

1. Получено 10 строк

/* postgres> Script-1 dogs postgres> Script-2 X

/* Вывести распределение (количество) клиентов по сферам деятельности, отсортировав результат по убыванию количества */

select job_industry_category, count(customer_id) as cnt_customer

from customer c

group by job_industry_category

order by count(customer_id) desc;

customer 1 X

Введите SQL выражение чтобы отфильтровать результаты

job_industry_category

cnt_customer

1	Manufacturing	799
2	Financial Services	774
3	n/a	656
4	Health	602
5	Retail	358
6	Property	267
7	IT	223
8	Entertainment	136
9	Agriculture	113
10	Telecommunications	72

2. Получено 120 строк

/* postgres> Script-1 dogs postgres> Script-2 X

/* Найти сумму транзаций за каждый месяц по сферам деятельности, отсортировав по месяцам и по сфере деятельности. */

select date_trunc('month', to_date(transaction_date, 'DD.MM.YYYY'))::date as partition_month, c.job_industry_category, sum(list_price) as sum_transactions

from "transaction" t

join customer c

on t.customer_id = c.customer_id

group by date_trunc('month', to_date(transaction_date, 'DD.MM.YYYY'))::date, c.job_industry_category

order by date_trunc('month', to_date(transaction_date, 'DD.MM.YYYY'))::date, c.job_industry_category;

customer 1 X

Введите SQL выражение чтобы отфильтровать результаты

partition_month

job_industry_category

sum_transactions

1	2017-01-01	Agriculture	43 513,812
2	2017-01-01	Entertainment	64 009,934
3	2017-01-01	Financial Services	366 383,78
4	2017-01-01	Health	286 860,44
5	2017-01-01	IT	107 783,414
6	2017-01-01	Manufacturing	365 232,38
7	2017-01-01	n/a	316 819,72
8	2017-01-01	Property	100 686,97
9	2017-01-01	Retail	182 375,72
10	2017-01-01	Telecommunications	31 210,2
11	2017-02-01	Agriculture	60 016,81
12	2017-02-01	Entertainment	63 965,99
13	2017-02-01	Financial Services	375 961,8
14	2017-02-01	Health	268 525,78
15	2017-02-01	IT	93 961,79
16	2017-02-01	Manufacturing	389 260,53
17	2017-02-01	n/a	260 711,89
18	2017-02-01	Property	112 255,41
19	2017-02-01	Retail	146 107,58
20	2017-02-01	Telecommunications	27 678,2
21	2017-03-01	Agriculture	49 048,492
22	2017-03-01	Entertainment	77 122,44
23	2017-03-01	Financial Services	322 268,28
24	2017-03-01	Health	290 575,88
25	2017-03-01	IT	100 212,4
26	2017-03-01	Manufacturing	377 784,72
27	2017-03-01	n/a	278 553,9
28	2017-03-01	Property	125 143,59
29	2017-03-01	Retail	172 366,56
30	2017-03-01	Telecommunications	24 541,277

3. Получено 7 строк. Если NULL не имеет смысловой нагрузки, можно добавить условие на brand is not null.

/* postgres> Script-1 dogs postgres> Script-2 X

/* Вывести количество онлайн-заказов для всех брендов в рамках подтвержденных заказов клиентов из сферы IT. */

select brand, count(transaction_id) as cnt_orders

from "transaction" t

join customer c

on t.customer_id = c.customer_id and c.job_industry_category = 'IT'

where t=1

and t.online_order = 'True'

and t.order_status = 'Approved'

group by brand

order by brand;

transaction 1 X

Введите SQL выражение чтобы отфильтровать результаты

brand

cnt_orders

1		8
2	Giant Bicycles	89
3	Norco Bicycles	92
4	OHM Cycles	78
5	Solex	101
6	Trek Bicycles	82
7	WeareA2B	90

4. Без оконных функций 3494 строки.

/* Найдите по всем клиентам сумму всех транзакций (list_price), максимум, минимум и количество транзакций. Выполните двумя способами: используя только group by и используя только оконные функции. */

```
select
  customer_id,
  sum(list_price) as sum_list_price,
  max(list_price) as max_list_price,
  min(list_price) as min_list_price,
  count(t.transaction_id) as cnt_transactions
from "transaction" t
group by customer_id
order by sum(t.list_price) desc, count(t.transaction_id) desc;
```

transaction 1 X

Введите SQL выражение чтобы отфильтровать результаты

	customer_id	sum_list_price	max_list_price	min_list_price	cnt_transactions
1	2183	19 071,322	2 005,66	230,91	14
2	1129	18 349,27	1 992,93	290,62	13
3	1597	18 052,68	2 091,47	360,4	12
4	941	17 898,459	2 091,47	1 057,51	10
5	2788	17 258,94	2 083,94	183,86	11
6	936	17 160,24	2 005,66	183,86	12
7	1887	17 133,922	2 091,47	688,63	11
8	1302	17 035,83	1 977,36	71,16	13
9	1140	16 199,24	2 083,94	183,86	13
10	2309	16 122,341	2 091,47	290,62	12
11	729	15 825,999	2 091,47	586,45	10
12	1103	15 447,92	1 977,36	230,91	12
13	1317	15 370,81	2 091,47	569,56	9
14	2874	15 091,91	2 005,66	544,05	11
15	2762	15 071,26	1 997,68	183,86	10
16	255	14 949,91	1 810	441,49	12
17	1672	14 908,461	1 977,36	363,01	13
18	1329	14 818,119	1 793,43	543,39	11
19	322	14 782,2295	2 005,66	12,01	12
20	3048	14 756,089	1 890,39	543,39	13
21	2860	14 673,35	1 812,75	175,89	10
22	999	14 662,139	2 091,47	495,72	10
23	1460	14 633,24	2 091,47	1 228,07	8
24	2894	14 621,73	2 083,94	1 129,13	9
25	3060	14 609,221	1 992,93	533,51	11
26	445	14 593,221	1 894,19	543,39	11
27	2476	14 578,691	1 807,45	12,01	14
28	2464	14 574,661	1 843,03	726,01	12

Обновить Save Cancel Экспорт данных ... 200 3494

С оконными функциями 20 тысяч строк.

/* Найдите по всем клиентам сумму всех транзакций (list_price), максимум, минимум и количество транзакций. Выполните двумя способами: используя только group by и используя только оконные функции. */

```
select
  customer_id,
  sum(list_price) over (partition by customer_id) as sum_list_price,
  max(list_price) over (partition by customer_id) as max_list_price,
  min(list_price) over (partition by customer_id) as min_list_price,
  count(t.transaction_id) over (partition by customer_id) as cnt_transactions
from "transaction" t
order by sum_list_price desc, cnt_transactions desc;
```

transaction 1 X

Введите SQL выражение чтобы отфильтровать результаты

	customer_id	sum_list_price	max_list_price	min_list_price	cnt_transactions
1	2183	19 071,32	2 005,66	230,91	14
2	2183	19 071,32	2 005,66	230,91	14
3	2183	19 071,32	2 005,66	230,91	14
4	2183	19 071,32	2 005,66	230,91	14
5	2183	19 071,32	2 005,66	230,91	14
6	2183	19 071,32	2 005,66	230,91	14
7	2183	19 071,32	2 005,66	230,91	14
8	2183	19 071,32	2 005,66	230,91	14
9	2183	19 071,32	2 005,66	230,91	14
10	2183	19 071,32	2 005,66	230,91	14
11	2183	19 071,32	2 005,66	230,91	14
12	2183	19 071,32	2 005,66	230,91	14
13	2183	19 071,32	2 005,66	230,91	14
14	2183	19 071,32	2 005,66	230,91	14
15	1129	18 349,271	1 992,93	290,62	13
16	1129	18 349,271	1 992,93	290,62	13
17	1129	18 349,271	1 992,93	290,62	13
18	1129	18 349,271	1 992,93	290,62	13
19	1129	18 349,271	1 992,93	290,62	13
20	1129	18 349,271	1 992,93	290,62	13
21	1129	18 349,271	1 992,93	290,62	13
22	1129	18 349,271	1 992,93	290,62	13
23	1129	18 349,271	1 992,93	290,62	13
24	1129	18 349,271	1 992,93	290,62	13
25	1129	18 349,271	1 992,93	290,62	13
26	1129	18 349,271	1 992,93	290,62	13
27	1129	18 349,271	1 992,93	290,62	13

Обновить Save Cancel Экспорт данных ... 200 20 000

5. Максимум

Минимум

The screenshot shows a PostgreSQL IDE with two tabs: "Script-1" and "Script-2". The "Script-2" tab is active, displaying a SQL query:

```
/* Найти имена и фамилии клиентов с минимальной/максимальной
 * Напишите отдельные запросы для минимальной и максимальной */

select c.first_name, c.last_name, sum(t.list_price)
from "transaction" t
join customer c
on t.customer_id = c.customer_id
where l=1
and t.list_price is not null
group by c.first_name, c.last_name
order by sum(t.list_price)
limit 1;
```

Below the script, the results of the query are displayed in a table titled "customer 1":

	first_name	last_name	sum
1	Hamlen	Slograve	60.34

6. Получено 3494 строки.

The screenshot shows the PostgreSQL IDE interface. The top toolbar includes icons for opening files, saving, and running queries. The main editor window displays a SQL query:

```

/* Вывести только самые первые транзакции клиентов. Решить с помощью оконных функций. */

select distinct tmp_table.transaction_id
from
(select distinct transaction_id,
 row_number() over (partition by customer_id order by date_trunc('month', to_date(transaction_date, 'DD-MM-YYYY'))::date) as rv_list
 from "transaction" t) as tmp_table
where rv_list = 1;

```

Below the query editor, the results of the query are displayed in a table with the following columns:

transaction_id
1
11
12
26
30
32
34
37
51
54
59
61
64
71
74
85
100
109
119
123
128
131
138
140
143
147
148
170

The bottom status bar indicates that 200 rows were received (200 строк получено - 0.02s, 2024-02-25 19:28:17).

7. Максимальный интервал 357 дней у Susanetta.

Script-1

dogs

Script-2

```
with
intervals_dt as
(select distinct c.first_name, c.last_name, c.job_title,
LEAD(to_date(transaction_date, 'DD.MM.YYYY')) OVER (PARTITION BY t.customer_id ORDER BY to_date(transaction_date, 'DD.MM.YYYY') - to_date(transaction_date, 'DD.MM.YYYY') AS interval_dt
from "transaction" t
left join customer c
on t.customer_id = c.customer_id),
max_interval as
(select max(interval_dt) as interval_dt
from intervals_dt)
select distinct c.first_name, c.last_name, c.job_title, c.interval_dt
from intervals_dt c
join max_interval m
on c.interval_dt = m.interval_dt;
```

customer 1

Введите SQL выражение чтобы отфильтровать результаты

	first_name	last_name	job_title	interval_dt
1	Susanetta		Legal Assistant	357

Значение

Susanetta