

## 1. Клиенты по сферам деятельности

DBeaver 25.2.4 - <postgres 2> Script-2

Файл Редактирование Навигация Поиск Редактор SQL База данных Окна Справка

Auto

postgres 2 <postgres> Script <postgres 3> Sc... \*<postgres 2> S...

```
select job_industry_category, count(*) as client_industry
from customers c
group by c.job_industry_category
order by client_industry desc
```

customers 1

select job\_industry\_category | Введите SQL выражение чтобы отфильтровать результаты

Таблица	A-Z job_industry_category	123 client_industry
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	Argiculture	113
10	Telecommunications	72
11	[NULL]	1

## 2. Общая сумма доходов

```
postgres 2 <postres> Script <postres 3> Sc... *<postres 2> S... customers product orders c
    select extract(year from o.order_date) as year,
           extract(month from o.order_date) as month,
           c.job_industry_category,
           sum(p.list_price * oi.quantity) as total_sum
      from orders o
     join order_items oi on o.order_id = oi.order_id
     join product p on oi.product_id = p.product_id
     join customers c on o.customer_id = c.customer_id
    where o.order_status = 'Approved'
   group by year, month, c.job_industry_category
  order by year, month, c.job_industry_category
```

customers 1 ×

select extract(year from o.order\_date) as year, extract(month from o.order\_date) as month, c.job\_industry\_category, sum(p.list\_price \* oi.quantity) as total\_sum

	year	month	job_industry_cat	total_sum
1	2 017	1	Argiculture	481 158,3300000001
2	2 017	1	Entertainment	632 572,1600000005
3	2 017	1	Financial Services	3 987 494,0399999963
4	2 017	1	Health	3 030 781,3699999996
5	2 017	1	IT	1 257 043,2500000002
6	2 017	1	Manufacturing	3 775 077,7300000002
7	2 017	1	n/a	3 891 784,1899999999
8	2 017	1	Property	907 482,1700000005
9	2 017	1	Retail	2 053 150,6699999997
10	2 017	1	Telecommunications	368 980,79
11	2 017	1	[NULL]	64 759,92
12	2 017	2	Argiculture	740 904,13

Обновить Save Cancel Экспорт данных ... 200 122 122 строк п

## 3. Уникальные заказы от IT-клиентов

```
postgres 2 <postres> Script <postres 3> Sc... *<postres 2> S... customers product
```

```
select p.brand, count(case when o.online_order = true and o.order_status = 'Approved' and c.job_industry_category = 'IT' then o.order_id end) as online_orders
  from product p
 left join order_items oi on p.product_id = oi.product_id
 left join orders o on oi.order_id = o.order_id
 left join customers c on o.customer_id = c.customer_id
 group by p.brand
 order by p.brand
```

product 1 ×

select p.brand, count(distinct o.order\_id) as online\_orders

brand	online_orders
Giant Bicycles	656
Norco Bicycles	3 306
OHM Cycles	2 834
Solex	2 991
Trek Bicycles	3 881
WeareA2B	2 775
	3 098

## 4.1. GROUP BY по подтвержденным заказам клиентов

```
postgres 2 <postres> Script <postres 3> Scri... *<postres 2> Sc... customers product orders order_it...
```

```

select c.customer_id, sum(oi.quantity * p.list_price) as total_sum,
       max(oi.quantity * p.list_price) as max_order,
       min(oi.quantity * p.list_price) as min_order,
       count(distinct o.order_id) as orders_count,
       avg(oi.quantity * p.list_price) as avg_order
  from customers c
  join orders o on c.customer_id = o.customer_id
  join order_items oi on o.order_id = oi.order_id
  join product p on oi.product_id = p.product_id
 where o.order_status = 'Approved'
 group by c.customer_id
 order by total_sum desc, orders_count desc

```

customers 1 ×

select c.customer\_id, sum(oi.quantity \* p.list\_price) as total\_sum ... Введите SQL выражение чтобы отфильтровать результаты

	customer_id	total_sum	max_order	min_order	orders_count	avg_order
1	2 659	297 691,35	16 369	60,05	12	3 235,7755434783
2	637	286 565,12	17 796,24	12,01	13	3 048,565106383
3	3 451	279 682,2199999999	16 865,73	48,04	9	2 715,3613592233
4	2 476	255 260,23	14 957,28	60,05	14	2 686,9497894737
5	1 611	246 483,25	10 945,26	96,08	12	3 286,4433333333
6	1 405	245 164,6199999998	10 945,26	24,02	11	2 291,2581308411
7	298	231 090,75	18 903,9	24,02	11	2 567,675
8	3 326	227 676,33	17 508,87	12,01	13	2 069,7848181818
9	2 358	226 872,5499999999	11 665,36	36,03	11	2 120,3042056075
10	289	226 393,8899999999	15 943,44	24,02	10	2 156,1322857143
11	1 129	220 194,51	20 914,7	1 743,72	13	10 485,4528571429
12	1 103	218 985,07	14 347,44	84,07	12	3 841,8433333333
13	2 459	218 141,93	15 486,3	60,05	11	2 988,2456164384

Обновить Save Cancel Экспорт данных ... 200 3 493 ... 3493 строк получе...

## 4.2. Оконная функция, по подтвержденным заказам клиентов

```
postgres 2 <postres> Script <postres 3> Scri... *<postres 2> Sc... customers product orders order_it...
```

```

with orders_total as (
  select o.customer_id, o.order_id, sum(oi.quantity * p.list_price) as order_total
    from orders o
   join order_items oi on o.order_id = oi.order_id
   join product p on oi.product_id = p.product_id
  where o.order_status = 'Approved'
 group by o.customer_id, o.order_id
)
select distinct customer_id,
           sum(order_total) over (partition by customer_id) as total_sum,
           max(order_total) over (partition by customer_id) as max_order,
           min(order_total) over (partition by customer_id) as min_order,
           count(*) over (partition by customer_id) as orders_count,
           avg(order_total) over (partition by customer_id) as avg_order
  from orders_total
 order by total_sum desc, orders_count desc

```

orders 1 ×

with orders\_total as ( select o.customer\_id, o.order\_id, ... Введите SQL выражение чтобы отфильтровать результаты

	customer_id	total_sum	max_order	min_order	orders_count	avg_order
1	2 659	297 691,35	53 966,6	983,55	12	24 807,6125
2	637	286 565,12	43 173,28	5 396,66	13	22 043,4707692308
3	3 451	279 682,22	48 569,94	2 455,77	9	31 075,8022222222
4	2 476	255 260,23	43 173,28	3 878,52	14	18 232,8735714286
5	1 611	246 483,25	53 966,6	3 190,14	12	20 540,2708333333
6	1 405	245 164,62	53 966,6	3 216,84	11	22 287,6927272727
7	298	231 090,75	43 173,28	1 557,04	11	21 008,25
8	3 326	227 676,33	43 173,28	5 396,66	13	17 513,5638461538
9	2 358	226 872,55	53 966,6	6 210,81	11	20 624,7772727273
10	289	226 393,89	53 966,6	1 703,77	10	22 639,389

Обновить Save Cancel Экспорт данных ... 200 3 493 ... 3493 строк получе...

#### 4.3. GROUP BY по всем заказам

```
postgres 2 <postgres> Script <postgres 3> Sc... *<postgres 2> S... X customers product orders order_items
▶
  select customer_id, sum(order_total) as total_sum, max(order_total) as max_order,
         min(order_total) as min_order, count(order_total) as orders_count,
         avg(order_total) as avg_order
    from (
      select o.customer_id, o.order_id, sum(oi.quantity * p.list_price) as order_total
        from orders o
       join order_items oi on o.order_id = oi.order_id
       join product p on oi.product_id = p.product_id
      group by o.customer_id, o.order_id
    ) as orders_per_client
   group by customer_id
  order by total_sum desc, orders_count desc
```

orders 1

	customer_id	total_sum	max_order	min_order	orders_count	avg_order
1	2 659	297 691,35	53 966,6	983,55	12	24 807,6125
2	637	286 565,12	43 173,28	5 396,66	13	22 043,4707692308
3	3 451	279 682,22	48 569,94	2 455,77	9	31 075,8022222222
4	2 476	255 260,23	43 173,28	3 878,52	14	18 232,8735714286
5	1 611	246 483,25	53 966,6	3 190,14	12	20 540,2708333333
6	1 405	245 164,62	53 966,6	3 216,84	11	22 287,6927272727
7	298	231 090,75	43 173,28	1 557,04	11	21 008,25
8	3 326	227 676,33	43 173,28	5 396,66	13	17 513,5638461538
9	2 358	226 872,55	53 966,6	6 210,81	11	20 624,7772727273
10	289	226 393,89	53 966,6	1 703,77	10	22 639,389
11	714	224 615,18	43 173,28	2 542,44	11	20 419,5618181818
12	1 129	220 194,51	36 690	6 501,12	13	16 938,0392307692
13	1 103	218 985,07	48 569,94	1 292,84	12	18 248,7558333333

#### 4.4. Окночная функция, по всем заказам

```
postgres 2 <postgres> Script <postgres 3> Sc... *<postgres 2> S... X customers product orders order_items
▶
  with orders_total as (
    select o.customer_id, o.order_id, sum(oi.quantity * p.list_price) as order_total
      from orders o
     join order_items oi on o.order_id = oi.order_id
     join product p on oi.product_id = p.product_id
    group by o.customer_id, o.order_id
  )
  select distinct customer_id,
             sum(order_total) over (partition by customer_id) as total_sum,
             max(order_total) over (partition by customer_id) as max_order,
             min(order_total) over (partition by customer_id) as min_order,
             count(*) over (partition by customer_id) as orders_count,
             avg(order_total) over (partition by customer_id) as avg_order
    from orders_total
   order by total_sum desc, orders_count desc
```

orders 1

	customer_id	total_sum	max_order	min_order	orders_count	avg_order
1	2 659	297 691,35	53 966,6	983,55	12	24 807,6125
2	637	286 565,12	43 173,28	5 396,66	13	22 043,4707692308
3	3 451	279 682,22	48 569,94	2 455,77	9	31 075,8022222222
4	2 476	255 260,23	43 173,28	3 878,52	14	18 232,8735714286
5	1 611	246 483,2500000001	53 966,6	3 190,14	12	20 540,2708333333
6	1 405	245 164,62	53 966,6	3 216,84	11	22 287,6927272727
7	298	231 090,75	43 173,28	1 557,04	11	21 008,25
8	3 326	227 676,33	43 173,28	5 396,66	13	17 513,5638461538
9	2 358	226 872,55	53 966,6	6 210,81	11	20 624,7772727273
10	289	226 393,89	53 966,6	1 703,77	10	22 639,389
11	714	224 615,10	43 172,20	2 512,44	11	20 419,5610101010

## 5. Топ-3 МИН и топ-3 МАКС

```
postgres 2 <postgres> Script <postgres 3> Scri... *<postgres 2> Sc... customers product orders
with customers_total as (
    select c.customer_id, c.first_name, c.last_name,
    coalesce(sum(oi.quantity * p.list_price), 0) as total_sum
    from customers c
    left join orders o on c.customer_id = o.customer_id
    left join order_items oi on o.order_id = oi.order_id
    left join product p on oi.product_id = p.product_id
    group by c.customer_id, c.first_name, c.last_name
),
ranked as (
    select first_name, last_name, total_sum,
    row_number() over (order by total_sum) as rank_min,
    row_number() over (order by total_sum desc) as rank_max
    from customers_total
)
select first_name, last_name, total_sum, 'min' as type
from ranked where rank_min <= 3
union all
select first_name, last_name, total_sum, 'max' as type
from ranked where rank_max <= 3
order by type, total_sum |
```

Результат 1 ×

with customers\_total as ( select c.customer\_id, c.first\_name, c.last\_name, coalesce(sum(oi.quantity \* p.list\_price), 0) as total\_sum from customers c left join orders o on c.customer\_id = o.customer\_id left join order\_items oi on o.order\_id = oi.order\_id left join product p on oi.product\_id = p.product\_id group by c.customer\_id, c.first\_name, c.last\_name ), ranked as ( select first\_name, last\_name, total\_sum, row\_number() over (order by total\_sum) as rank\_min, row\_number() over (order by total\_sum desc) as rank\_max from customers\_total ) select first\_name, last\_name, total\_sum, 'min' as type from ranked where rank\_min <= 3 union all select first\_name, last\_name, total\_sum, 'max' as type from ranked where rank\_max <= 3 order by type, total\_sum |

	AZ first_name	AZ last_name	123 total_sum	AZ type
1	Ruprecht	Loreit	279 682,2199999999	max
2	Mercy	Wilsons	286 565,1200000001	max
3	Konstance	Elgey	297 691,3499999999	max
4	Murielle	Klimkov	0	min
5	Thadeus	Buxsey	0	min
6	Sabina	Tarbin	0	min

Обновить Save Cancel Экспорт данных ... 200 6 6 строк по

## 6. Вторые транзакции клиентов

```
postgres 2 <postgres> Script <postgres 3> Scri... *<postgres 2> Sc... customers product orders
select c.first_name, c.last_name, sub.order_id, sub.order_date, sub.total_sum
from (
    select o.order_id, o.customer_id, o.order_date, sum(oi.quantity * p.list_price) as total_sum,
    row_number() over (partition by o.customer_id order by o.order_date) as rank
    from orders o
    inner join order_items oi on o.order_id = oi.order_id
    inner join product p on oi.product_id = p.product_id
    where o.order_status = 'Approved'
    group by o.order_id, o.customer_id, o.order_date
) sub
inner join customers c on sub.customer_id = c.customer_id
where sub.rank = 2 |
```

customers(+1) ×

select c.first\_name, c.last\_name, sub. | Введите SQL выражение чтобы отфильтровать результаты

	AZ first_name	AZ last_name	123 order_id	order_date	123 total_sum
1	Laraine	Medendorp	13 424	2017-02-21	6 617,5
2	Eli	Bockman	6 743	2017-06-11	5 268,48
3	Arlin	Dearle	15 188	2017-03-24	5 375,26
4	Talbot		14 648	2017-06-18	9 016,92
5	Sheila-kathryn	Calton	19 993	2017-04-28	11 638,9
6	Curr	Duckhouse	8 204	2017-02-06	23 971,6
7	Fina	Merali	18 549	2017-02-24	635,61
8	Rod	Inder	19 844	2017-01-28	2 091,47
9	Mala	Lind	2 979	2017-03-06	5 268,48
10	Fiorenze	Birdall	10 250	2017-07-13	15 926,76
11	Uriah	Bisatt	16 846	2017-06-02	752,64
12	Sawyer	Flattman	12 242	2017-07-23	742,66
13	Gabriele	Norcross	8 905	2017-02-16	5 262,45

Обновить Save Cancel Экспорт данных ... 200 3 444 строк

7. Макс интервал между двумя последовательными заказами

```
postgres 2 <postgres> Script <postgres 3> Scri... *<postgres 2> Sc... customers product o
with order_interval as (
    select o.customer_id, o.order_date,
    lag(o.order_date) over (partition by o.customer_id order by o.order_date) as previous
    from orders o
    where o.order_status = 'Approved'
),
intervals as (
    select oi.customer_id, case when oi.previous is not null
    then (oi.order_date - oi.previous) else null
    end as interval_days
    from order_interval oi
)
select c.first_name, c.last_name, c.job_title,
max(i.interval_days) as max_interval
from intervals i
inner join customers c on i.customer_id = c.customer_id
where i.interval_days is not null
group by c.customer_id, c.first_name, c.last_name, c.job_title
having count(*) > 1
```

	AZ first_name	AZ last_name	AZ job_title	123 max_interval
1	Mead	Wolland	Teacher	143
2	Nevile	Abraham	Computer Systems Analyst IV	77
3	Humfrey	Boyse		84
4	Meryl	Dumbar	Assistant Media Planner	232
5	Anatollo	Stendell		142
6	Gabie	Skett	Accountant III	154
7	Araldo	Klamp	Software Test Engineer IV	215
8	Skylar	Brise	Media Manager I	176

8. Топ-5 по сегментам (всего 3 сегмента, 1 клиент с null, итого - 16 значений)

```
postgres 2 <postgres> Script <postgres 3> Scri... *<postgres 2> Sc... customers product orders
with cust_profit as (
    select c.customer_id, c.first_name, c.last_name, c.wealth_segment,
    coalesce(sum(p.list_price * oi.quantity), 0) as profit
    from customers c
    left join orders o on c.customer_id = o.customer_id and o.order_status = 'Approved'
    left join order_items oi on o.order_id = oi.order_id
    left join product p on oi.product_id = p.product_id
    group by c.customer_id, c.first_name, c.last_name, c.wealth_segment
),
ranked as (
    select cp.*,
    row_number() over (partition by cp.wealth_segment order by cp.profit desc) as segment_rank
    from cust_profit cp
)
select r.first_name, r.last_name, r.wealth_segment, r.profit
from ranked r
where r.segment_rank <= 5
order by r.wealth_segment, r.profit desc
```

Таблица	AZ first_name	AZ last_name	AZ wealth_segment	123 profit
9	Hal	Braddon	High Net Worth	255 260,23
10	Terencio		High Net Worth	246 483,2499999999
11	Hercule		Mass Customer	220 194,51
12	Fonz	Jankin	Mass Customer	197 676,2
13	Bale	Ney	Mass Customer	175 565,02
14	Jillie	Fyndon	Mass Customer	174 753,99
15	Andras	Yurockin	Mass Customer	163 871,7
16	[NULL]	[NULL]	[NULL]	102 536,54