

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.

```
-- Делаем 3 временных таблицы для заказов
-- до 01.01.17, 01.02.17, 01.03.17
USE orders
CREATE TABLE orders_010117
SELECT *
  FROM orders.orders_
  WHERE o_date < date('2017-01-01')

USE orders
CREATE TABLE orders_010217
SELECT *
  FROM orders.orders_
  WHERE o_date < date('2017-02-01')

USE orders
CREATE TABLE orders_010317
SELECT *
  FROM orders.orders_
  WHERE o_date < date('2017-03-01')
```

```
-- создаем три временных таблицы с разбиением пользователей на группы
-- тут пример для группы до 01.03.17
-- для других групп меняем таблицу и дату
USE orders
CREATE TABLE user_groups_010317
SELECT user_id, user_group
  FROM
  (
  SELECT user_id,r,f,m,s,
    CASE WHEN (r = 3 AND f =3 AND m = 3) OR (r = 2 AND f =3 AND m = 3) THEN "Vip"
      WHEN f = 1 AND r = 3 THEN "New"
      WHEN r = 1 THEN "Lost"
    ELSE "Regular" END AS user_group
  FROM
  (
  SELECT user_id,SUM(price) s,
    CASE WHEN TIMESTAMPTDIFF(DAY,MAX(o_date),date('2017-02-28')) > 60 THEN "1"
      WHEN TIMESTAMPTDIFF(DAY,MAX(o_date),date('2017-02-28')) <= 60 AND TIMESTAMPTDIFF(DAY,o_date,date('2017-02-28')) > 30 THEN "2"
    ELSE "3" END AS r,
    CASE WHEN COUNT(*) <= 1 THEN "1"
      WHEN COUNT(*) > 1 AND COUNT(*) <= 4 THEN "2"
    ELSE "3" END AS f,
    CASE WHEN SUM(price) < 5000 THEN "1"
      WHEN SUM(price) >= 5000 AND SUM(price) < 15000 THEN "2"
    ELSE "3" END AS m
  FROM orders.orders_010317
  GROUP BY user_id
) t_rfm
) t
```

```
-- все пользователи в группах
SELECT COUNT(user_id) c_New
  FROM user_groups_010117
 WHERE user_group = 'New'
UNION ALL
SELECT COUNT(user_id) c_Vip
  FROM user_groups_010117
 WHERE user_group = 'Vip'
UNION ALL
SELECT COUNT(user_id) c_Reg
  FROM user_groups_010117
 WHERE user_group = 'Regular'
UNION ALL
SELECT COUNT(user_id) c_Lost
  FROM user_groups_010117
 WHERE user_group = 'Lost'
UNION ALL
SELECT COUNT(user_id) c_All
  FROM user_groups_010117
```

```
SELECT COUNT(user_id) c_New
  FROM user_groups_010217
 WHERE user_group = 'New'
UNION ALL
SELECT COUNT(user_id) c_Vip
  FROM user_groups_010217
 WHERE user_group = 'Vip'
UNION ALL
SELECT COUNT(user_id) c_Reg
  FROM user_groups_010217
 WHERE user_group = 'Regular'
UNION ALL
SELECT COUNT(user_id) c_Lost
  FROM user_groups_010217
 WHERE user_group = 'Lost'
UNION ALL
SELECT COUNT(user_id) c_All
  FROM user_groups_010217
```

```
SELECT COUNT(user_id) c_New
  FROM user_groups_010317
 WHERE user_group = 'New'
UNION ALL
SELECT COUNT(user_id) c_Vip
  FROM user_groups_010317
 WHERE user_group = 'Vip'
UNION ALL
SELECT COUNT(user_id) c_Reg
  FROM user_groups_010317
 WHERE user_group = 'Regular'
UNION ALL
SELECT COUNT(user_id) c_Lost
  FROM user_groups_010317
 WHERE user_group = 'Lost'
UNION ALL
SELECT COUNT(user_id) c_All
  FROM user_groups_010317
```

Получаем всех пользователей для всех групп.

	01.01.2017	01.02.2017	01.03.2017
New	66392	32298	35352
Vip	9053	9300	9325
Reg	98457	103267	60471
Lost	271190	335967	409412
All	445092	480832	514560

Далее создаем временные таблицы для определения кто попал из какой группы в какую.

```
-- таблицы которые содержат user_id группу пользоавателя до 01.01.17 и до 01.02.17
USE orders
CREATE TABLE u01to02
SELECT u1.user_id,u1.user_group gr1,u2.user_group gr2
  FROM user_groups_010117 u1
 JOIN user_groups_010217 u2
  ON u1.user_id = u2.user_id;

-- таблицы которые содержат user_id группу пользоавателя до 01.02.17 и до 01.03.17
USE orders
CREATE TABLE u02to03
SELECT u1.user_id,u1.user_group gr1,u2.user_group gr2
  FROM user_groups_010217 u1
 JOIN user_groups_010317 u2
  ON u1.user_id = u2.user_id;
```

И теперь можем посмотреть кто откуда вышел и куда вошел с помощью новых таблиц.
С помощью подобных запросов:

```
SELECT COUNT(user_id)
  FROM orders.u02to03
 WHERE gr1 = 'New' AND gr2 = 'Vip'
```

Строки означают группу, из которой пользователи вышли, а столбцы в какую группу попали.

01->02					
	New	Regular	Vip	Lost	All
New	0	64322	13	2057	66392
Regular	0	29827	1255	67375	98457
Vip	0	0	7314	1739	9053
Lost	0	5705	689	264796	271190
02->03					
	New	Regular	Vip	Lost	All
New	4633	27657	8	0	32298
Regular	0	23649	968	78650	103267
Vip	0	0	7402	1898	9300
Lost	0	6178	925	328864	335967

01->02					
	New	Regular	Vip	Lost	All
New	0,00%	96,88%	0,02%	3,10%	100,00%
Regular	0,00%	30,29%	1,27%	68,43%	100,00%
Vip	0,00%	0,00%	80,79%	19,21%	100,00%
Lost	0,00%	2,10%	0,25%	97,64%	100,00%
02->03					
	New	Regular	Vip	Lost	All
New	14,34%	85,63%	0,02%	0,00%	100,00%
Regular	0,00%	22,90%	0,94%	76,16%	100,00%
Vip	0,00%	0,00%	79,59%	20,41%	100,00%
Lost	0,00%	1,84%	0,28%	97,89%	100,00%

Получаем примерно одинаковые соотношения входа и выхода для групп.

Пользователи находившиеся до 01.02.17 в New еще не успели попасть в Lost до 01.03.17 , потому что в феврале меньше 30 дней.

Можно заметить, что более 85% купивших один раз покупают снова.

Также можно наблюдать отток Regular, составляющий более 70%, почти весь уходящий в Lost.

Отток Vip составляет около 20% с переходом в Lost.

Выход из Lost явление крайне редкое и составляет около 2%.

Переходы из New в Vip крайне малы и составляют 0.2%.

Переходы в New из других групп невозможны по определению, также невозможен переход из Vip в Regular.

Показатели перехода групп до 01 месяца в состояние до 02 месяца скорее всего завышены по причинам праздников, поэтому прогнозы оттока в группах лучше делать по показателям следующего месяца. По-хорошему, наверное имеет смысл сделать подобные расчеты на весь год и при прогнозе учитывать средние значения или прогнозировать конкретный месяц по значениям такого же месяца.