

RFM-аналіз для мережі аптек

ЗАДАЧА:

Після сезону виручки починають падати в аптеках. Планується стимуляція продажів за рахунок персональних пропозицій клієнтам з допомогою смс-розсилок.

Потрібно:

- Провести класифікацію клієнтів
- Сформулювати гіпотези з пропозиціями по групах.

Працюємо в таблиці `bonuscheques` де записані всі покупки з використанням карти лояльності.

`datetime` - дата і час

`shop` - аптека

`card` - номер карти лояльності

`bonus_earned` - нараховано бонусів

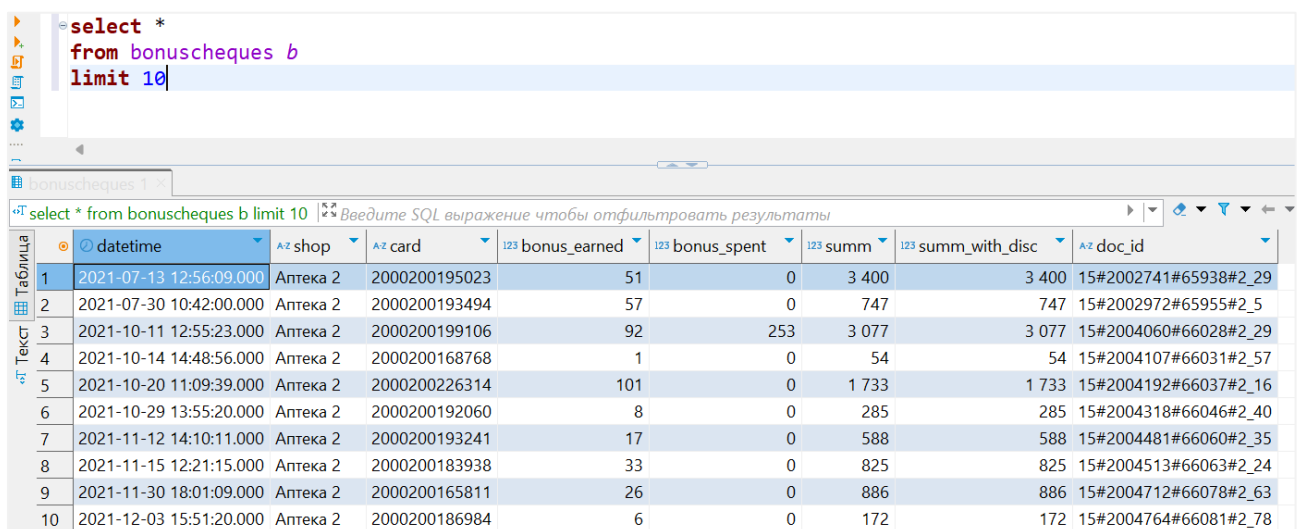
`bonus_spent` - списано бонусів

`summ` - сума без знижки

`summ_with_disc` - остаточна сума зі знижкою

`doc_id` - ідентифікатор касового чека

Приклад таблиці:



The screenshot shows a database management tool with a SQL query editor at the top and a results table below. The query is `select * from bonuscheques b limit 10`. The results table has 10 rows and 9 columns: `datetime`, `shop`, `card`, `bonus_earned`, `bonus_spent`, `summ`, `summ_with_disc`, and `doc_id`. The data shows transactions from various pharmacies over time, with columns for earned and spent bonuses, and sums with and without discounts.

	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_29
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_29
4	2021-10-14 14:48:56.000	Аптека 2	2000200168768	1	0	54	54	15#2004107#66031#2_57
5	2021-10-20 11:09:39.000	Аптека 2	2000200226314	101	0	1 733	1 733	15#2004192#66037#2_16
6	2021-10-29 13:55:20.000	Аптека 2	2000200192060	8	0	285	285	15#2004318#66046#2_40
7	2021-11-12 14:10:11.000	Аптека 2	2000200193241	17	0	588	588	15#2004481#66060#2_35
8	2021-11-15 12:21:15.000	Аптека 2	2000200183938	33	0	825	825	15#2004513#66063#2_24
9	2021-11-30 18:01:09.000	Аптека 2	2000200165811	26	0	886	886	15#2004712#66078#2_63
10	2021-12-03 15:51:20.000	Аптека 2	2000200186984	6	0	172	172	15#2004764#66081#2_78

- Перевіримо дату першого та останнього запису та наявність пропусків в даних.

```

select min(datetime) as first,
       max(datetime) as last,
       max(datetime) - min(datetime) as period,
       count(distinct date(datetime)) as cnt_unique_days
from bonuscheques b

```

	first	last	period	cnt_unique_days
1	2021-07-12 08:09:44.000	2022-06-09 21:49:45.000	332 days 13:40:01	333

Бачимо, що у нас дані за 11 місяців. Так як це менше року, то будемо робити аналіз по всіх записах в таблиці. Також ми дізнались, що система працювала без збоїв, дані записані за 333 дні і це співпадає з періодом між датами. Але якщо пропадає інтернет то картка записується некоректно тому будемо використовувати лише записи, що починаються з 200*

Кроки:

- Групуємо дані по картам лояльності клієнтів
- Рахуємо для кожного клієнта кількість днів від його останньої покупки (recency)
- Рахуємо загальну кількість покупок по кожному клієнту (frequency)
- Сумуємо всі суми покупок на кожного клієнта (monetary)
- Сортуюмо по максимальній сумі покупок, мінімальній давності та максимальній кількості покупок

```

select card,
       extract('day' from (select max(datetime) from bonuscheques b) - max(datetime)) as recency,
       count(*) as frequency,
       sum(summ_with_disc) as monetary
from bonuscheques b
where card like '200%'
group by card
order by monetary desc, recency asc, frequency desc

```

	card	recency	frequency	monetary
1	2000200170860	1	129	162 687
2	2000200196556	0	103	122 410
3	2000200196600	9	65	120 757
4	2000200189985	0	217	112 033
5	2000200204541	2	68	102 011
6	2000200150411	6	29	90 315

Далі треба сформувати групи для RFM-аналізу. Кожний стовпчик ділимо на три категорії від 1 до 3 (1 - відмінно, 2 - нормально, 3 - погано).

Давність останньої покупки - чим менше тим краще.

Кількість покупок і сума - навпаки, чим більше тим краще.

- Треба задати пороги значень і вирішити, що для нашого бізнесу добре, що нормально і що погано.

Рішення перше:

Воно максимально просте, але воно всерівно дозволить зекономити кошти на розсилці.

Ми не будемо пропонувати персональну скидку людям які одного разу купили в нас пачку презервативів.

Рішення:

```
with gr_tab as (
select card,
       extract('day' from (select max(datetime) from bonuscheques b) - max(datetime)) as recency,
       count(*) as frequency,
       sum(summ_with_disc) as monetary
from bonuscheques b
where card like '200%'
group by card
order by monetary desc, recency asc, frequency desc
)
select *,
       ntile(3) over (order by recency) as R,
       ntile(3) over (order by frequency desc) as F,
       ntile(3) over (order by monetary desc) as M
from gr_tab
```

card	recency	frequency	monetary	r	f	m
2000200240976	0	9	8 962	1	1	1
2000200253969	0	1	1 163	1	2	2
2000200254904	0	2	1 436	1	2	2
2000200194248	0	46	55 452	1	1	1
2000200189985	0	217	112 033	1	1	1

Попередню згруповану таблицю добавимо в конструкцію **with** і назвемо таблицю “gr_tab”

За допомогою віконної функції **ntile(3) over (order by recency)** ми відсортували стовбчик recency, розбили його на три рівні частини і присвоїли кожній частині ранг (за замовчуванням сортування йде від меншого до більшого, якщо треба від більшого, то указуємо конструкцію desc).

- Добавимо ще стовбець з об'єднаною групою для більшої простоти роботи

```
with gr_tab as (
select card,
       extract('day' from (select max(datetime) from bonuscheques b) - max(datetime)) as recency,
       count(*) as frequency,
       sum(summ_with_disc) as monetary
from bonuscheques b
where card like '200%'
group by card
order by monetary desc, recency asc, frequency desc
), t2 as (
select *,
       ntile(3) over (order by recency) as R,
       ntile(3) over (order by frequency desc) as F,
       ntile(3) over (order by monetary desc) as M
from gr_tab
)
select *, r*100 + f*10 + m as rfm_group
from t2
```

card	recency	frequency	monetary	r	f	m	rfm_group
2000200240976	0	9	8 962	1	1	1	111
2000200253969	0	1	1 163	1	2	2	122
2000200254904	0	2	1 436	1	2	2	122
2000200194248	0	46	55 452	1	1	1	111
2000200189985	0	217	112 033	1	1	1	111

Тут ми використали трюк з множенням та додаванням)))

- Подивимось кількість клієнтів у кожній групі

```

with gr_tab as (
select card,
       extract('day' from (select max(datetime) from bonuscheques b) - max(datetime)) as recency,
       count(*) as frequency,
       sum(summ_with_disc) as monetary
from bonuscheques b
where card like '200%'
group by card
order by monetary desc, recency asc, frequency desc
), t2 as (
select *,
       ntile(3) over (order by recency) as R,
       ntile(3) over (order by frequency desc) as F,
       ntile(3) over (order by monetary desc) as M
from gr_tab
), final_table as (
select *, r*100 + f*10 + m as rfm_group
from t2
)
select rfm_group, count(card) as cnt_clients
from final_table
group by rfm_group
order by rfm_group

```

Результат 1

rfm_group	cnt_clients
111	846
112	179
113	26

Маємо 27 груп. Повна таблиця:

Результат 1

rfm_group	cnt_clients
111	846
112	179
113	26
121	111
122	269
123	120
131	22
132	109
133	294
211	447
212	180
213	29
221	157
222	371
223	193
231	42
232	151
233	405
311	156
312	98
313	15
321	159
322	406
323	189
331	36
332	212
333	704

Тепер ми маємо уявлення як у нас розбитий пул клієнтів по категоріям. Де 111 – VIP клієнт, а 333 – це купував давно, один раз, на маленьку суму. 704 смски вже можемо не відправляти точно)))

Рішення друге:

Припустимо, що ми поспілкувались з фахівцями (менеджерами, фармацевтами, маркетологами) і вияснили як правильно розподілити клієнтську базу.

Отже тепер ділимо так:

Давність покупки(R): група 1 це - до 30 днів, 2 - від 31 до 90 днів, 3 - більше 90 днів

Частота покупок(F): 1 - це більше 4 покупок за рік, 2 - 3-4 покупки, 3 - 1-2 покупки

Сума витрат(M): тут вирішили що краще поділити статистично по 33-му та 66-му процентилям.

В згрупованій таблиці **gr_tab** додаємо три колонки по умові за допомогою конструкції **CASE**

```
with gr_tab as (
  select card,
         extract('day' from (select max(datetime) from bonuscheques b) - max(datetime)) as recency,
         count(*) as frequency,
         sum(summ_with_disc) as monetary
  from bonuscheques b
  where card like '200%'
  group by card
  order by monetary desc, recency asc, frequency desc
)
select *,
       case
         when recency <= 30 then '1'
         when recency <= 90 then '2'
         else '3'
       end as r,
       case
         when frequency >= 5 then '1'
         when frequency >= 3 then '2'
         else '3'
       end as f,
       case -- тут розподіляємо стосовно пецентилів
         when monetary >= (select percentile_cont(0.66) within group (order by monetary)
                             from gr_tab) then '1'
         when monetary >= (select percentile_cont(0.33) within group (order by monetary)
                             from gr_tab) then '2'
         else '3'
       end as m
  from gr_tab
```

	card	recency	frequency	monetary	r	f	m
1	2000200170860	1	129	162 687	1	1	1
2	2000200196556	0	103	122 410	1	1	1
3	2000200196600	9	65	120 757	1	1	1

І щоб не створювати додаткових конструкцій, я приберу аліаси(назви) в нових колонках і згрупую їх в одну функцією **CONCAT()**. (rfm_group)

```

with gr_tab as (
select card,
       extract('day' from (select max(datetime) from bonuscheques b) - max(datetime)) as recency,
       count(*) as frequency,
       sum(summ_with_disc) as monetary
from bonuscheques b
where card like '200%'
group by card
order by monetary desc, recency asc, frequency desc
)
select *,
       concat(case
               when recency <=30 then '1'
               when recency <= 90 then '2'
               else '3'
               end,
               case
               when frequency >= 5 then '1'
               when frequency >= 3 then '2'
               else '3'
               end,
               case -- тут розподіляємо стосовно пецентилів
               when monetary >= (select percentile_cont(0.66) within group (order by monetary)
                                   from gr_tab) then '1'
               when monetary >= (select percentile_cont(0.33) within group (order by monetary)
                                   from gr_tab) then '2'
               else '3'
               end) as rfm_group
from gr_tab

```

card	recency	frequency	monetary	rfm_group
2000200170860	1	129	162 687	111
2000200196556	0	103	122 410	111
2000200196600	9	65	120 757	111

Подивимось чи змінилась кількість клієнтів в наших групах.

```

with gr_tab as (
select card,
       extract('day' from (select max(datetime) from bonuscheques b) - max(datetime)) as recency,
       count(*) as frequency,
       sum(summ_with_disc) as monetary
from bonuscheques b
where card like '200%'
group by card
order by monetary desc, recency asc, frequency desc
), final_table as (
select *,
       concat(case
               when recency <=30 then '1'
               when recency <= 90 then '2'
               else '3'
               end,
               case
               when frequency >= 5 then '1'
               when frequency >= 3 then '2'
               else '3'
               end,
               case -- тут розподіляємо стосовно пецентилів
               when monetary >= (select percentile_cont(0.66) within group (order by monetary)
                                   from gr_tab) then '1'
               when monetary >= (select percentile_cont(0.33) within group (order by monetary)
                                   from gr_tab) then '2'
               else '3'
               end) as rfm_group
from gr_tab
)
select rfm_group, count(card)
from final_table
group by rfm_group
order by rfm_group

```

Результат:

Результат 1		
with gr_tab as (select card, extract('day' fro Введите SQL выражение чтобы отфильтровать результаты		
Таблица	rfm_group	count
1	111	587
2	112	35
3	113	5
4	121	126
5	122	112
6	123	18
7	131	79
8	132	255
9	133	299
10	211	319
11	212	34
12	213	3
13	221	124
14	222	155
15	223	27
16	231	130
17	232	333
18	233	403
19	311	188
20	312	42
21	313	1
22	321	203
23	322	175
24	323	31
25	331	259
26	332	814
27	333	1 169

Одразу видно що VIP клієнтів тут вже менше, а клієнтів «333» на багато більше.

Для різних компаній різні пороги і єдиного правильного рішення напевно немає. Або я його просто ще не знаю))

Рухаємось далі, отже ми маємо 27 груп клієнтів. Об'єднаємо ці групи в смислові кластери:

			RFM
Кількість покупок (F)			
Давність покупки (R)	111	121	131
	112	122	132
	113	123	133
	211	221	231
	212	222	232
	213	223	233
	311	321	331
	312	322	332
	313	323	333
Сума витрат (M)			
			xx1 - Клієнти з високим чеком
			VIP клієнти
			Постійні клієнти
			Нові клієнти
			Постійні: сплячі та покидають
			Сплячі:
			Середня к-ть покупок та разові
			Покидають
			Покинули

Формулюємо гіпотези з пропозиціями по групах

xx1 - Клієнти з високим чеком:

Звернути особливу увагу на клієнтів з високим чеком, проаналізувати ці групи клієнтів додатково. Виділити на активацію цих груп більше ресурсів.

VIP клієнти:

Запросити у VIP-клуб, запропонувати програму лояльності для віпів.

Постійні клієнти:

Акції спрямовані на підвищення середнього чеку. Сезонні пропозиції та знижки. Пропозиції з супутніми товарами. Знижки на другу одиницю.

Нові клієнти:

Welcome-повідомлення з інформацією про нас і наші вигоди. Для нових клієнтів з високим чеком - спеціальна пропозиція.

Постійні сплячі та покидають:

Провести опитування і дізнатись причину. Максимальні зусилля на реактивацію.

Сплячі

Середня к-ть покупок та разові:

Пробуємо стимулювати релевантними товарами, пропозиціями купити той самий товар ще раз, пропозиції з обмеженим терміном.

Покидають:

Пробуємо стимулювати релевантними товарами, пропозиціями купити той самий товар ще раз, відправляємо пропозиції з обмеженим терміном.

Покинули:

Не тратимось. Стандартний ланцюг листів реактивації. Можливо клієнтам з високим чеком відправити пропозицію з обмеженим терміном або релевантними товарами.

Приклади смс:

Лише в квітні подвійні бонуси на покупки більше 500 грн.

Чому ви більше не заходите? Що не так? Ми можемо це виправити? Лише 3 питання заповніть анкету.

Даруємо персональну знижку 10%. Зробіть будь-яку покупку до кінця місяці для активації.

Для всіх нових клієнтів знижка 7% на наступні 3 покупки. Дякуємо за те що ви з нами.

Разом вигідніше. Знижка 15% на кожен третій товар у чеку. Лише до кінця місяця.

Ви стали VIP – клієнтом. Доступ в кабінет для справжніх друзів за посиланням.

Раніше ви купували Товар «Назва». Даруємо знижку 10% на всі товари з категорії «Назва категорії»

Бажаю всім високих прибутків. Дякую.