

## ОПИСАНИЕ ЗАДАЧИ

/\* В качестве ДЗ сделаем карту поведения пользователей.

- \* Мы обсуждали, что всех пользователей можно разделить, к примеру, на
- \* New (совершили только 1 покупку),
- \* Regular (совершили 2 или более на сумму не более столько-то),
- \* Vip (совершили дорогие покупки и достаточно часто),
- \* Lost (раньше покупали хотя бы раз и с даты последней покупки прошло больше 3 месяцев).
- \* Вся база должна войти в эти группы (т.е. каждый пользователь должен попадать только в одну из этих групп).

Задача:

1. Уточнить критерии групп New, Regular, Vip, Lost
2. По состоянию на 1.01.2017 понимаем, кто попадает в какую группу, подсчитываем кол-во пользователей в каждой.
3. По состоянию на 1.02.2017 понимаем, кто вышел из каждой из групп, а кто вошел.
4. Аналогично смотрим состояние на 1.03.2017, понимаем кто вышел из каждой из групп, а кто вошел.
5. В итоге делаем вывод, какая группа уменьшается, какая увеличивается и продумываем, в чем может быть причина

## SQL-запрос

```
SET @last_date := '2017/03/01'; -- (select MAX(o_date) FROM orders_20190822);
SET @totalCC := (select SUM(price) FROM orders_20190822);
```

```
SET @R3 := 30;
SET @R2 := 60;
SET @F1 := 1;
SET @F2 := 2;
SET @M1 := 5000000;
SET @M2 := 100000000;
```

WITH

```
R as (
    SELECT R0.user_id, mdate, DATEDIFF(@last_date,R0.mdate) days
    FROM (SELECT user_id, MAX(o_date) mdate FROM orders_20190822 where o_date < (@last_date) GROUP BY user_id) R0 ),
```

```
F as (
    SELECT user_id, COUNT(id_o) c FROM orders_20190822 where o_date < (@last_date) GROUP BY user_id ),
```

```
M as (
    SELECT user_id, SUM(price) m FROM orders_20190822 where o_date < (@last_date) GROUP BY user_id),
```

```
RFM as (
    SELECT
        R.user_id,
        R.days R,
        F.c F,
        M.m M,
        CASE WHEN R.days <= @R3 THEN 3
              WHEN R.days <= @R2 THEN 2
        ELSE 1 END as R1,
        CASE WHEN F.c <= @F1 THEN 1
              WHEN F.c <= @F2 THEN 2
        ELSE 3 END as F1,
        CASE WHEN M.m <= @M1 THEN 1
              WHEN M.m <= @M2 THEN 2
        ELSE 3 END as M1
    FROM R
    inner join F on R.user_id = F.user_id
    inner join M on R.user_id = M.user_id),
```

USERGROUP as (/\*ВВОДИМ ГРУППИРОВКУ\*/

```
SELECT
    user_id,
    R,
    F,
    M,
    CONCAT(R1,F1,M1) RFM,
    CASE WHEN CONCAT(R1,F1,M1) IN ('333','332', '322') THEN 'VIP'
          WHEN CONCAT(R1,F1,M1) IN ('313','212', '111','112','113','211','213','311','312') THEN 'NEW'
          WHEN CONCAT(R1,F1,M1) LIKE '1%' THEN 'LOST'
    ELSE 'REGULAR' END as GR
FROM RFM)
```

```
SELECT
    GR UserGroup,
    COUNT(user_id) Users_in_Group,
    SUM(M) Sum_in_Group,
    cast(100*SUM(M)/@totalCC as decimal(3,0)) Percent_of_TotalCC
```

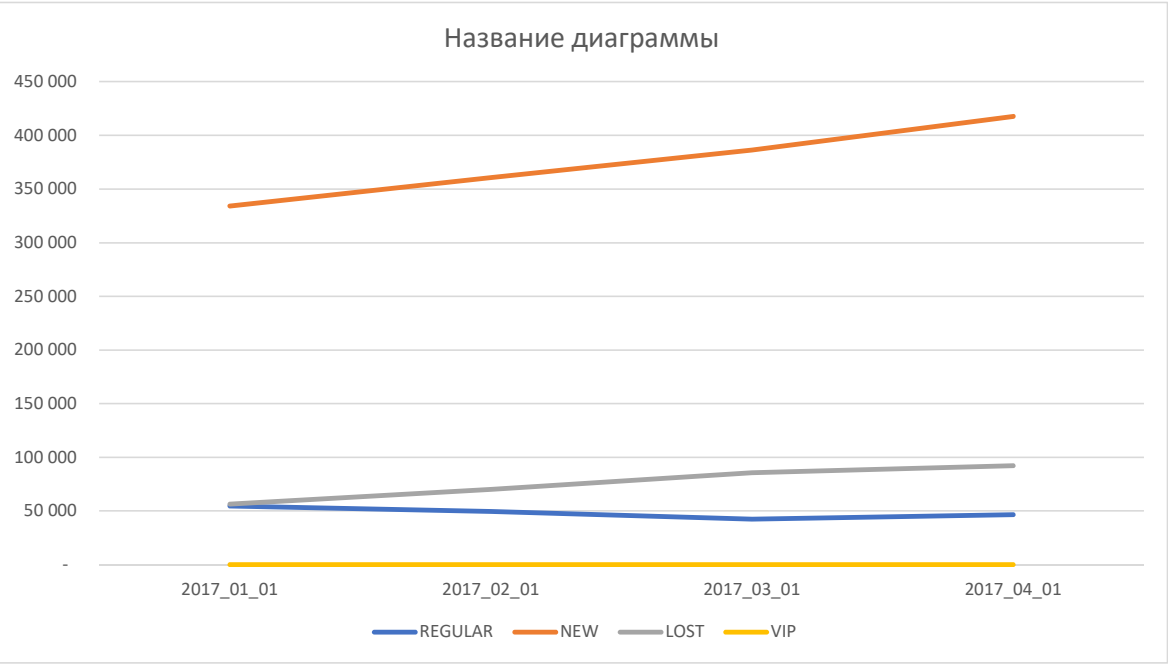
```
FROM USERGROUP
GROUP BY GR
```

## Анализ поведения пользователей

Трижды используем запрос, изменяя параметр @last\_date, для построения сводной таблицы, отражающей динамику в группах

Users in Group on date				
GroupName	2017_01_01	2017_02_01	2017_03_01	2017_04_01
REGULAR	54 604	49758	42394	46537
NEW	334 087	360832	386411	417650
LOST	56 400	70240	85753	92092
VIP	1	2	2	2
Total	445 092	480 832	514 560	556 281

Построим графики для наглядности



Как видно из графиков, происходит небольшое снижение в группе REGULAR на фоне прироста в группах NEW и LOST. Снижение в REGULAR в пределах небольших отклонений NEW и LOST практически параллельны. Это говорит нам о том, что их поведение возможно типично -- новые периодически прихс старые же, насытившись, постепенно перестают покупать