RFM-анализ клиентской базы аптечной сети

Основной мотив исследования такой: скоро сезон спадет и выручки в аптеке начнут падать. У ребят есть механизм СМС-рассылок, но они не хотят тратить бюджеты впустую - они хотят провести персонализированную рассылку.

Идея такая: нет смысла предлагать персональную скидку для повышения среднего чека тем, кто вообще покупал один раз и то аскорбинку.

Цели исследования:

1. Провести классификацию пользователей
2. Сформировать рекомендации - какой группе пользователей что предлагать

Для решения этой задачи вам предлагается базово сосредоточится на RFM-анализе.

Работу проводим с БД apteka, таблица bonuscheques.
Структура таблицы:

bonuscheques

Таблица с информацией о транзакциях по бонусной системе.

- datetime - дата и время совершения транзакции
- shop - название аптеки
- card - номер бонусной карты
- bonus_earned - количество начисленных бонусов
- bonus_spent - количество потраченных бонусов
- summ - сумма чека
- summ_with_disc - сумма чека с учетом скидок и списаний бонусов
- doc_id - номер документа

Если в момент покупки касса была в оффлайн-режиме, то вместо номера карты записывается зашифрованная последовательность символов. В таком случае номер карты силами этой базы данных никак восстановить нельзя.

```sql
select *
from bonuscheques b
```

bonuscheques 1

select * from bonuscheques b | Введите SQL выражение чтобы отфильтровать результаты

	datetime	shop	card	bonus_earned	bonus_spent	summ	summ_with_disc	doc_id
1	2021-07-13 12:56:09.000	Аптека 2	2000200195023	51	0	3 400	3 400	15#2002741#65938#2_2
2	2021-07-30 10:42:00.000	Аптека 2	2000200193494	57	0	747	747	15#2002972#65955#2_5
3	2021-10-11 12:55:23.000	Аптека 2	2000200199106	92	253	3 077	3 077	15#2004060#66028#2_2
4	2021-10-14 14:48:56.000	Аптека 2	2000200168768	1	0	54	54	15#2004107#66031#2_5
5	2021-10-20 11:09:39.000	Аптека 2	2000200226314	101	0	1 733	1 733	15#2004192#66037#2_1
6	2021-10-29 13:55:20.000	Аптека 2	2000200192060	8	0	285	285	15#2004318#66046#2_4
7	2021-11-12 14:10:11.000	Аптека 2	2000200193241	17	0	588	588	15#2004481#66060#2_3
8	2021-11-15 12:21:15.000	Аптека 2	2000200183938	33	0	825	825	15#2004513#66063#2_2
9	2021-11-30 18:01:09.000	Аптека 2	2000200165811	26	0	886	886	15#2004712#66078#2_6
10	2021-12-03 15:51:20.000	Аптека 2	2000200186984	6	0	172	172	15#2004764#66081#2_7
11	2022-01-12 10:58:59.000	Аптека 2	2000200187189	12	0	427	427	15#2000125#66121#2_7
12	2022-01-28 10:44:36.000	Аптека 2	2000200209331	23	31	491	491	15#2000424#66137#2_9
13	2022-01-29 16:06:39.000	Аптека 2	2000200176015	41	0	1 379	1 379	15#2000447#66138#2_8
14	2022-01-31 10:58:23.000	Аптека 2	2000200166214	247	7	9 097	9 097	15#2000468#66140#2_7
15	2022-02-01 10:28:09.000	Аптека 2	2000200231912	60	0	380	380	15#2000492#66141#2_3
16	2022-03-02 10:04:46.000	Аптека 2	2000200191766	55	0	1 864	1 864	15#2001150#66170#2_1
17	2022-03-21 17:46:01.000	Аптека 2	2000200214083	11	0	368	368	15#2001757#66189#2_7

```sql
-- RFM-подготовка данных
-- Извлечение и агрегация метрик Recency, Frequency, Monetary по клиентам

with rfm_data as (
  select
    card as customer_id,
    max(datetime::date) as last_purchase_date,-- реальная дата последней покупки
    current_date - max(datetime::date) as recency_days,-- R: сколько дней прошло (оригинал)
    (current_date - max(datetime::date)) - 1273 as norm_recency, -- R: нормализованная (1...333)
    count(*) as purchase_count, -- F
    sum(summ) as total_spent  -- M
  from bonuscheques
  where card like '2000%'
  group by card
)
select *
from rfm_data;
```

Результат 1 | bonuscheques 2

with rfm_data as (select | Введите SQL выражение чтобы отфильтровать резу.

	customer_id	last_purchase_date	recency_days	norm_recency	purchase_count	total_spent
1	2000200229544	2021-11-16	1 479	209	2	1 657
2	2000200171775	2021-12-12	1 453	183	2	1 033
3	2000200229094	2022-01-10	1 424	154	2	1 235
4	2000200192497	2021-09-15	1 541	271	1	670
5	2000200240211	2022-03-21	1 354	84	1	1 136
6	2000200216834	2021-08-26	1 561	291	1	151
7	2000200232919	2022-05-10	1 304	34	5	1 574
8	2000200227953	2022-04-13	1 331	61	1	289
9	2000200151142	2022-04-18	1 326	56	1	853
10	2000200197751	2022-03-24	1 351	81	15	10 447
11	2000200238904	2022-04-01	1 343	73	1	586
12	2000200235484	2022-03-25	1 350	80	1	295
13	2000200181583	2022-03-10	1 365	95	1	1 130
14	2000200193555	2021-09-08	1 548	278	2	2 841
15	2000200229988	2022-01-03	1 431	161	1	317
16	2000200186571	2021-09-11	1 545	275	1	311
17	2000200220268	2022-03-12	1 363	93	4	998
18	2000200190950	2021-08-27	1 560	290	1	1 958
19	2000200208563	2022-05-13	1 301	31	3	2 236
20	2000200209089	2021-09-08	1 548	278	3	3 561
21	2000200232100	2022-05-26	1 288	18	5	531
22	2000200220749	2021-11-13	1 482	212	1	550

```sql
-- Анализируем распределение метрики Recency (давности последней покупки) по клиентам.
with rfm_raw as (
  select
    (current_date - max(datetime::date)) as recency_days  -- количество дней с момента последней покупки
  from bonuscheques
```

```
  where card is not null and card like '2000%'
  group by card
),
stats as (
  select
    min(recency_days) as min_recency,
    max(recency_days) as max_recency
  from rfm_raw
),
rfm_normalized as (
  select
    recency_days,
    1 + (recency_days - min_recency) * (332 - 1) / (max_recency - min_recency) as norm_recency
  from rfm_raw, stats
)
select
  count(*) as total_customers,  -- считаем количество строк
  min(norm_recency) as min_recency_norm,   -- будет ~1.0 -- минимальное значение дней с последней покупки
  max(norm_recency) as max_recency_norm,   -- будет ~332.0 -- максимальное значение дней с последней покупки
  avg(norm_recency) as avg_recency_norm, -- среднее количество дней с последней покупки по всем клиентам
  percentile_cont(0.5) within group (order by norm_recency) as median_recency_norm, -- медиана
  percentile_cont(0.25) within group (order by norm_recency) as q1_recency_norm, -- первый квартиль
  percentile_cont(0.75) within group (order by norm_recency) as q3_recency_norm -- третий квартиль
from rfm_normalized
```

Результат 1 | Результат 2

with rfm_raw as (select (cu | Введите SQL выражение чтобы отфильтровать результаты

	total_customers	min_recency_norm	max_recency_norm	avg_recency_norm	median_recency_norm	q1_recency_norm	q3_recency_norm
1	5 926	1	332	113,3908201147	87	30	183

```
-- Анализ распределения метрики Frequency (количество покупок) по клиентам.
select
  count(*) as total_customers,  --считаем количество строк
  min(quantity) as min_freq, -- минимальное количество покупок по клиентам
  max(quantity) as max_freq, -- максимальное количество покупок по клиентам
  avg(quantity) as avg_freq, -- среднее количество покупок по клиентам
  percentile_cont(0.5) within group (order by quantity) as median_freq, -- находим медиану
  percentile_cont(0.25) within group (order by quantity) as q1_freq, -- первый квартиль
  percentile_cont(0.75) within group (order by quantity) as q3_freq -- третий квартиль
from (
  select
    card as customer_id,
    count(*) as quantity
  from bonuscheques
  where card is not null and card like '2000%'
  group by card
) rfm
```

Результат 1

select count(*) as total_customers, min | Введите SQL выражение чтобы отфильтровать результаты

	total_customers	min_freq	max_freq	avg_freq	median_freq	q1_freq	q3_freq
1	5 926	1	217	3,5563617955	2	1	4

```
-- Анализ распределения метрики Monetary(траты) по клиентам.
select
  count(*) as total_customers,
  min(monetary) as min_mon, -- минимальные траты
  max(monetary) as max_mon, -- максимальные траты
  avg(monetary) as avg_mon, -- средние траты
  percentile_cont(0.5) within group (order by monetary) as median_mon,
  percentile_cont(0.25) within group (order by monetary) as q1_mon,
  percentile_cont(0.75) within group (order by monetary) as q3_mon
from (
  select
    card as customer_id,
    sum(summ) as monetary
  from bonuscheques
  where card is not null and card like '2000%'
  group by card
) rfm
```

Результат 1

select count(*) as total_customers, min | Введите SQL выражение чтобы отфильтровать результаты

	total_customers	min_mon	max_mon	avg_mon	median_mon	q1_mon	q3_mon
1	5 926	21	163 943	3 239,4262571718	1 475,5	729	3 315,75

```
-- ABC-анализ по метрике Monetary (общая сумма покупок клиента)
with rfm_raw as (
  select
     card as customer_id,
     max(datetime::date) as last_purchase_date, --дата последней покупки
     sum(summ) as monetary -- общая сумма всех покуок
  from bonuscheques
  where card is not null and card like '2000%'
  group by card -- группируем по карте(customer_id)
  ),
monetary_ranks as ( -- для каждого клиента
  select
     customer_id,
     monetary,
     sum(monetary) over () as total_revenue, -- сумма всех monetary по всем клиентам.
     sum(monetary) over (order by monetary desc rows unbounded preceding) as cum_revenue -- сумма monetary всех клиентов,
отсортированных по monetary в порядке убывания
  from rfm_raw
)
select *,
  case
     when cum_revenue <= 0.8 * total_revenue then 'A' -- Клиенты, чья накопленная выручка до 80%
     when cum_revenue <= 0.95 * total_revenue then 'B' -- Клиенты, чья накопленная выручка до 95% (включая тех, кто уже в A)
     else 'C'
  end as abc_class
from monetary_ranks
```

bonuscheques 1

with rfm_raw as (select card as custom | Введите SQL выражение чтобы отфильт

	customer_id	monetary	total_revenue	cum_revenue	abc_class
1	2000200170860	163 943	19 196 840	163 943	A
2	2000200196556	122 941	19 196 840	286 884	A
3	2000200196600	120 782	19 196 840	407 666	A
4	2000200189985	112 173	19 196 840	519 839	A
5	2000200204541	102 314	19 196 840	622 153	A
6	2000200150411	90 386	19 196 840	712 539	A
7	2000200186205	89 039	19 196 840	801 578	A
8	2000200205364	84 244	19 196 840	885 822	A
9	2000200189596	83 940	19 196 840	969 762	A
10	2000200195436	82 203	19 196 840	1 051 965	A
11	2000200193746	79 476	19 196 840	1 131 441	A
12	2000200186939	76 255	19 196 840	1 207 696	A
13	2000200196112	70 162	19 196 840	1 277 858	A
14	2000200197836	69 519	19 196 840	1 347 377	A
15	2000200189077	69 371	19 196 840	1 416 748	A
16	2000200204442	64 644	19 196 840	1 481 392	A
17	2000200224594	59 488	19 196 840	1 540 880	A

```
--В классических описаниях RFM часто используется 3-уровневая градация (High / Medium / Low),
--выбор 5 уровней обусловлен следующими соображениями:
--1)объем выборки больше 6000 клиентов, позволяет проводить более тонкую сегментацию
--2)5-балльная шкала даёт 125 уникальных RFM-кодов (5×5×5), что позволяет выявлять узкие, но ценные подгруппы (например, «активные, но мало
тратящие»)
--3)5-уровневая RFM-шкала широко применяется в ритейле, e-commerce и финансовых сервисах при работе с базами от нескольких тысяч клиентов и выше.
Это де-факто отраслевой стандарт для бизнесов, стремящихся к персонализации и повышению LTV.
with rfm_raw as (  -- Присвоение RFM-баллов
  select
     card as customer_id, -- Уникальный идентификатор клиента (карта)
```

```sql
      max(datetime::date) as last_purchase_date, -- дата последней покупки клиента
      (current_date - max(datetime::date)) - 1273 as recency_days, -- сколько дней прошло с последней покупки (Recency)
      count(*) as quantity, -- количество покупок клиента (Frequency)
      sum(summ) as monetary -- общая сумма покупок клиента (Monetary)
  from bonuscheques -- таблица с чеками
  where card is not null and card like '2000%' -- фильтрация: карта не пустая и начинается с '2000'
  group by card -- группировка по карте (клиенту)
),
rfm_scored as (
  select
      customer_id,
      recency_days,
      quantity,
      monetary,
                                case -- R: активность (чем меньше дней — тем выше балл)
        when recency_days <= 60 then 5  -- очень активные
        when recency_days <= 120 then 4 -- активные
        when recency_days <= 180 then 3 -- средняя активность
        when recency_days <= 365 then 2 -- спящие
        else 1                          -- давно не были
      end as r_score,
      case-- F: частота (чем больше покупок — тем выше балл)
        when quantity >= 10 then 5     -- частые покупатели
        when quantity >= 5 then 4      -- часто
        when quantity >= 3 then 3      -- иногда
        when quantity >= 2 then 2      -- редко
        else 1                         -- одноразовые
      end as f_score,
      case -- M: сумма (чем больше — тем выше балл)
        when monetary >= 5000 then 5   -- крупные траты
        when monetary >= 2500 then 4   -- высокие
        when monetary >= 1000 then 3   -- средние
        when monetary >= 500 then 2    -- низкие
        else 1                         -- минимальные
      end as m_score
  from rfm_raw
),
rfm_final as (
  select
      customer_id,
      recency_days,
      quantity,
      monetary,
      r_score,
      f_score,
      m_score,
      cast(r_score as text) || cast(f_score as text) || cast(m_score as text) as rfm_code,-- Объединяем баллы в один RFM-код (например, "543")
      -- Сегментация клиентов по RFM-баллам
      case
         when r_score >= 4 and f_score >= 3 and m_score >= 3 then 'Лучшие клиенты' -- активные, часто покупают, много тратят
        when r_score <= 2 and (f_score >= 4 or m_score >= 4) then 'Спящие клиенты (высокая ценность)'
        when r_score <= 2 and (f_score >= 3 or m_score >= 3) then 'Спящие клиенты (средняя ценность)' -- давно не были, но раньше активно тратили/покупали
        when r_score >= 4 and f_score >= 2 and m_score = 3 then 'Активные, но мало тратящие' -- часто, но мало
        when r_score >= 4 and f_score >= 2 and m_score < 3  then 'Потенциальные клиенты'
        when quantity = 1 and monetary >= 1000 then 'Одноразовые (высокий потенциал)'
        when quantity = 1 and monetary < 1000  then 'Одноразовые (низкий потенциал)' -- купили один раз и мало
        else 'Средние клиенты' -- все остальные
      end as segment
  from rfm_scored
)
select *
from rfm_final
order by monetary desc -- сортировка по общей сумме покупок по убыванию;
```

Результат 1 | bonuscheques 2

with rfm_raw as (select card as custo | Введите SQL выражение чтобы отфильтровать результаты

	customer_id	recency_days	quantity	monetary	r_score	f_score	m_score	rfm_code	segment
1	2000200170860	2	129	163 943	5	5	5	555	Лучшие клиенты
2	2000200196556	1	103	122 941	5	5	5	555	Лучшие клиенты
3	2000200196600	10	65	120 782	5	5	5	555	Лучшие клиенты
4	2000200189985	1	217	112 173	5	5	5	555	Лучшие клиенты
5	2000200204541	3	68	102 314	5	5	5	555	Лучшие клиенты
6	2000200150411	7	29	90 386	5	5	5	555	Лучшие клиенты
7	2000200186205	4	70	89 039	5	5	5	555	Лучшие клиенты
8	2000200205364	5	73	84 244	5	5	5	555	Лучшие клиенты
9	2000200189596	5	35	83 940	5	5	5	555	Лучшие клиенты
10	2000200195436	16	60	82 203	5	5	5	555	Лучшие клиенты
11	2000200193746	2	62	79 476	5	5	5	555	Лучшие клиенты
12	2000200186939	16	66	76 255	5	5	5	555	Лучшие клиенты
13	2000200196112	6	53	70 162	5	5	5	555	Лучшие клиенты
14	2000200197836	2	59	69 519	5	5	5	555	Лучшие клиенты
15	2000200189077	4	57	69 371	5	5	5	555	Лучшие клиенты
16	2000200204442	6	61	64 644	5	5	5	555	Лучшие клиенты
17	2000200224594	1	39	59 488	5	5	5	555	Лучшие клиенты
18	2000200194248	1	46	55 576	5	5	5	555	Лучшие клиенты
19	2000200196020	19	36	55 219	5	5	5	555	Лучшие клиенты
20	2000200208242	4	31	49 763	5	5	5	555	Лучшие клиенты
21	2000200198765	3	52	48 256	5	5	5	555	Лучшие клиенты
22	2000200220756	1	34	47 077	5	5	5	555	Лучшие клиенты

```
with rfm_raw as (  -- Присвоение RFM-баллов
  select
    card as customer_id, -- Уникальный идентификатор клиента (карта)
    max(datetime::date) as last_purchase_date, -- дата последней покупки клиента
    current_date - max(datetime::date) as recency_days, -- сколько дней прошло с последней покупки (Recency)
    count(*) as quantity, -- количество покупок клиента (Frequency)
    sum(summ) as monetary -- общая сумма покупок клиента (Monetary)
  from bonuscheques -- таблица с чеками
  where card is not null and card like '2000%' -- фильтрация: карта не пустая и начинается с '2000'
  group by card -- группировка по карте (клиенту)
),
rfm_scored as (
  select
    customer_id,
    recency_days,
    quantity,
    monetary,
    -- R (Recency): чем меньше дней прошло — тем активнее клиент. Балл 5 — самый активный.
    ntile(5) over (order by recency_days desc) as r_score, -- разбиваем на 5 групп по убыванию дней (т.е. самые свежие покупки
получают 5)
    -- F (Frequency): чем больше покупок — тем выше балл
    ntile(5) over (order by quantity asc) as f_score, -- разбиваем на 5 групп по возрастанию (т.е. больше покупок — выше балл)
    -- M (Monetary): чем больше сумма — тем выше балл
    ntile(5) over (order by monetary asc) as m_score -- разбиваем на 5 групп по возрастанию (т.е. больше трат — выше балл)
  from rfm_raw
),
rfm_final as (
  select
    customer_id,
    recency_days,
    quantity,
    monetary,
    r_score,
    f_score,
    m_score,
    cast(r_score as text) || cast(f_score as text) || cast(m_score as text) as rfm_code,-- Объединяем баллы в один RFM-код (например,
"543")
    -- Сегментация клиентов по RFM-баллам
    case
      when r_score >= 4 and f_score >= 4 and m_score >= 4 then 'Лучшие клиенты' -- активные, часто покупают, много тратят
      when r_score <= 2 and (f_score >= 4 or m_score >= 4) then 'Спящие клиенты' -- давно не были, но раньше активно
тратили/покупали
      when r_score >= 4 and f_score >= 3 and m_score <= 2 then 'Активные, но мало тратящие' -- часто, но мало
      when quantity = 1 and m_score <= 2 then 'Одноразовые' -- купили один раз и мало
      else 'Средние' -- все остальные
    end as segment
  from rfm_scored
),
```

```sql
monetary_ranks as ( -- ABC-анализ по Monetary
  select
    customer_id, -- Уникальный индификатор клиента
    monetary, -- Общая сумма покупок клиента
    sum(monetary) over () as total_revenue, -- Общая выручка по всем клиентам (сумма всех monetary)
    sum(monetary) over (order by monetary desc rows unbounded preceding) as cum_revenue -- Накопленная выручка: суммируем
monetary клиентов, отсортированных по убыванию monetary
    -- Это позволяет определить, какой вклад в общую выручку делает каждый клиент в порядке убывания
  from rfm_raw -- таблица с клиентами и их метриками (из предыдущего шага)
),
abc as (
  select
    customer_id,
    case
      when cum_revenue <= 0.8 * total_revenue then 'a'  -- Если накопленная выручка (cum_revenue) <= 80% от общей — класс A (самые
ценные клиенты)
      when cum_revenue <= 0.95 * total_revenue then 'b' -- Если <= 95% — класс B (средние по ценности)
      else 'c'  -- Остальные — класс C (наименее ценные)
    end as abc_class
  from monetary_ranks
),
final_enriched as ( -- Объединение RFM и ABC
  select
    f.*, -- Все колонки из RFM-анализа (customer_id, rfm_code, segment и т.д.)
    a.abc_class -- Добавляем ABC-класс к каждому клиенту
  from rfm_final f -- результаты RFM-сегментации
  join abc a on f.customer_id = a.customer_id
)
select -- Финальный отчёт: статистика по сегментам
  segment, -- RFM-сегмент клиента (например, "Лучшие клиенты", "Спящие" и т.д.)
  abc_class, -- ABC-класс клиента (a, b, c)
  count(*) as customers, -- количество клиентов в группе
  round(avg(monetary), 2) as avg_monetary, -- средняя сумма покупок в группе
  round(avg(recency_days), 1) as avg_recency, -- среднее количество дней с последней покупки
  round(avg(quantity), 1) as avg_frequency -- среднее количество покупок
from final_enriched
group by segment, abc_class -- группируем по RFM-сегменту и ABC-классу
order by avg_monetary desc; -- сортируем по средней сумме покупок по убыванию
```

Результат 1 | Результат 2

with rfm_raw as (select card as custo | Введите SQL выражение чтобы отфильтровать результаты

Таблица | Текст

	segment	abc_class	customers	avg_monetary	avg_recency	avg_frequency
1	Лучшие клиенты	a	1 298	9 225,5	37,4	9,7
2	Средние клиенты	a	342	4 743,77	110,9	3,6
3	Спящие клиенты (высокая ценность)	a	234	4 307,4	237,9	2,9
4	Одноразовые (высокий потенциал)	a	168	3 779,7	76,8	1
5	Спящие клиенты (средняя ценность)	a	35	2 409,66	244,3	1,7
6	Активные, но мало тратящие	a	13	2 392,69	52,5	2
7	Спящие клиенты (высокая ценность)	b	12	1 780	211,1	5,3
8	Лучшие клиенты	b	354	1 687	49,1	3,9
9	Активные, но мало тратящие	b	259	1 565,64	48,8	2
10	Спящие клиенты (средняя ценность)	b	467	1 488,91	250,3	1,6
11	Одноразовые (высокий потенциал)	b	470	1 457,03	78,4	1
12	Средние клиенты	b	188	1 448,76	155,9	2,8
13	Потенциальные клиенты	b	59	940,98	45,1	2,6
14	Одноразовые (низкий потенциал)	b	159	940,19	157,1	1
15	Спящие клиенты (средняя ценность)	c	11	695,82	241,4	3,2
16	Средние клиенты	c	139	608,55	192,4	2,1
17	Потенциальные клиенты	c	209	606,12	54,7	2,4
18	Одноразовые (низкий потенциал)	c	1 509	491	152,1	1

Сформировать рекомендации - какой группе пользователей что предлагать.

Важно:

- Удержать лояльных
- Вернуть спящих
- Повысить чек у активных
- Превратить одноразовых в постоянных

Ключевые клиенты, A-класс.

🔹 **Что предлагать:**

- Персонализированные привилегии: эксклюзивные скидки, ранний доступ к новинкам, подарки за крупные покупки.
- Программы лояльности 2.0: накопление бонусов с повышенным коэффициентом, VIP-статус.
- Подарки при достижении целей: например, "При покупке на 5000 руб. — бесплатная доставка или подарок".

Потерянные высокодоходные клиенты, A/B-класс.

🔹 **Что предлагать:**

- Скидка 15–20% на первую покупку после долгого перерыва.
- Напоминание о накопленных бонусах: “У вас 500 бонусов — успейте потратить до конца месяца!”.

Активные, но мало тратящие (RFM) + B/C

🔹 **Что предлагать:**

- Пакетные предложения: “Купите 3 препарата — получите 4-й бесплатно”.
- Бонусы за увеличение чека: “При покупке на 1000+ — 10% бонусов дополнительно”.

- Образовательный контент: “Как правильно принимать этот препарат? Читайте наш гайд”.

Одноразовые (RFM) + C

🔹 **Что предлагать:**

- Приветственный бонус: “Спасибо за первый визит! Получите 10% скидку на следующую покупку”.
- Товары-лидеры продаж: дешевые, популярные, быстрые в принятии решения (например, обезболивающие, витамины).
- Простая программа лояльности: “Зарегистрируйтесь и получите бонус сразу”.