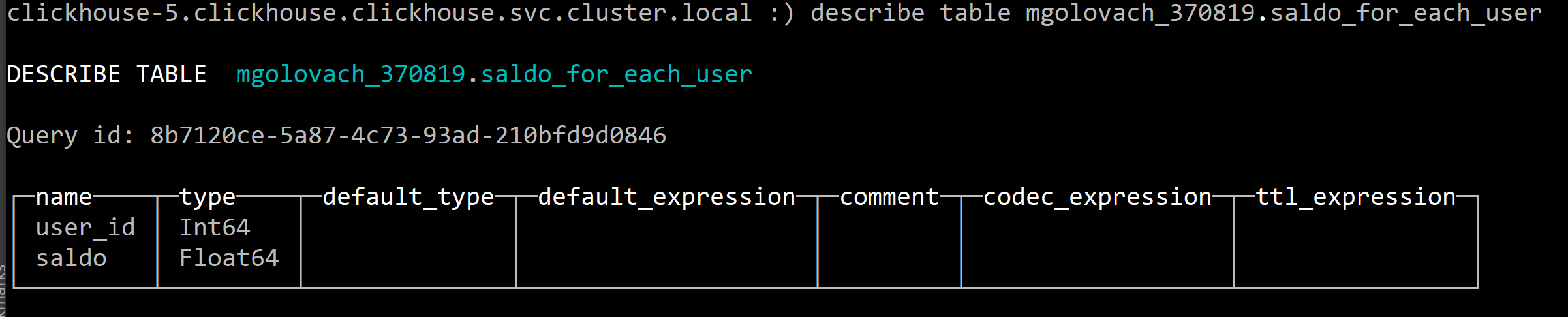
and

mgolovach\_370819.distr\_transactions



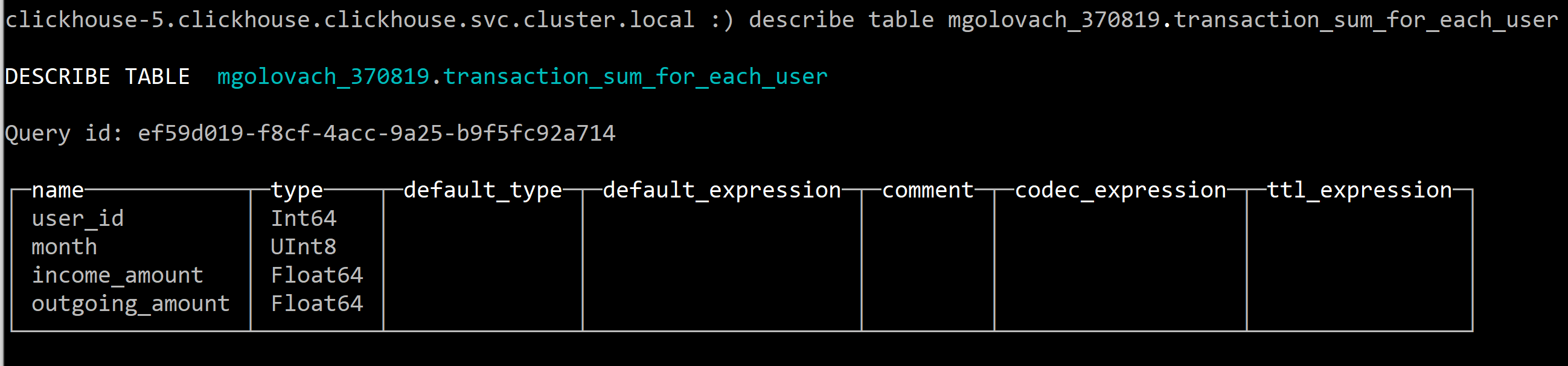
**Users saldo for the current moment**:

mgolovach\_370819.saldo\_for\_each\_user



**The sums for incoming and outcoming transactions by months for each user**:

mgolovach\_370819.transaction\_sum\_for\_each\_user



**Sharding expression clarification**

I’ve chosen sharding expression as 32 bit hash of expression “toYYYYMM(datetime)” for shards to be split by years and months of years.

In transaction dataset I saw that there is only 2018 year, so dataset will be split by 12 shards, 1 shard for each month in this case, so transactions will be +- equally distributed cluster, ~1 million per clickhouse-N.clickhouse.

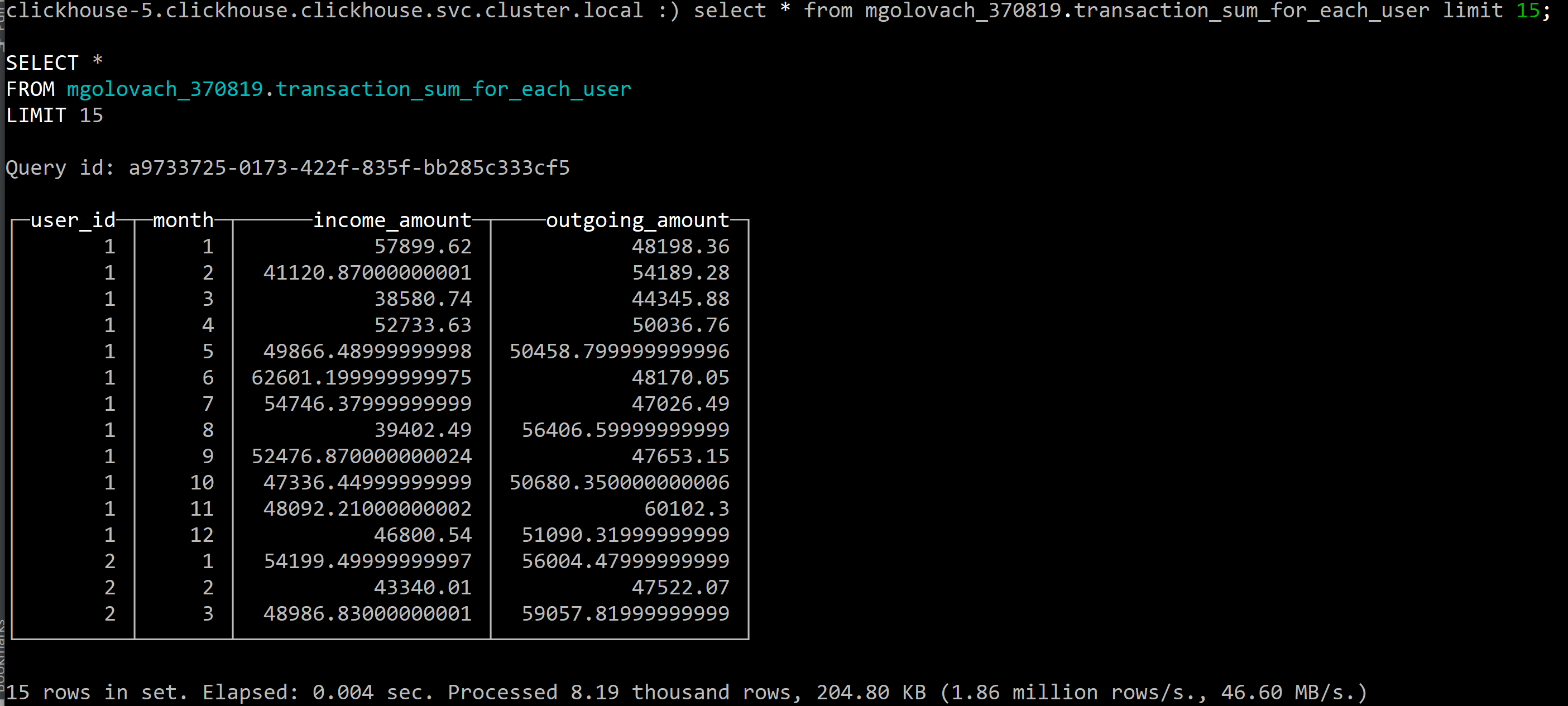
**MV’s choice**

I’ve chosen 3, 4 views:

3 – **The sums for incoming and outcoming transactions by months for each user.**

-- The sums for incoming and outcoming transactions by months for each user.  
-- 1 - getting incomes(out), outcomes(in\_table),  
-- 2 - matching out, in\_table by user ids,  
-- 3 - grouping by user id and month, getting sum of amounts  
CREATE MATERIALIZED VIEW mgolovach\_370819.transaction\_sum\_for\_each\_user ON CLUSTER kube\_clickhouse\_cluster  
 ENGINE = *AggregatingMergeTree* ORDER BY (user\_id, month) POPULATE  
AS  
select  
 out.user\_id\_out as user\_id,  
 out.month as month,  
 *sum*(out.amount) as income\_amount,  
 *sum*(in\_table.amount) as outgoing\_amount  
from (select out.user\_id\_out,  
 *toMonth*(out.datetime) as month,  
 *count*(out.user\_id\_out) as incoming,  
 *sum*(out.amount) as amount  
-- from mgolovach\_370819.transactions out  
 from mgolovach\_370819.distr\_transactions out  
 group by out.user\_id\_out, *toMonth*(out.datetime)) out inner join  
 (select in\_table.user\_id\_in,  
 *toMonth*(in\_table.datetime) as month,  
 *count*(in\_table.user\_id\_in) as outcoming,  
 *sum*(in\_table.amount) as amount  
-- from mgolovach\_370819.transactions in\_table  
 from mgolovach\_370819.distr\_transactions in\_table  
 group by in\_table.user\_id\_in, *toMonth*(in\_table.datetime)) in\_table  
 on user\_id\_in = user\_id\_out and out.month = in\_table.month  
group by out.user\_id\_out, out.month;

Output:



4 - **Users saldo for the current moment.**

-- Users saldo for the current moment.  
CREATE MATERIALIZED VIEW mgolovach\_370819.saldo\_for\_each\_user  
 ON CLUSTER kube\_clickhouse\_cluster  
 ENGINE = *AggregatingMergeTree* ORDER BY (user\_id, saldo) POPULATE  
AS  
select out.user\_id\_out as user\_id, (out.amount - in\_table.amount) as saldo  
from (select out.user\_id\_out,  
 *count*(out.user\_id\_out) as incoming,  
 *sum*(out.amount) as amount  
 from mgolovach\_370819.transactions out  
 group by out.user\_id\_out) out inner join  
 (select in\_table.user\_id\_in,  
 *count*(in\_table.user\_id\_in) as outcoming,  
 *sum*(in\_table.amount) as amount  
 from mgolovach\_370819.transactions in\_table  
 group by in\_table.user\_id\_in) in\_table  
 on user\_id\_in = user\_id\_out;

Output:

