

-- Главная задача: сделать RFM-анализ на основе данных по продажам за 2 года (из предыдущего дз).

-- Что делаем:

-- 1. Определяем критерии для каждой буквы R, F, M (т.е. к примеру, R – 3 для клиентов, которые покупали <= 30 дней от последней даты в базе, R – 2 для клиентов, которые покупали > 30 и менее 60 дней от последней даты в базе и т.д.)

-- 2. Для каждого пользователя получаем набор из 3 цифр (от 111 до 333, где 333 – самые классные пользователи)

-- 3. Вводим группировку, к примеру, 333 и 233 – это Vip, 1XX – это Lost, остальные Regular (можете ввести более глубокую сегментацию)

-- 4. Для каждой группы из п. 3 находим кол-во пользователей, кот. попали в них и % товарооборота, которое они сделали на эти 2 года.

-- 5. Проверяем, что общее кол-во пользователей бьется с суммой кол-во пользователей по группам из п. 3 (если у вас есть логические ошибки в создании групп, у вас не сойдутся цифры). То же самое делаем и по деньгам.

-- 6. Результаты присылаем.

-- Recency Frequency Monetary анализ

DROP VIEW IF EXISTS r_f_n;

CREATE VIEW r_f_n AS SELECT

```
    user_id,
    CONCAT(Recency, Frequency, Monetary) +0 RFM
from (
    select
        user_id,
        case
            -- допустим, что анализ проводим на начало следующего года:
            when DATEDIFF('2018-01-01', max_o_date) <= 30 then 3
            when DATEDIFF('2018-01-01', max_o_date) > 30
                 and DATEDIFF('2018-01-01', max_o_date) <= 60 then 2
            when DATEDIFF('2018-01-01', max_o_date) > 60 then 1
        end Recency,
        case
            when n >= 5 then 3
            when n >= 2
                 and n < 5 then 2
            when n < 2 then 1
        end Frequency,
        case
            when s > 20000 then 3
            when s > 5000
                 and s <= 20000 then 2
            when s <= 5000 then 1
        end Monetary
    from
        (
            select
                user_id,
                max(o_date) as max_o_date,
                count(id_o) as n,
                sum(price) as s
            from orders_20190822
            group by user_id
        ) as tb1;
) as tb1;
```

```
select * from r_f_n limit 10;
```

	123 user_id T	123 RFM T
1	337 544	121
2	171 642	333
3	260 596	132
4	1 105 609	111
5	982 696	111
6	1 105 614	111
7	1 105 617	111
8	907 785	132
9	527 939	232
10	1 105 621	111

-- Вводим группировку:

```
DROP VIEW IF EXISTS status;
CREATE VIEW status AS SELECT
    user_id,
    case
        when RFM = 233
            or RFM = 333 then 'VIP'
        when RFM < 200 then 'Lost'
        else 'Regular'
    end Status
from r_f_n;

select * from status limit 10;
```

	123 user_id T	ABG Status T
1	337 544	Lost
2	171 642	VIP
3	260 596	Lost
4	1 105 609	Lost
5	982 696	Lost
6	1 105 614	Lost
7	1 105 617	Lost
8	907 785	Lost
9	527 939	Regular
10	1 105 621	Lost

```
-- Для каждой группы находим кол-во пользователей, которые попали в них,
-- и % товарооборота, которое они сделали на эти 2 года.
```

```
select
    Status,
    count(*) 'Кол-во',
    round(sum(percent), 2) '%TO'
from status
join
    -- определим % товарооборот для каждого user_a
    /*можно было его определить в самом начале и протащить через все таблицы и не
    использовать join*/
    (
        select
            user_id,
            sum(price) / (select sum(price) from orders_20190822) * 100 as percent
        from orders_20190822
        group by user_id
    ) as tb
on status.user_id = tb.user_id
group by status;
```

	Status	Кол-во	%TO
1	Lost	790 638	65,75
2	VIP	13 105	16,08
3	Regular	211 376	18,16

```
-- Проверяем, что общее кол-во пользователей бьется с суммой кол-во пользователей по группам.
```

```
select
    count(distinct user_id) 'Кол-во',
    'Исходные данные'
from orders_20190822
union all
select
    count(user_id),
    'Расчитанные данные'
from status;
```

	Кол-во	Исходные данные
1	1 015 119	Исходные данные
2	1 015 119	Расчитанные данные