

# RFM-анализ клиентской базы аптечной сети

Разделим клиентов на данные группы:

## **Recency:**

- (1) Недавние покупатели
- (2) Давние покупатели
- (3) Спящие покупатели

## **Frequency:**

- (1) Частые
- (2) Умеренные
- (3) Редкие

## **Monetary:**

- (1) Большой чек
- (2) Средний чек
- (3) Маленький чек

## RFM SQL Запрос Классификация пользователей

----- MAIN RFM SCRIPT with PERCENTILE\_CONT -----

```
WITH rfm_data AS (
  SELECT
    card,
    MAX(datetime) AS last_purchase_date,
    COUNT(DISTINCT doc_id) AS purchase_frequency,
    SUM(summ) AS total_spent,
    -- Расчет ресенсу как разница между максимальной датой покупки и последней датой покупки
    (SELECT MAX(datetime) FROM bonuscheques) - MAX(datetime) AS recency
  FROM
    bonuscheques
  WHERE
    card ~ '^[0-9]+$' -- Условие для фильтрации карт, состоящих только из цифр
  GROUP BY
    card
),
percentiles AS (
  SELECT
    PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY recency) AS recency_p25,
    PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY recency) AS recency_median,
    PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY recency) AS recency_p75,
    PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY purchase_frequency) AS frequency_median,
    PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY purchase_frequency) AS frequency_p75,
    PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY total_spent) AS monetary_median,
    PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY total_spent) AS monetary_p75
  FROM
    rfm_data
)
SELECT
  r.card,
  -- Конкатенация RFM групп в один столбец
  (CASE
```

```

    WHEN r.recency <= recency_p25 THEN '1' -- Недавние
    WHEN r.recency <= recency_median THEN '2' -- Спящие
    ELSE '3' -- Давние
END ||
',' || -- Разделитель между группами
CASE
    WHEN r.purchase_frequency >= frequency_p75 THEN '1' -- Частые покупатели
    WHEN r.purchase_frequency >= frequency_median THEN '2' -- Умеренные покупатели
    ELSE '3' -- Редкие покупатели
END ||
',' || -- Разделитель между группами
CASE
    WHEN r.total_spent >= monetary_p75 THEN '1' -- Высокий чек
    WHEN r.total_spent >= monetary_median THEN '2' -- Средний чек
    ELSE '3' -- Низкий чек
END) AS rfm_group -- Новый столбец с объединенными значениями
FROM
    rfm_data r,
    percentiles p -- Соединяем с подзапросом для перцентилей
ORDER BY
    rfm_group; -- Сортировка RFM

```

#### ----- RECENCY SCRIPT WITH PERCENTILE\_CONT -----

```

WITH max_date AS (
    SELECT MAX(datetime) AS max_purchase_date
    FROM bonuscheques
),
recency_data AS (
    SELECT
        card,
        (md.max_purchase_date - MAX(b.datetime)) AS recency_days -- Используем MAX(datetime) для расчета
        давности
    FROM
        bonuscheques b,
        max_date md
    WHERE
        b.card ~ '^[0-9]+$' -- Условие для фильтрации карт, состоящих только из цифр
    GROUP BY
        b.card, md.max_purchase_date -- Добавляем max_purchase_date в GROUP BY
),
percentiles AS (
    SELECT
        PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY recency_days) AS p25,
        PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY recency_days) AS median
    FROM
        recency_data
)
SELECT
    r.card,
    r.recency_days,
    CASE
        WHEN r.recency_days <= p.p25 THEN '1' -- Недавние
        WHEN r.recency_days <= p.median THEN '2' -- Спящие
        else '3' -- Давние
    END AS recency_group
FROM
    recency_data r,
    percentiles p
ORDER BY
    r.recency_days; -- Сортировка по кол-ву дней с последней покупки

```

```

----- FREQUENCY SCRIPT PERCENTILE_CONT -----
WITH rfm_data AS (
  SELECT
    card,
    COUNT(*) AS purchase_frequency -- Подсчет количества покупок для каждого клиента
  FROM
    bonuscheques
  WHERE
    card ~ '^[0-9]+$' -- Условие для фильтрации карт, состоящих только из цифр
  GROUP BY
    card
),
percentiles AS (
  SELECT
    PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY purchase_frequency) AS frequency_median,
    PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY purchase_frequency) AS frequency_p75
  FROM
    rfm_data
)
SELECT
  r.card,
  r.purchase_frequency,
  CASE
    WHEN r.purchase_frequency >= p.frequency_p75 THEN '1' -- Частые покупатели
    WHEN r.purchase_frequency >= p.frequency_median THEN '2' -- Умеренные покупатели
    ELSE '3' -- Редкие покупатели
  END AS frequency_group -- Новый столбец с группой Frequency
FROM
  rfm_data r,
  percentiles p -- Соединяем с подзапросом для перцентилей
ORDER BY
  purchase_frequency; -- Сортировка по частоте покупок

```

```

----- MONETARY SCRIPT WITH PERCENTILE_CONT -----
WITH rfm_data AS (
  SELECT
    card,
    SUM(summ) AS total_spent -- Подсчет общего объема расходов для каждого клиента
  FROM
    bonuscheques
  WHERE
    card ~ '^[0-9]+$' -- Условие для фильтрации карт, состоящих только из цифр
  GROUP BY
    card
),
percentiles AS (
  SELECT
    PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY total_spent) AS monetary_median,
    PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY total_spent) AS monetary_p75
  FROM
    rfm_data
)
SELECT
  r.card,
  r.total_spent,
  CASE
    WHEN r.total_spent >= p.monetary_p75 THEN '1' -- Высокие расходы
    WHEN r.total_spent >= p.monetary_median THEN '2' -- Умеренные расходы
    ELSE '3' -- Низкие расходы
  END AS monetary_group -- Новый столбец с группой Monetary
FROM
  rfm_data r,
  percentiles p -- Соединяем с подзапросом для перцентилей

```

```
ORDER BY
total_spent; -- Сортировка по сумме чека
```

Скорректируем диапазоны с помощью диаграммы.

```
WITH max_date AS (
  SELECT MAX(datetime) AS max_purchase_date
  FROM bonuscheques
),
recency_data AS (
  SELECT
    ROUND(EXTRACT(DAY FROM (md.max_purchase_date - MAX(b.datetime))) / 10.0) * 10 AS recency_days -- Округляем
    до ближайшего десятка
  FROM
    bonuscheques b,
    max_date md
  WHERE
    b.card ~ '^[0-9]+$' -- Условие для фильтрации карт, состоящих только из цифр
  GROUP BY
    b.card, md.max_purchase_date -- Убираем card из выборки
),
percentiles AS (
  SELECT
    PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY recency_days) AS p25,
    PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY recency_days) AS median
  FROM
    recency_data
)
SELECT
  r.recency_days,
  CASE
    WHEN r.recency_days <= p.p25 THEN '1' -- Недавние
    WHEN r.recency_days <= p.median THEN '2' -- Спящие
    ELSE '3' -- Давние
  END AS recency_group,
  COUNT(*) AS customer_count -- Подсчет количества клиентов в каждой группе
FROM
  recency_data r,
  percentiles p
GROUP BY
  r.recency_days, recency_group -- Группировка по recency_days и recency_group
ORDER BY
  customer_count; -- Сортировка по recency_days
```

### Recency:

(1) Недавние покупатели <= 20 (Покупка раз в месяц - Нормальный показатель, т.к мы делаем RFM анализ для аптеки).

(2) Давние покупатели <= 80

(3) Спящие покупатели > 80

### Frequency:

(1) Частые >= 11

(2) Умеренные >= 5

(3) Редкие else

### Monetary:

(1) Большой чек >= 3316

(2) Средний чек >= 1477

(3) Маленький чек else

Выполним проверку по одному покупателю.

Возьмем покупателя с номером карты 2000200225706. Он относится к группе 1-1-1.

```
select b.card, ('2022-06-09' - max(date(b.datetime))) as Recency, count(*) as Frequency, sum(summ) as Monetary
from bonuscheques b
where card = '2000200225706'
group by b.card
```

Получим результат:

	A-Z card	123 recency	123 frequency	123 monetary
1	2000200225706	7	14	17 544

Все группы соответствуют диапазонам.

Дополним основной запрос и посмотрим сколько всего получилось групп и какое количество клиентов в каждой группе.

```
WITH rfm_data AS (
    SELECT
        card,
        MAX(datetime) AS last_purchase_date,
        COUNT(DISTINCT doc_id) AS purchase_frequency,
        SUM(summ) AS total_spent,
        -- Расчет recency как разница между максимальной датой покупки и последней датой покупки
        (SELECT MAX(datetime) FROM bonuscheques) - MAX(datetime) AS recency
    FROM
        bonuscheques
    WHERE
        card ~ '^[0-9]+$' -- Условие для фильтрации карт, состоящих только из цифр
    GROUP BY
        card
),
percentiles AS (
    SELECT
        PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY recency) AS recency_p25,
        PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY recency) AS recency_median,
        PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY recency) AS recency_p75,
        PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY purchase_frequency) AS frequency_median,
        PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY purchase_frequency) AS frequency_p75,
        PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY total_spent) AS monetary_median,
        PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY total_spent) AS monetary_p75
    FROM
        rfm_data
),
rfm_groups AS (
    SELECT
        r.card,
        (CASE
            WHEN r.recency <= recency_p25 THEN '1' -- Недавние
            WHEN r.recency <= recency_median THEN '2' -- Спящие
            ELSE '3' -- Давние
        END) ||
        '-' || -- Разделитель между группами
        (CASE
            WHEN r.purchase_frequency >= frequency_p75 THEN '1' -- Частые покупатели
            WHEN r.purchase_frequency >= frequency_median THEN '2' -- Умеренные покупатели
            ELSE '3' -- Редкие покупатели
        END) ||
        '-' || -- Разделитель между группами
        (CASE
            WHEN r.total_spent >= monetary_p75 THEN '1' -- Высокий чек
            WHEN r.total_spent >= monetary_median THEN '2' -- Средний чек
            ELSE '3' -- Низкий чек
        END) AS rfm_group -- Новый столбец с объединенными значениями
    FROM
        rfm_data
)
```

```

    rfm_data r,
    percentiles p -- Соединяем с подзапросом для перцентилей
)
SELECT
    rfm_group,
    COUNT(*) AS group_count -- Подсчет количества групп
FROM
    rfm_groups
GROUP BY
    rfm_group -- Группировка по rfm_group
ORDER BY
    rfm_group; -- Сортировка по rfm_group

```

Разделим все группы на 3 сегмента.

```

WITH rfm_data AS (
    SELECT
        card,
        MAX(datetime) AS last_purchase_date,
        COUNT(DISTINCT doc_id) AS purchase_frequency,
        SUM(summ) AS total_spent,
        -- Расчет ресенсу как разница между максимальной датой покупки и последней датой покупки
        (SELECT MAX(datetime) FROM bonuscheques) - MAX(datetime) AS recency
    FROM
        bonuscheques
    WHERE
        card ~ '^[0-9]+$' -- Условие для фильтрации карт, состоящих только из цифр
    GROUP BY
        card
),
percentiles AS (
    SELECT
        PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY recency) AS recency_p25,
        PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY recency) AS recency_median,
        PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY recency) AS recency_p75,
        PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY purchase_frequency) AS frequency_median,
        PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY purchase_frequency) AS frequency_p75,
        PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY total_spent) AS monetary_median,
        PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY total_spent) AS monetary_p75
    FROM
        rfm_data
),
rfm_groups AS (
    SELECT
        r.card,
        (CASE
            WHEN r.recency <= recency_p25 THEN '1' -- Недавние
            WHEN r.recency <= recency_median THEN '2' -- Спящие
            ELSE '3' -- Давние
        ) AS recency_group,
        (CASE
            WHEN r.purchase_frequency >= frequency_p75 THEN '1' -- Частые покупатели
            WHEN r.purchase_frequency >= frequency_median THEN '2' -- Умеренные покупатели
            ELSE '3' -- Редкие покупатели
        ) AS frequency_group,
        (CASE
            WHEN r.total_spent >= monetary_p75 THEN '1' -- Высокий чек
            WHEN r.total_spent >= monetary_median THEN '2' -- Средний чек
            ELSE '3' -- Низкий чек
        ) AS total_spent_group
    FROM
        rfm_data r
    JOIN
        percentiles p
    ON
        1=1
)

```

```

END) AS rfm_group -- Новый столбец с объединенными значениями
FROM
  rfm_data r,
  percentiles p -- Соединяем с подзапросом для перцентилей
)
SELECT
  rfm_group,
  COUNT(*) AS group_count, -- Подсчет количества групп
  CASE
    WHEN rfm_group IN ('1-1-1') THEN 'VIP'
    WHEN rfm_group IN ('1-1-2', '1-1-3', '1-2-1', '1-2-2', '1-2-3', '1-3-1', '1-3-2', '1-3-3') THEN 'Постоянные'
    WHEN rfm_group IN ('2-3-3', '2-3-2', '2-3-1', '2-2-3', '2-2-2', '2-2-1', '2-1-3', '2-1-1', '2-1-2') THEN 'Спящие'
    WHEN rfm_group IN ('3-3-3', '3-3-2', '3-3-1', '3-2-3', '3-2-2', '3-2-1', '3-1-3', '3-1-2', '3-1-1') THEN 'Уходящие'
  END AS segment -- Новый столбец с сегментами
FROM
  rfm_groups
GROUP BY
  rfm_group -- Группировка по rfm_group
ORDER BY
  rfm_group; -- Сортировка по rfm_group

```

## Уходящие

Сегмент	Характеристика	Что отправляем
333	Потерянные экономные	Скорее всего, эти клиенты уже ушли от нас, и не стоит тратить слишком больших усилий на этот сегмент. Отправляем автоматическую цепочку реактивации.
332	Одноразовые	
331	Одноразовые	
323	Уходящие редкие	Эти клиенты сделали у нас несколько покупок, а потом перестали покупать. Их можно попытаться вернуть и обязательно получить обратную связь, почему они от нас ушли. Отправляем реаквационную рассылку с опросом.
322		
321		
313	Уходящие постоянные	Этих клиентов обязательно нужно попытаться вернуть. Отправляем им реаквационные письма, запрашиваем обратную связь, предлагаем бонусы, программы лояльности.
312	Уходящие хорошие постоянные	
311	Уходящие VIP	

## Спящие

Клиенты, которые покупали не так давно и которых можно вернуть.

Сегмент	Характеристика	Что отправляем
233	Спящие разовые с маленьким чеком	Эти клиенты ещё помнят о нас. Нужно попробовать расшевелить их стимулирующими акциями. Отправляем им рассылки с интересными акциями и выгодными предложениями.
232		
231		
223	Спящие редкие с маленьким чеком	
222	Спящие редкие со средним чеком	
221	Спящие редкие с высоким чеком	Еще не так давно они часто покупали у вас, а сейчас перестали. Нужно обязательно узнать, в чем дел. И попытаться вернуть их. Отправляем им рассылку с интересными акциями и рассылку с опросом.
213	Спящие постоянные с маленьким чеком	
222	Спящие постоянные со средним чеком	
221	Спящие постоянные с высоким чеком	

## Постоянные

Клиенты, которые часто покупают товары или новички, которые только познакомились с компанией.

Сегмент	Характеристика	Что отправляем
111	VIP	Отправляем им особые предложения, например участие в программе лояльности - приглашение в клуб любимых клиентов.
112	Постоянные со средним чеком	Наши самые хорошие клиенты, можно попробовать увеличить сумму чека. Отправляем им специальные предложения, сопутствующие товары.
113	Постоянные с маленьким чеком	Можно попытаться увеличить сумму чека допродажей. Отправляем им рассылки с сопутствующими товарами.
121	Постоянные с высоким чеком	Эти клиенты и так хорошо покупают, не стоит утомлять их лишними рассылками. Отправляем только обычные рассылки.
122	Постоянные со средним чеком	Нужно попытаться увеличить средний чек продаж. Отправляем им рассылки с сопутствующими товарами.
123	Постоянные с маленьким чеком	
131	Новички с высоким чеком. Потенциальные VIP	В дополнение к welcome-цепочке отправляем им интересное предложение, чтобы удержать интерес.
132	Новички со средним чеком	Отправляем welcome-цепочку с описанием преимуществ, ответами на частые вопросы, обучающие материалы.
133	Новички с маленьким чеком	

Далее на основе анализа создадим маркетинговые активности под каждый сегмент. Лучших клиентов мы стараемся удержать, середнячков «раскрутить» до лучших, а уходящих и почти потерянных - вернуть.

Рассказывать про новинки и акции наиболее лояльным клиентам.

Высылать реактивационные промокоды или спецпредложения тем, кто давно не взаимодействовал с компанией.

Дарить больше бонусов тем, кто покупает редко, но на большие суммы.

Показывать рекламу «уснувшей» базе.

Через 2-3 месяца необходимо сделать повторный анализ для измерения изменений.