

Использование СУБД PostgreSQL



Работу выполнил Александров Михаил
Александрович в рамках курса по
анализу данных

1. Запрос "Топ-3 аптеки"

а. Вывести топ 3 аптеки по объему продаж (GROUP BY, LIMIT)

```
1  /*
2      1. Запрос "Топ-3 аптеки"
3          а. Вывести топ 3 аптеки по объему продаж (GROUP BY, LIMIT)
4  */
5  SELECT
6      pharmacy_name,
7      SUM(price * COUNT) AS total_sales
8  FROM
9      pharma_orders
10 GROUP BY
11     pharmacy_name
12 ORDER BY
13     total_sales DESC
14 LIMIT 3;
15
```

pharmacy_name	total_sales
Столичная	1896483
Доктор Айболит	1893776
Здравсити	1863037

2. Запрос "Топ-3 лекарства"

а. Вывести топ 3 лекарства по объему продаж

```
1  /*
2      2. Запрос "Топ-3 лекарства"
3      а. Вывести топ 3 лекарства по объему продаж
4  */
5  SELECT
6      drug,
7      SUM(price * COUNT) AS total_sales
8  FROM
9      pharma_orders
10 GROUP BY
11     drug
12 ORDER BY
13     total_sales DESC
14 LIMIT 3;
15
```

drug	total_sales
Ибупрофен	777548
Цитрамон	776332
Аквафорте	771579

3. Запрос "Аптеки от 1.8 млн оборота«

а. Вывести аптеки, имеющие более 1.8 млн оборота (HAVING)

```
1 /*
2     3. Запрос "Аптеки от 1.8 млн оборота"
3     а. Вывести аптеки, имеющие более 1.8 млн оборота (HAVING)
4 */
5 SELECT
6     pharmacy_name,
7     SUM(price * COUNT) AS total_revenue
8 FROM
9     pharma_orders
10 GROUP BY
11     pharmacy_name
12 HAVING
13     SUM(price * COUNT) > 1800000;
14
15
```

pharmacy_name	total_revenue
Столичная	1896483
Доктор Айболит	1893776
Здравсити	1863037

4. Запрос "Накопленная сумма продаж по каждой аптеке"
(OVER)

```
1  /*
2      4. Запрос "Накопленная сумма продаж по каждой аптеке" (OVER)
3  */
4  SELECT
5      pharmacy_name,
6      order_id,
7      drug,
8      price,
9      COUNT,
10     city,
11     report_date,
12     SUM(price * COUNT) OVER (PARTITION BY pharmacy_name ORDER BY report_date) AS cumulative_sales
13 FROM
14     pharma_orders;
15
16
```

pharmac...	order_id	drug	price	count	city	report_date	cumulative_sales
Столичная	634960	Лоратадин	684	5	Москва	2024-02-01	8218
Столичная	332739	Парацетамол	396	1	Москва	2024-02-01	8218
Столичная	591973	Аспирин	450	5	Санкт-Петер...	2024-02-01	8218
Столичная	332956	Аквамарис	538	4	Москва	2024-02-01	8218
Столичная	415834	Аквамарис	565	5	Москва	2024-02-02	24310
Столичная	401784	Аквамарис	887	2	Москва	2024-02-02	24310
Столичная	441727	Ибупрофен	450	3	Санкт-Петер...	2024-02-02	24310
Столичная	454719	Омепразол	513	5	Санкт-Петер...	2024-02-02	24310
Столичная	668923	Аспирин	367	5	Москва	2024-02-02	24310
Столичная	542102	Ибупрофен	450	2	Санкт-Петер...	2024-02-02	24310

2-ой вариант

```
1  /*
2      4. Запрос "Накопленная сумма продаж по каждой аптеке" (OVER)
3  */
4  SELECT
5      pharmacy_name,
6      MIN(report_date) AS start_date,
7      MAX(report_date) AS end_date,
8      SUM(price * COUNT) AS cumulative_sales
9  FROM
10     pharma_orders
11  GROUP BY
12     pharmacy_name;
13
14
15
16
```

pharmacy_name	start_date	end_date	cumulative_sales
Столичная	2024-02-01	2024-06-01	1896483
Аптека №1	2024-02-01	2024-06-01	1756481
Аптека.ру	2024-02-01	2024-06-01	1743513
Горздрав	2024-02-01	2024-06-01	1741435
Доктор Айболит	2024-02-01	2024-06-01	1893776
Здравсити	2024-02-01	2024-06-01	1863037

5. Запрос "Количество клиентов в аптеках"
- a. Соединить таблицы заказов и клиентов (JOIN)
 - b. Посчитать кол-во уникальных клиентов на каждую аптеку (DISTINCT)
 - c. Отсортировать аптеки по убыванию кол-ва клиентов (ORDER BY)

```
1  /*
2      5. Запрос "Количество клиентов в аптеках"
3          a. Соединить таблицы заказов и клиентов (JOIN)
4          b. Посчитать кол-во уникальных клиентов на каждую аптеку (DISTINCT)
5          c. Отсортировать аптеки по убыванию кол-ва клиентов (ORDER BY)
6  */
7  SELECT
8      o.pharmacy_name,
9      COUNT(DISTINCT o.customer_id) AS unique_client_count
10 FROM
11     pharma_orders o
12 JOIN
13     customers c ON o.customer_id = c.customer_id
14 GROUP BY
15     o.pharmacy_name
16 ORDER BY
17     unique_client_count DESC;
18
19
20
21
22
23
```

pharmacy_name	unique_client_count
Столичная	441
Аптека.ру	437
Доктор Айболит	430
Здравсити	428
Аптека №1	427

6. Запрос "Лучшие клиенты"

- а. Соединить таблицы заказов и клиентов (JOIN)
- б. Посчитать тотал сумму заказов для каждого клиента
- с. Проранжировать клиентов по убыванию суммы заказа (row_number)
- д. Оставить топ-10 клиентов

Run

PostgreSQL

PostgreSQL.34

```
1  /*
2      6. Запрос "Лучшие клиенты"
3      а. Соединить таблицы заказов и клиентов (JOIN)
4      б. Посчитать тотал сумму заказов для каждого клиента
5      с. Проранжировать клиентов по убыванию суммы заказа (row_number)
6      д. Оставить топ-10 клиентов
7  */
8  SELECT
9      C.customer_id,
10     C.last_name,
11     C.first_name,
12     C.second_name,
13     SUM(o.price * o.count) AS total_order_value,
14     ROW_NUMBER() OVER (ORDER BY SUM(o.price * o.count) DESC) AS RANK
15 FROM
16     pharma_orders o
17 JOIN
18     customers C ON o.customer_id = C.customer_id
19 GROUP BY
20     C.customer_id, C.last_name, C.first_name, C.second_name
21 ORDER BY
22     total_order_value DESC
23 LIMIT 10;
24
```

customer_id	last_name	first_name	second_name	total_order_value	rank
37574595	Буров	Бажен	Германович	45174	1
65801768	Ершов	Велимир	Устинович	45100	2
46335390	Григорьева	Ульяна	Тимофеевна	43870	3
54610616	Максимов	Остап	Гертрудович	43790	4
45681620	Брагина	Акулина	Оскарровна	41731	5

7. Запрос "Накопленная сумма по клиентам"

а. Соединить таблицы заказов и клиентов

б. Соединить ФИО в одно поле

в. Рассчитать накопленную сумму по каждому клиенту

```
1  /*
2      7. Запрос "Накопленная сумма по клиентам"
3      а. Соединить таблицы заказов и клиентов
4      б. Соединить ФИО в одно поле
5      в. Рассчитать накопленную сумму по каждому клиенту
6  */
7  SELECT
8      CONCAT(C.last_name, ' ', C.first_name, ' ', C.second_name) AS full_name,
9      SUM(po.price * po.count) AS total_amount
10 FROM
11     customers C
12 JOIN
13     pharma_orders po ON C.customer_id = po.customer_id
14 GROUP BY
15     C.customer_id, C.last_name, C.first_name, C.second_name
16 ORDER BY
17     total_amount DESC;
18
19
20
```

full_name	total_amount
Буров Бажен Германович	45174
Ершов Велимир Устинович	45100
Григорьева Ульяна Тимофеевна	43870
Максимов Остап Гертрудович	43790
Брагина Акулина Оскаровна	41731
Коновалова Олимпиада Семеновна	40376
Воронова Лукия Афанасьевна	39863

8. Запрос "Самые частые клиенты аптек Горздрав и Здравсити"
- а Сделать две временные таблицы: для аптеки горздрав и здравсити (WITH)
 - б. Внутри каждой соединить таблицы заказов и клиентов (JOIN)
 - с. Внутри каждой привести данные в формат "клиент - кол-во заказов в аптеке"
 - д. Внутри каждой оставить топ 10 клиентов каждой аптеки
 - е. Объединить клиентов с помощью UNION

Run

PostgreSQL

PostgreSQL.34

```
1  /*
2      8. Запрос "Самые частые клиенты аптек Горздрав и Здравсити"
3          а Сделать две временные таблицы: для аптеки горздрав и здравсити (WITH)
4          б. Внутри каждой соединить таблицы заказов и клиентов (JOIN)
5          с. Внутри каждой привести данные в формат "клиент - кол-во заказов в аптеке"
6          д. Внутри каждой оставить топ 10 клиентов каждой аптеки
7          е. Объединить клиентов с помощью UNION
8  */
9  WITH gorzdrav_clients AS (
10      SELECT
11          CONCAT(C.last_name, ' ', C.first_name, ' ', C.second_name) AS client_name,
12          COUNT(po.order_id) AS order_count
13      FROM
14          pharma_orders po
15      JOIN
16          customers C ON po.customer_id = C.customer_id
17      WHERE
18          po.pharmacy_name = 'Горздрав'
19      GROUP BY
20          C.customer_id, C.last_name, C.first_name, C.second_name
21      ORDER BY
22          order_count DESC
23      LIMIT 10
24  ),
25  zdravsiti_clients AS (
26      SELECT
27          CONCAT(C.last_name, ' ', C.first_name, ' ', C.second_name) AS client_name,
28          COUNT(po.order_id) AS order_count
```

client_name	order_count	pharmacy
Беляев Клавдий Измаилович	8	Горздрав
Филиппова Майя Максимовна	7	Горздрав
Симонова Евдокия Ильинична	6	Горздрав
Молчанов Богдан Зиновьевич	6	Горздрав

Запрос 9. "Сравнение динамики продаж между Москвой и Санкт-Петербургом по аптекам":

- а. Посчитать продажи лекарств в Москве по аптекам и по месяцам;
- б. Посчитать продажи лекарств в Санкт-Петербурге по аптекам и по месяцам;
- с. Соединить таблицы по аптекам и по месяцам и найти разницу в процентах;

1-ый вариант

```
1 /*
2     Запрос 1. "Сравнение динамики продаж между Москвой и Санкт-Петербургом по аптекам":
3         а. Посчитать продажи лекарств в Москве по аптекам и по месяцам;
4         б. Посчитать продажи лекарств в Санкт-Петербурге по аптекам и по месяцам;
5         с. Соединить таблицы по аптекам и по месяцам и найти разницу в процентах;
6 */
7 WITH msk_sales AS (
8     SELECT
9         pharmacy_name,
10        drug,
11        DATE_TRUNC('month', TO_DATE(report_date, 'YYYY-MM-DD')) AS MONTH,
12        SUM(price * COUNT) AS total_sales,
13        ROUND(SUM(price * COUNT) * 100.0 / SUM(SUM(price * COUNT)) OVER (PARTITION BY DATE_TRUNC('month', TO_DATE(report_date, 'YYYY-MM-DD'))
14    FROM
15        pharma_orders
16    WHERE
17        city = 'Москва'
18    GROUP BY
19        pharmacy_name, drug, MONTH
```

pharmac...	drug	month	msk...	msk_perc_month	spb_city	spb_perc_month	diff_perc_month
Столичная	Аквалор	2024-02-01...	Москва	5.07	Санкт-Петер...	9.13	-4.06
Столичная	Аквалор	2024-03-01...	Москва	5.32	Санкт-Петер...	6.58	-1.26
Столичная	Аквалор	2024-04-01...	Москва	6.30	Санкт-Петер...	12.35	-6.05
Столичная	Аквалор	2024-05-01...	Москва	4.69	Санкт-Петер...	7.72	-3.03
Столичная	Аквалор	2024-06-01...	Москва	20.46	Санкт-Петер...	0.00	20.46
Столичная	Аквамарис	2024-02-01...	Москва	9.01	Санкт-Петер...	2.92	6.09
Столичная	Аквамарис	2024-03-01...	Москва	4.49	Санкт-Петер...	4.95	-0.46
Столичная	Аквамарис	2024-04-01...	Москва	10.96	Санкт-Петер...	5.97	4.99
Столичная	Аквамарис	2024-05-01...	Москва	5.82	Санкт-Петер...	10.01	-4.19

Запрос 9. "Сравнение динамики продаж между Москвой и Санкт-Петербургом по аптекам":

- а. Посчитать продажи лекарств в Москве по аптекам и по месяцам;
- б. Посчитать продажи лекарств в Санкт-Петербурге по аптекам и по месяцам;
- с. Соединить таблицы по аптекам и по месяцам и найти разницу в процентах;

2-ой вариант

```
1  /*
2      Запрос 9. "Сравнение динамики продаж между Москвой и Санкт-Петербургом по аптекам":
3          а. Посчитать продажи лекарств в Москве по аптекам и по месяцам;
4          б. Посчитать продажи лекарств в Санкт-Петербурге по аптекам и по месяцам;
5          с. Соединить таблицы по аптекам и по месяцам и найти разницу в процентах;
6  */
7  WITH msk_sales AS (
8      SELECT
9          pharmacy_name,
10         drug,
11         TO_CHAR(TO_DATE(report_date, 'YYYY-MM-DD'), 'TMMonth') AS MONTH,
12         SUM(price * COUNT) AS total_sales,
13         ROUND(SUM(price * COUNT) * 100.0 / SUM(SUM(price * COUNT)) OVER (PARTITION BY pharmacy_name, TO_CHAR(TO_DATE(report_date, 'YYYY-MM-DD'), 'TMMonth')
14     FROM
15         pharma_orders
16     WHERE
17         city = 'Москва'
18     GROUP BY
19         pharmacy_name, drug, MONTH
```

pharmacy_name	drug	month	msk__perc_month	spb__perc_month	diff_perc_month
Столичная	Аквалор	April	6.30	12.35	-6.05
Столичная	Аквалор	February	5.07	9.13	-4.06
Столичная	Аквалор	June	20.46	0.00	20.46
Столичная	Аквалор	March	5.32	6.58	-1.26
Столичная	Аквалор	May	4.69	7.72	-3.03
Столичная	Аквамарис	April	10.96	5.97	4.99
Столичная	Аквамарис	February	9.01	2.92	6.09
Столичная	Аквамарис	June	3.77	0.00	3.77
Столичная	Аквамарис	March	4.49	4.95	-0.46

Запрос 10. Лекарства от насморка.

Выделяем препараты, начинающиеся со слова “аква” (с использованием оператора LIKE).

Приводим данные к нижнему регистру, группируем и подсчитываем общий объем продаж для каждого препарата.

Ранжируем по убыванию объема продаж и подсчитываем долю продаж каждого лекарства в общем объеме.

Run

PostgreSQL

```
1 /*
2     Запрос 10. Лекарства от насморка.
3     Выделяем препараты, начинающиеся со слова “аква” (с использованием оператора LIKE).
4     Приводим данные к нижнему регистру, группируем и подсчитываем общий объем продаж для каждого препарата.
5     Ранжируем по убыванию объема продаж и подсчитываем долю продаж каждого лекарства в общем объеме.
6 */
7 WITH lower_case_drugs AS (
8     SELECT
9         LOWER(drug) AS drug_name,
10        SUM(price * COUNT) AS total_sales
11    FROM
12        pharma_orders
13    WHERE
14        LOWER(drug) LIKE '%аква%'
15    GROUP BY
16        LOWER(drug)
17 ),
18 total_sales AS (
19     SELECT
20         SUM(total_sales) AS overall_sales
21    FROM
22        lower_case_drugs
23 )
24
25 SELECT
```

drug_name	total_sales	sales_percentage
аквафорте	771579	27.13
аквамарис	706240	24.83
аква-нормикс	685418	24.10
аквалор	681191	23.95

- Запрос 11. Кто наши клиенты.
- а. Вычисляем возраст клиентов на основе даты рождения с использованием функции для работы с датами;
 - б. Затем используем оператор CASE WHEN для расчета, кто наши клиенты;
 - в. Описываем каждую группу мужчин: мужчины младше 30 лет, мужчины от 30 до 45лет, и мужчины 45+ лет;
 - г. Описываем каждую группу женщин: женщины младше 30 лет, женщины от 30 до 45лет, и женщины 45+ лет;
 - д. Подсчитываем долю продаж на каждую из этих групп.

Run

PostgreSQL

PostgreSQL.45

PostgreSQL.46

```
1 /*
2     Запрос 11. Кто наши клиенты.
3         а. Вычисляем возраст клиентов на основе даты рождения с использованием функции для работы с датами;
4         б. Затем используем оператор CASE WHEN для расчета, кто наши клиенты;
5         в. Описываем каждую группу мужчин: мужчины младше 30 лет, мужчины от 30 до 45лет, и мужчины 45+ лет;
6         г. Описываем каждую группу женщин: женщины младше 30 лет, женщины от 30 до 45лет, и женщины 45+ лет;
7         д. Подсчитываем долю продаж на каждую из этих групп.
8 */
9 WITH age_gender_sales AS (
10     SELECT
11         C.customer_id,
12         C.gender,
13         SUM(po.price * po.count) AS total_sales,
14         EXTRACT(YEAR FROM AGE(TO_DATE(C.date_of_birth, 'YYYY-MM-DD')))) AS age
15     FROM
16         customers C
17     JOIN
18         pharma_orders po ON C.customer_id = po.customer_id
19     GROUP BY
20         C.customer_id, C.gender, C.date_of_birth
21 ),
22 grouped_sales AS (

```

age_group	group_sales	sales_share_percentage
1Мужчины младше 30 лет	1460626	13.41
2Мужчины от 30 до 45 лет	1199953	11.01
3Мужчины 45+ лет	2853001	26.19
4Женщины младше 30 лет	1262272	11.59
5Женщины от 30 до 45 лет	1344784	12.34
6Женщины 45+ лет	2774089	25.46

Запрос 11. Кто наши клиенты.

2-ой вариант

```
Run PostgreSQL PostgreSQL.45 PostgreSQL.46
1 /* 2-ой вариант
2     Запрос 11. Кто наши клиенты.
3         а. Вычисляем возраст клиентов на основе даты рождения с использованием функции для работы с датами;
4         б. Затем используем оператор CASE WHEN для расчета, кто наши клиенты;
5         в. Описываем каждую группу мужчин: мужчины младше 30 лет, мужчины от 30 до 45 лет, и мужчины 45+ лет;
6         г. Описываем каждую группу женщин: женщины младше 30 лет, женщины от 30 до 45 лет, и женщины 45+ лет;
7         е. Подсчитываем долю продаж на каждую из этих групп.
8 */
9 WITH age_gender_sales AS (
10     SELECT
11         C.customer_id,
12         C.gender,
13         SUM(po.price * po.count) AS total_sales,
14         EXTRACT(YEAR FROM AGE(TO_DATE(C.date_of_birth, 'YYYY-MM-DD')))) AS age
15     FROM
16         customers C
17     JOIN
18         pharma_orders po ON C.customer_id = po.customer_id
19     GROUP BY
20         C.customer_id, C.gender, C.date_of_birth
21 ),
22 grouped_sales AS (
```

age_group	group_sales	sales_share_percentage
3Мужчины 45+ лет	2853001	26.19
6Женщины 45+ лет	2774089	25.46
1Мужчины младше 30 лет	1460626	13.41
5Женщины от 30 до 45 лет	1344784	12.34
4Женщины младше 30 лет	1262272	11.59
2Мужчины от 30 до 45 лет	1199953	11.01

12_1. Разделить кампании на три группы по сумме покупок (NTILE).

Есть campaign_id, которые нужно разделить их на три группы по сумме покупок.

То есть, нужно сначала понять общую сумму в каждой кампании.

```
1  /*
2      1. Разделить кампании на три группы по сумме покупок (NTILE).
3      Есть campaign_id, которые нужно разделить их на три группы по сумме покупок.
4      То есть, нужно сначала понять общую сумму в каждой кампании.
5  */
6  WITH campaign_totals AS (
7      SELECT
8          campaign_id,
9          SUM(purchase_amount) AS total_purchase_amount
10     FROM
11         marketing_data
12     GROUP BY
13         campaign_id
14 )
15 SELECT
16     campaign_id,
17     total_purchase_amount,
18     NTILE(3) OVER (ORDER BY total_purchase_amount DESC) AS group_id
19 FROM
20     campaign_totals;
21
22
```

campaign_id	total_purchase_amount	group_id
191	6433.26	1
118	5507.09	1
116	5095.1104	1
157	5070.98	1
168	4868.25	1
111	4672.13	1

12_2. Найти последующую покупку внутри каждой рекламной кампании (LEAD).

```
1  /*
2      2. Найти последующую покупку внутри каждой рекламной кампании (LEAD).
3  */
4  SELECT
5      customer_id,
6      campaign_id,
7      purchase_amount,
8      purchase_date,
9      LEAD(purchase_amount) OVER (PARTITION BY campaign_id ORDER BY purchase_date) AS next_purchase_amount,
10     LEAD(purchase_date) OVER (PARTITION BY campaign_id ORDER BY purchase_date) AS next_purchase_date
11 FROM
12     marketing_data
13 ORDER BY
14     campaign_id, purchase_date;
15
16
17
18
```

customer_id	campaign_id	purchase_amount	purchase_date	next_purchase_a...	next_purchase_date
390	100	89.43	2023-04-02	416.29	2023-04-04
106	100	416.29	2023-04-04	866.85	2023-04-09
402	100	866.85	2023-04-09	743.71	2023-05-23
499	100	743.71	2023-05-23	403.21	2023-11-09
837	100	403.21	2023-11-09	275.99	2023-12-12
640	100	275.99	2023-12-12	367.36	2024-02-18
681	100	367.36	2024-02-18	926	2024-03-10

12_3. Найти 3 первых клиентов, привлеченных в рамках каждой кампании (LEAD).

```
1 /*
2     3. Найти 3 первых клиентов, привлеченных в рамках каждой кампании (LEAD).
3 */
4 WITH RankedCustomers AS (
5     SELECT
6         customer_id,
7         campaign_id,
8         purchase_amount,
9         purchase_date,
10        region,
11        channel,
12        age,
13        gender,
14        ROW_NUMBER() OVER (PARTITION BY campaign_id ORDER BY purchase_date) AS RANK
15     FROM
16         marketing_data
17 )
18 SELECT
19     customer_id,
20     campaign_id,
21     purchase_amount,
22     purchase_date,
```

customer_id	campaign_id	purchase_amount	purchase_date	region	channel	age	gender
390	100	89.43	2023-04-02	South	Offline	45	Male
106	100	416.29	2023-04-04	West	Online	57	Male
402	100	866.85	2023-04-09	North	Offline	18	Female
122	101	147.44	2023-01-10	South	Offline	59	Female
379	101	172.02	2023-05-29	South	Online	35	Male
943	101	623.87	2023-06-25	South	Online	42	Female
881	100	571.1	2023-11-10	East	Online	38	Male

```

1  /*
2      3. Найти 3 первых клиентов, привлеченных в рамках каждой кампании (LEAD).
3  */
4  WITH RankedCustomers AS (
5      SELECT
6          customer_id,
7          campaign_id,
8          purchase_amount,
9          purchase_date,
10         region,
11         channel,
12         age,
13         gender,
14         ROW_NUMBER() OVER (PARTITION BY campaign_id ORDER BY purchase_date) AS RANK
15     FROM
16         marketing_data
17 ),
18 Leads AS (
19     SELECT
20         customer_id,
21         campaign_id,
22         purchase_date,
23         LEAD(customer_id, 1) OVER (PARTITION BY campaign_id ORDER BY purchase_date) AS next_customer,
24         LEAD(customer_id, 2) OVER (PARTITION BY campaign_id ORDER BY purchase_date) AS next_customer_2
25     FROM

```

customer_id	campaign_id	purchase_date	next_customer	next_customer_2
390	100	2023-04-02	106	402
106	100	2023-04-04	402	NULL
402	100	2023-04-09	NULL	NULL
122	101	2023-01-10	379	943
379	101	2023-05-29	943	NULL
943	101	2023-06-25	NULL	NULL

12_4. Найти первого клиента, привлеченного в рамках каждой кампании (FIRST_VALUE).

```
1  /*
2      4. Найти первого клиента, привлеченного в рамках каждой кампании (FIRST_VALUE).
3  */
4  WITH FirstCustomers AS (
5      SELECT
6          customer_id,
7          campaign_id,
8          purchase_date,
9          FIRST_VALUE(customer_id) OVER (PARTITION BY campaign_id ORDER BY purchase_date) AS first_customer_id,
10         FIRST_VALUE(purchase_date) OVER (PARTITION BY campaign_id ORDER BY purchase_date) AS first_purchase_date
11     FROM
12         marketing_data
13 )
14 SELECT DISTINCT
15     campaign_id,
16     first_customer_id,
17     first_purchase_date
18 FROM
19     FirstCustomers
20 ORDER BY
21     campaign_id;
22
```

campaign_id	first_customer_id	first_purchase_date
100	390	2023-04-02
101	122	2023-01-10
102	884	2023-11-10
103	392	2023-05-25
104	388	2023-03-07
105	316	2023-03-09

12_5. Найти первых клиентов, совершивших покупки в каждом канале (FIRST_VALUE).

```
1 /*
2     5. Найти первых клиентов, совершивших покупки в каждом канале (FIRST_VALUE).
3 */
4 WITH FirstCustomers AS (
5     SELECT
6         customer_id,
7         channel,
8         purchase_date,
9         FIRST_VALUE(customer_id) OVER (PARTITION BY channel ORDER BY purchase_date) AS first_customer_id,
10        FIRST_VALUE(purchase_date) OVER (PARTITION BY channel ORDER BY purchase_date) AS first_purchase_date
11    FROM
12        marketing_data
13 )
14 SELECT DISTINCT
15     channel,
16     first_customer_id,
17     first_purchase_date
18 FROM
19     FirstCustomers
20 ORDER BY
21     channel;
22
```

channel	first_customer_id	first_purchase_date
Offline	103	2023-01-01
Online	436	2023-01-02

12_6. Найти последних клиентов, совершивших покупки в каждом канале (LAST_VALUE).

```
1  /*
2      6. Найти последних клиентов, совершивших покупки в каждом канале (LAST_VALUE).
3  */
4  WITH RankedPurchases AS (
5      SELECT
6          customer_id,
7          purchase_date,
8          channel,
9          LAST_VALUE(customer_id) OVER (
10             PARTITION BY channel
11             ORDER BY purchase_date
12             ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING
13         ) AS last_customer
14     FROM
15         marketing_data -- purchases
16 )
17
18 SELECT DISTINCT
19     channel,
20     last_customer
21 FROM
22     RankedPurchases
23 WHERE
24     last_customer IS NOT NULL;
25
26
```

channel	last_customer
Online	570
Offline	566

12_7. Если у тебя с LAST_VALUE будет такая же проблема, как на уроке, то используй FIRST_VALUE с сортировкой даты в обратную сторону.

Найти 2 последних клиентов, привлеченных в рамках каждой кампании (LAG).

```
1 /*
2  7. Если у тебя с LAST_VALUE будет такая же проблема, как на уроке, то используй
3  FIRST_VALUE с сортировкой даты в обратную сторону.
4  Найти 2 последних клиентов, привлеченных в рамках каждой кампании (LAG).
5 */
6 WITH RankedCustomers AS (
7     SELECT
8         customer_id,
9         campaign_id,
10        purchase_date,
11        LAG(customer_id, 1) OVER (PARTITION BY campaign_id ORDER BY purchase_date) AS previous_customer,
12        LAG(customer_id, 2) OVER (PARTITION BY campaign_id ORDER BY purchase_date) AS second_previous_customer
13    FROM
14        marketing_data --campaigns
15 )
16
17 SELECT
18     campaign_id,
19     customer_id AS current_customer,
20     previous_customer,
21     second_previous_customer
22 FROM
23     RankedCustomers
24 WHERE
25     previous_customer IS NOT NULL OR second_previous_customer IS NOT NULL;
26
27
28
```

campaign_id	current_customer	previous_customer	second_previous_customer
100	106	390	NULL
100	402	106	390
100	499	402	106
100	837	499	402

12_8. Найти последнего клиента, привлеченного внутри каждой кампании (LAST_VALUE).

Если у тебя с LAST_VALUE будет такая же проблема, как на уроке, то используй FIRST_VALUE с сортировкой даты в обратную сторону.

```
1 /*
2  8. Найти последнего клиента, привлеченного внутри каждой кампании (LAST_VALUE).
3  Если у тебя с LAST_VALUE будет такая же проблема, как на уроке, то используй
4  FIRST_VALUE с сортировкой даты в обратную сторону.
5 */
6 WITH LastCustomers AS (
7     SELECT
8         customer_id,
9         campaign_id,
10        purchase_date,
11        region,
12        channel,
13        age,
14        gender,
15        ROW_NUMBER() OVER (PARTITION BY campaign_id ORDER BY purchase_date DESC) AS rn
16    FROM
17        marketing_data
18 )
19 SELECT
20     customer_id,
21     campaign_id,
22     purchase_date,
23     region,
24     channel,
25     age,
26     gender
27 FROM
28     LastCustomers
29 WHERE
30     rn = 1;
```

customer_id	campaign_id	purchase_date	region	channel	age	gender
470	100	2024-03-10	East	Offline	47	Female
943	101	2023-06-25	South	Online	42	Female


```

1  /*
2   8. Найти последнего клиента, привлеченного внутри каждой кампании (LAST_VALUE).
3   Если у тебя с LAST_VALUE будет такая же проблема, как на уроке, то используй
4   FIRST_VALUE с сортировкой даты в обратную сторону.
5  */
6  SELECT DISTINCT
7      customer_id,
8      campaign_id,
9      purchase_date,
10     region,
11     channel,
12     age,
13     gender,
14     LAST_VALUE(customer_id) OVER (PARTITION BY campaign_id ORDER BY purchase_date
15                                   ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS last_customer
16 FROM
17     marketing_data
18 ORDER BY
19     campaign_id, purchase_date;
20
21

```

customer_id	campaign_id	purchase_date	region	channel	age	gender	last_customer
390	100	2023-04-02	South	Offline	45	Male	470
106	100	2023-04-04	West	Online	57	Male	470
402	100	2023-04-09	North	Offline	18	Female	470
499	100	2023-05-23	North	Online	59	Male	470
837	100	2023-11-09	West	Online	59	Female	470
640	100	2023-12-12	South	Online	61	Female	470
681	100	2024-02-18	South	Offline	30	Male	470
470	100	2024-03-10	East	Offline	47	Female	470

12_9. Найти последних клиентов, совершивших покупки в каждом регионе

```

1 /*
2     9. Найти последних клиентов, совершивших покупки в каждом регионе (LAST_VALUE).
3 */
4 SELECT DISTINCT
5     customer_id,
6     region,
7     purchase_amount,
8     purchase_date,
9     channel,
10    age,
11    gender,
12    LAST_VALUE(customer_id) OVER (PARTITION BY region ORDER BY purchase_date
13                                   ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS last_customer
14 FROM
15     marketing_data
16 ORDER BY
17     region, purchase_date;
18
19

```

customer_id	region	purchase_amount	purchase_date	channel	age	gender	last_customer
103	East	980.23	2023-01-01	Offline	31	Female	793
271	East	637.07	2023-01-04	Online	19	Female	793
72	East	85.1	2023-01-06	Offline	24	Female	793
615	East	160.38	2023-01-09	Online	20	Male	793
459	East	897.82	2023-01-14	Online	28	Female	793
100	East	180.6	2023-01-17	Online	63	Female	793
662	East	285.8	2023-01-21	Offline	55	Male	793
344	East	129.43	2023-01-24	Online	56	Male	793

```

1  /*
2   9. Найти последних клиентов, совершивших покупки в каждом регионе (LAST_VALUE).
3  */
4  WITH LastCustomers AS (
5      SELECT
6          customer_id,
7          region,
8          purchase_amount,
9          purchase_date,
10         channel,
11         age,
12         gender,
13         ROW_NUMBER() OVER (PARTITION BY region ORDER BY purchase_date DESC) AS rn
14     FROM
15         marketing_data
16 )
17 SELECT
18     customer_id,
19     region,
20     purchase_amount,
21     purchase_date,
22     channel,
23     age,
24     gender
25 FROM
26     LastCustomers
27 WHERE
28     rn = 1;

```

#	customer_id	region	purchase_am...	purchase_date	channel	age	gender
	793	East	682.85	2024-05-11	Offline	56	Male
	323	North	987.85	2024-05-08	Online	40	Male
	570	South	788.48	2024-05-14	Online	42	Male
	735	West	226.07	2024-05-12	Online	61	Male

12_10. Сделать поля с 5 первыми покупками внутри каждой рекламной кампании и проследить динамику.

Похожую задачу мы решали на уроке.

```
1 /*
2     10. Сделать поля с 5 первыми покупками внутри каждой рекламной кампании и проследить динамику.
3 */
4 WITH RankedPurchases AS (
5     SELECT
6         customer_id,
7         campaign_id,
8         purchase_amount,
9         purchase_date,
10        region,
11        channel,
12        age,
13        gender,
14        ROW_NUMBER() OVER (PARTITION BY campaign_id ORDER BY purchase_date) AS purchase_rank
15    FROM
16        marketing_data
17 )
18 SELECT
19     customer_id,
20     campaign_id,
21     purchase_amount,
22     purchase_date,
23     region,
24     channel,
25     age,
26     gender
27 FROM
28     RankedPurchases
29 WHERE
30     purchase_rank <= 5
31 ORDER BY
32     campaign id, purchase rank;
```

customer_id	campaign_id	purchase_amount	purchase_date	region	channel	age	gender
390	100	89.43	2023-04-02	South	Offline	45	Male
106	100	416.29	2023-04-04	West	Online	57	Male

```

1 /*
2     10. Сделать поля с 5 первыми покупками внутри каждой рекламной кампании и проследить динамику.
3 */
4 WITH ranked_purchases AS (
5     SELECT
6         *,
7         ROW_NUMBER() OVER (PARTITION BY campaign_id ORDER BY purchase_date) AS rn
8     FROM
9         marketing_data
10 )
11 SELECT
12     customer_id,
13     campaign_id,
14     purchase_amount,
15     purchase_date,
16     region,
17     channel,
18     age,
19     gender
20 FROM
21     ranked_purchases
22 WHERE
23     rn <= 5
24 ORDER BY
25     campaign_id, purchase_date;
--

```

customer_id	campaign_id	purchase_amount	purchase_date	region	channel	age	gender
390	100	89.43	2023-04-02	South	Offline	45	Male
106	100	416.29	2023-04-04	West	Online	57	Male
402	100	866.85	2023-04-09	North	Offline	18	Female
499	100	743.71	2023-05-23	North	Online	59	Male
837	100	403.21	2023-11-09	West	Online	59	Female

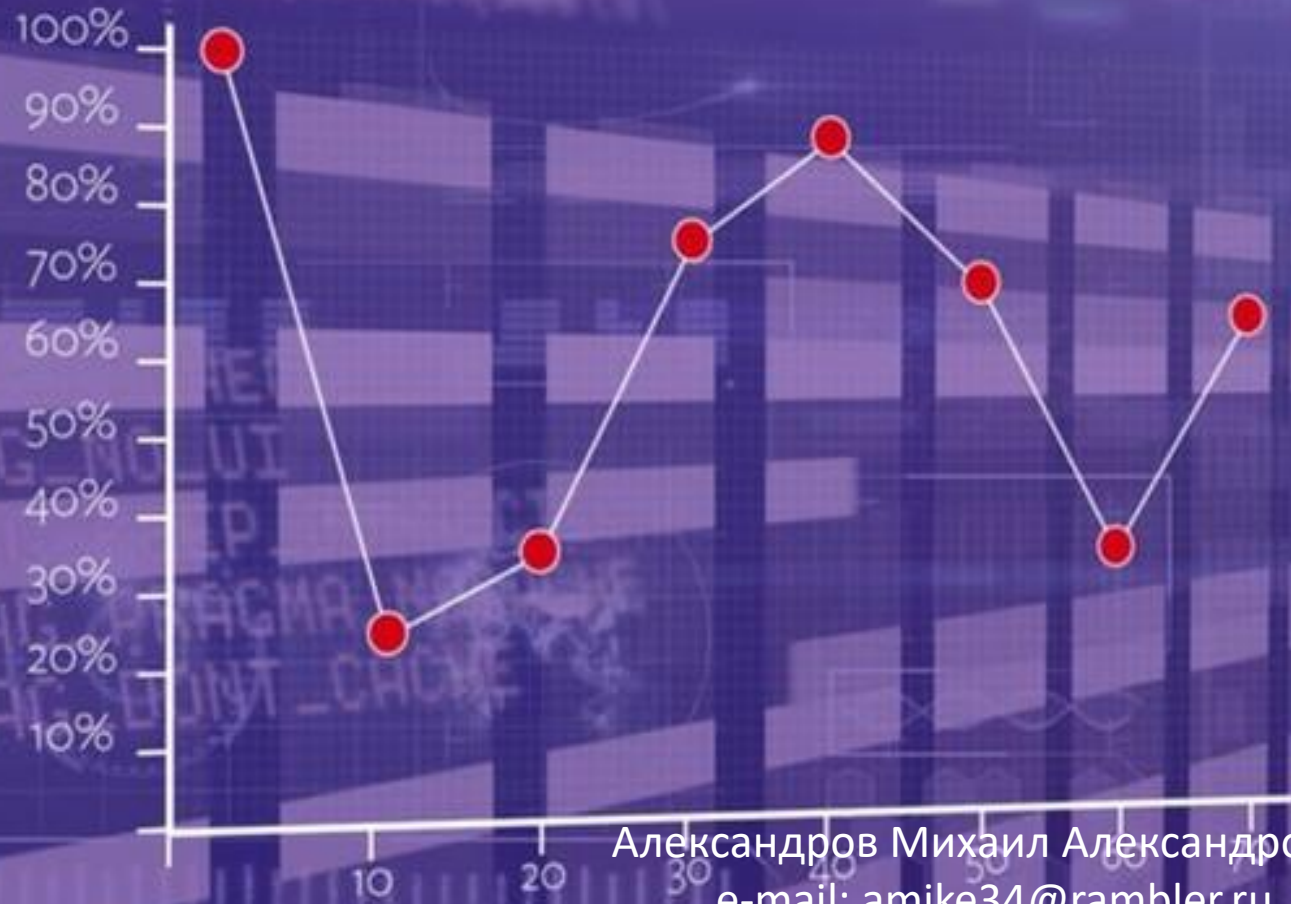
```

1  /*
2     10. Сделать поля с 5 первыми покупками внутри каждой рекламной кампании и проследить динамику.
3  */
4  SELECT
5     md.customer_id, md.campaign_id, md.purchase_amount, md.purchase_date, md.region,
6     md.channel, md.age, md.gender
7  FROM
8     marketing_data md
9  WHERE
10     (md.campaign_id, md.purchase_date)
11
12  IN (
13     SELECT
14         campaign_id, purchase_date
15     FROM
16         marketing_data
17     WHERE
18         campaign_id = md.campaign_id
19     ORDER BY
20         purchase_date
21     LIMIT 5
22  )
23  ORDER BY
24     md.campaign_id, md.purchase_date;
25
--

```

customer_id	campaign_id	purchase_amount	purchase_date	region	channel	age	gender
390	100	89.43	2023-04-02	South	Offline	45	Male
106	100	416.29	2023-04-04	West	Online	57	Male
402	100	866.85	2023-04-09	North	Offline	18	Female
499	100	743.71	2023-05-23	North	Online	59	Male
837	100	403.21	2023-11-09	West	Online	59	Female

Спасибо за внимание!



Александров Михаил Александрович
e-mail: amike34@rambler.ru
@amike34