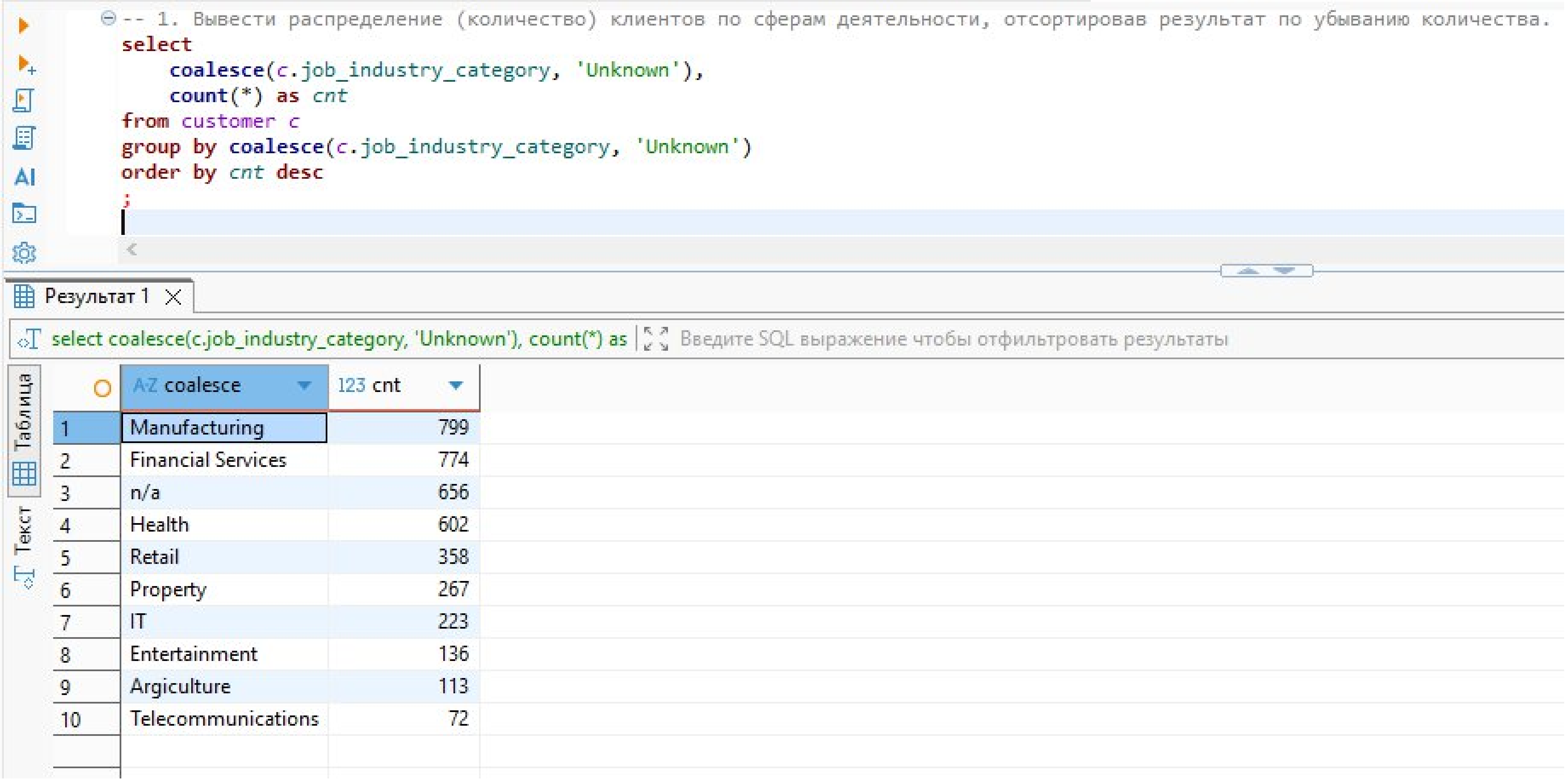
Задание 1

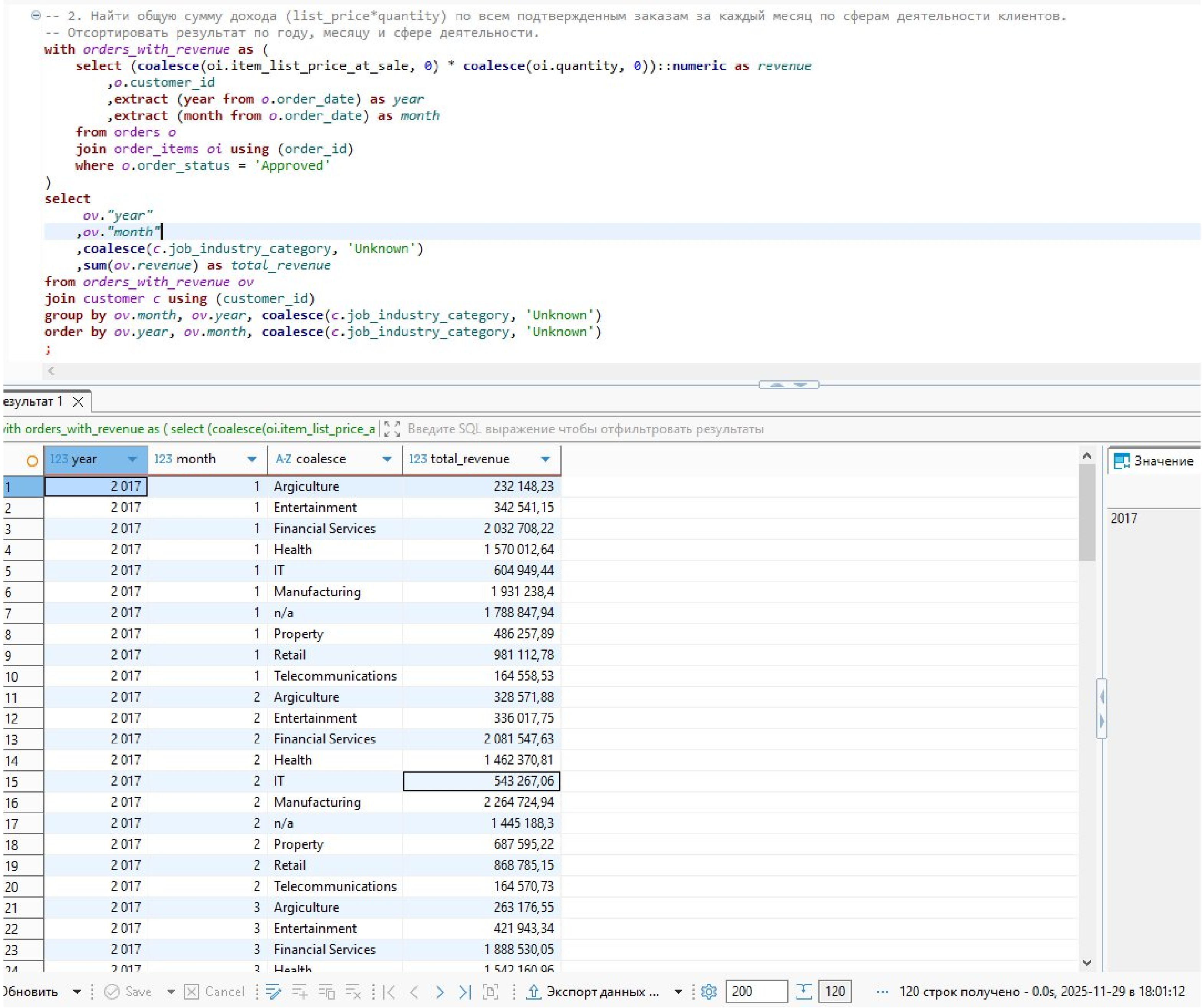


```
-- 1. Вывести распределение (количество) клиентов по сферам деятельности, отсортировав результат по убыванию количества.
select
    coalesce(c.job_industry_category, 'Unknown'),
    count(*) as cnt
from customer c
group by coalesce(c.job_industry_category, 'Unknown')
order by cnt desc
;
```

select coalesce(c.job_industry_category, 'Unknown'), count(*) as

	coalesce	cnt
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

Задание 2



```sql
-- 2. Найти общую сумму дохода (list_price*quantity) по всем подтвержденным заказам за каждый месяц по сферам деятельности клиентов.
-- Отсортировать результат по году, месяцу и сфере деятельности.
with orders_with_revenue as (
    select (coalesce(oi.item_list_price_at_sale, 0) * coalesce(oi.quantity, 0))::numeric as revenue
        ,o.customer_id
        ,extract (year from o.order_date) as year
        ,extract (month from o.order_date) as month
    from orders o
    join order_items oi using (order_id)
    where o.order_status = 'Approved'
)
select
     ov."year"
    ,ov."month"
    ,coalesce(c.job_industry_category, 'Unknown')
    ,sum(ov.revenue) as total_revenue
from orders_with_revenue ov
join customer c using (customer_id)
group by ov.month, ov.year, coalesce(c.job_industry_category, 'Unknown')
order by ov.year, ov.month, coalesce(c.job_industry_category, 'Unknown')
;
```

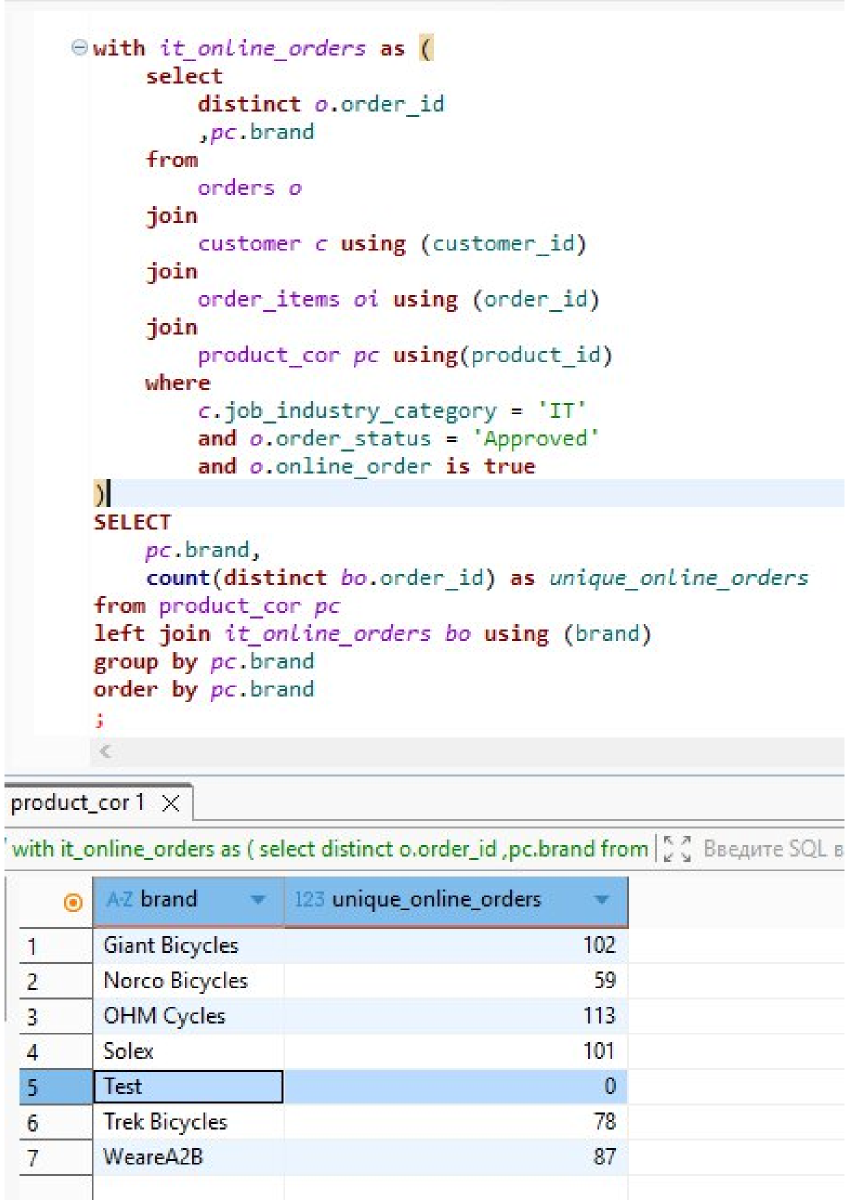
езультат 1
ith orders_with_revenue as (select (coalesce(oi.item_list_price_a
Введите SQL выражение чтобы отфильтровать результаты
Значение
2017

	123 year	123 month	A-Z coalesce	123 total_revenue
1	2 017	1	Argiculture	232 148,23
2	2 017	1	Entertainment	342 541,15
3	2 017	1	Financial Services	2 032 708,22
4	2 017	1	Health	1 570 012,64
5	2 017	1	IT	604 949,44
6	2 017	1	Manufacturing	1 931 238,4
7	2 017	1	n/a	1 788 847,94
8	2 017	1	Property	486 257,89
9	2 017	1	Retail	981 112,78
10	2 017	1	Telecommunications	164 558,53
11	2 017	2	Argiculture	328 571,88
12	2 017	2	Entertainment	336 017,75
13	2 017	2	Financial Services	2 081 547,63
14	2 017	2	Health	1 462 370,81
15	2 017	2	IT	543 267,06
16	2 017	2	Manufacturing	2 264 724,94
17	2 017	2	n/a	1 445 188,3
18	2 017	2	Property	687 595,22
19	2 017	2	Retail	868 785,15
20	2 017	2	Telecommunications	164 570,73
21	2 017	3	Argiculture	263 176,55
22	2 017	3	Entertainment	421 943,34
23	2 017	3	Financial Services	1 888 530,05
24	2 017	3	Health	1 542 160,96

120 строк получено - 0.0s, 2025-11-29 в 18:01:12

Задание 3

Для проверки скрипта был добавлен бренд, для которого нет заказов

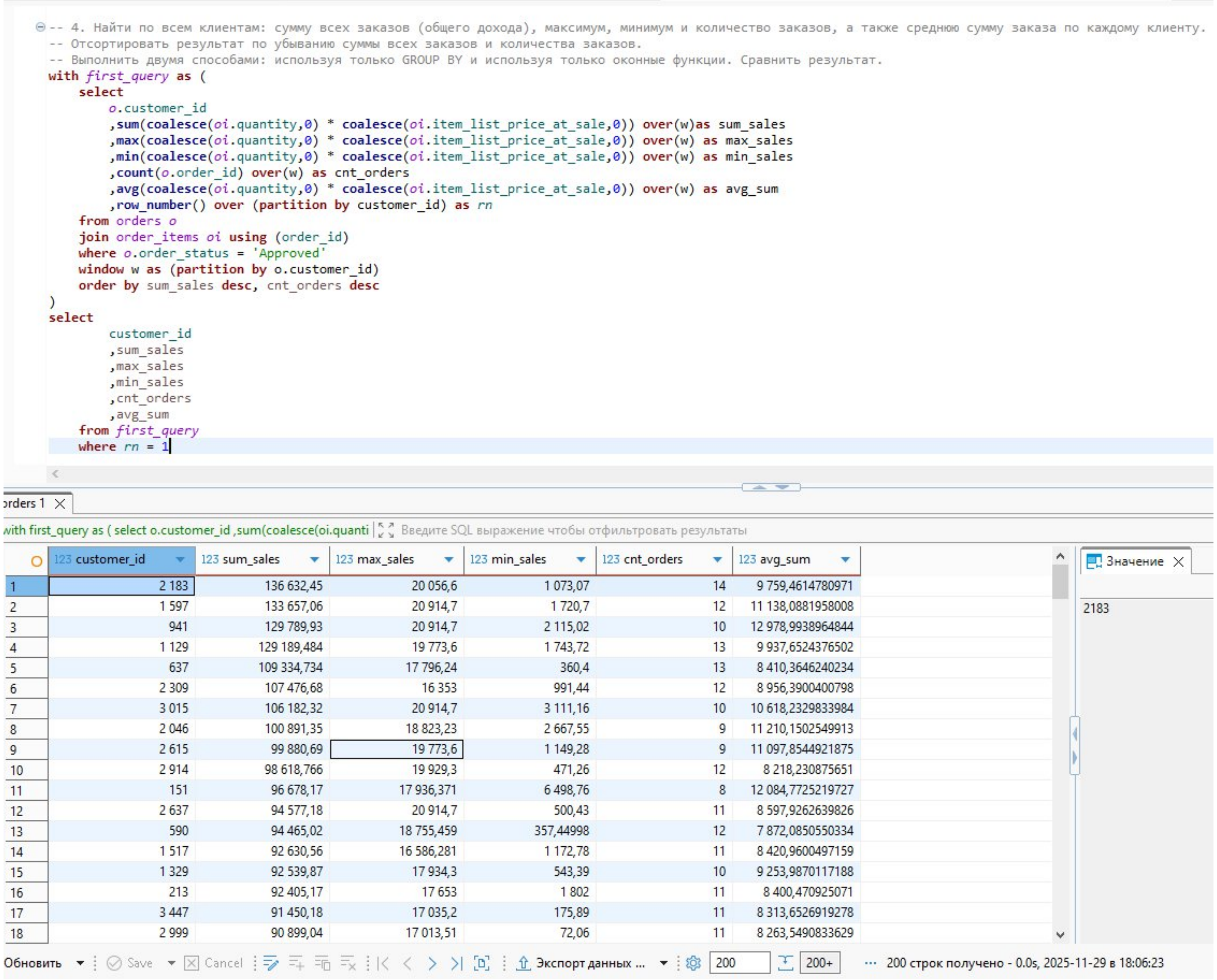


```sql
with it_online_orders as (
    select
        distinct o.order_id
        ,pc.brand
    from
        orders o
    join
        customer c using (customer_id)
    join
        order_items oi using (order_id)
    join
        product_cor pc using(product_id)
    where
        c.job_industry_category = 'IT'
        and o.order_status = 'Approved'
        and o.online_order is true
)
SELECT
    pc.brand,
    count(distinct bo.order_id) as unique_online_orders
from product_cor pc
left join it_online_orders bo using (brand)
group by pc.brand
order by pc.brand
;
```

product_cor 1
with it_online_orders as (select distinct o.order_id ,pc.brand from
Введите SQL в

	brand	unique_online_orders
1	Giant Bicycles	102
2	Norco Bicycles	59
3	OHM Cycles	113
4	Solex	101
5	Test	0
6	Trek Bicycles	78
7	WeareA2B	87

Задание 4



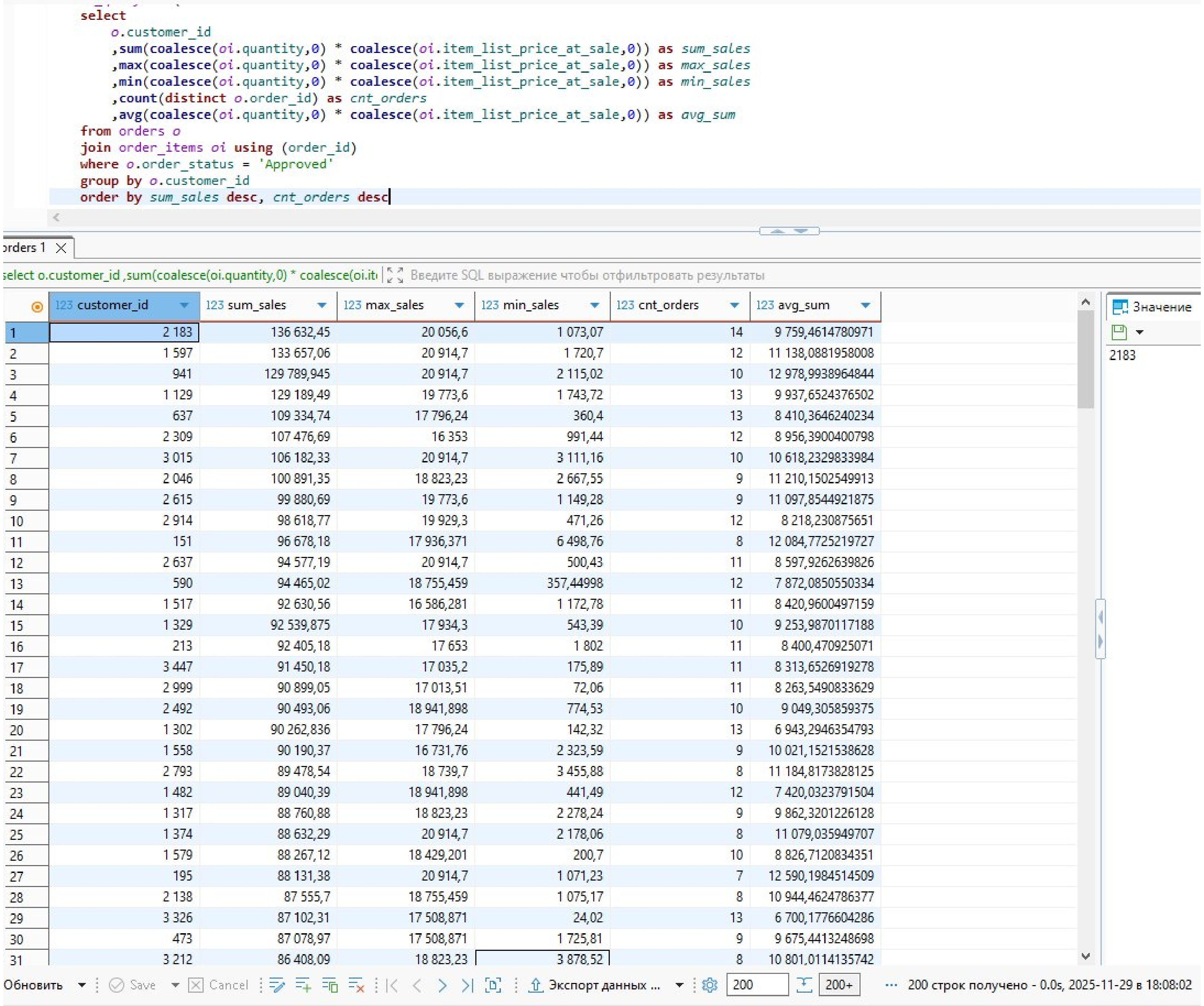
```sql
-- 4. Найти по всем клиентам: сумму всех заказов (общего дохода), максимум, минимум и количество заказов, а также среднюю сумму заказа по каждому клиенту.
-- Отсортировать результат по убыванию суммы всех заказов и количества заказов.
-- Выполнить двумя способами: используя только GROUP BY и используя только оконные функции. Сравнить результат.
with first_query as (
    select
        o.customer_id
        ,sum(coalesce(oi.quantity,0) * coalesce(oi.item_list_price_at_sale,0)) over(w)as sum_sales
        ,max(coalesce(oi.quantity,0) * coalesce(oi.item_list_price_at_sale,0)) over(w) as max_sales
        ,min(coalesce(oi.quantity,0) * coalesce(oi.item_list_price_at_sale,0)) over(w) as min_sales
        ,count(o.order_id) over(w) as cnt_orders
        ,avg(coalesce(oi.quantity,0) * coalesce(oi.item_list_price_at_sale,0)) over(w) as avg_sum
        ,row_number() over (partition by customer_id) as rn
    from orders o
    join order_items oi using (order_id)
    where o.order_status = 'Approved'
    window w as (partition by o.customer_id)
    order by sum_sales desc, cnt_orders desc
)
select
        customer_id
        ,sum_sales
        ,max_sales
        ,min_sales
        ,cnt_orders
        ,avg_sum
    from first_query
    where rn = 1
```

with first_query as (select o.customer_id ,sum(coalesce(oi.quanti

	customer_id	sum_sales	max_sales	min_sales	cnt_orders	avg_sum
1	2 183	136 632,45	20 056,6	1 073,07	14	9 759,4614780971
2	1 597	133 657,06	20 914,7	1 720,7	12	11 138,0881958008
3	941	129 789,93	20 914,7	2 115,02	10	12 978,9938964844
4	1 129	129 189,484	19 773,6	1 743,72	13	9 937,6524376502
5	637	109 334,734	17 796,24	360,4	13	8 410,3646240234
6	2 309	107 476,68	16 353	991,44	12	8 956,3900400798
7	3 015	106 182,32	20 914,7	3 111,16	10	10 618,2329833984
8	2 046	100 891,35	18 823,23	2 667,55	9	11 210,1502549913
9	2 615	99 880,69	19 773,6	1 149,28	9	11 097,8544921875
10	2 914	98 618,766	19 929,3	471,26	12	8 218,230875651
11	151	96 678,17	17 936,371	6 498,76	8	12 084,7725219727
12	2 637	94 577,18	20 914,7	500,43	11	8 597,9262639826
13	590	94 465,02	18 755,459	357,44998	12	7 872,0850550334
14	1 517	92 630,56	16 586,281	1 172,78	11	8 420,9600497159
15	1 329	92 539,87	17 934,3	543,39	10	9 253,9870117188
16	213	92 405,17	17 653	1 802	11	8 400,470925071
17	3 447	91 450,18	17 035,2	175,89	11	8 313,6526919278
18	2 999	90 899,04	17 013,51	72,06	11	8 263,5490833629

2183

200 строк получено - 0.0s, 2025-11-29 в 18:06:23



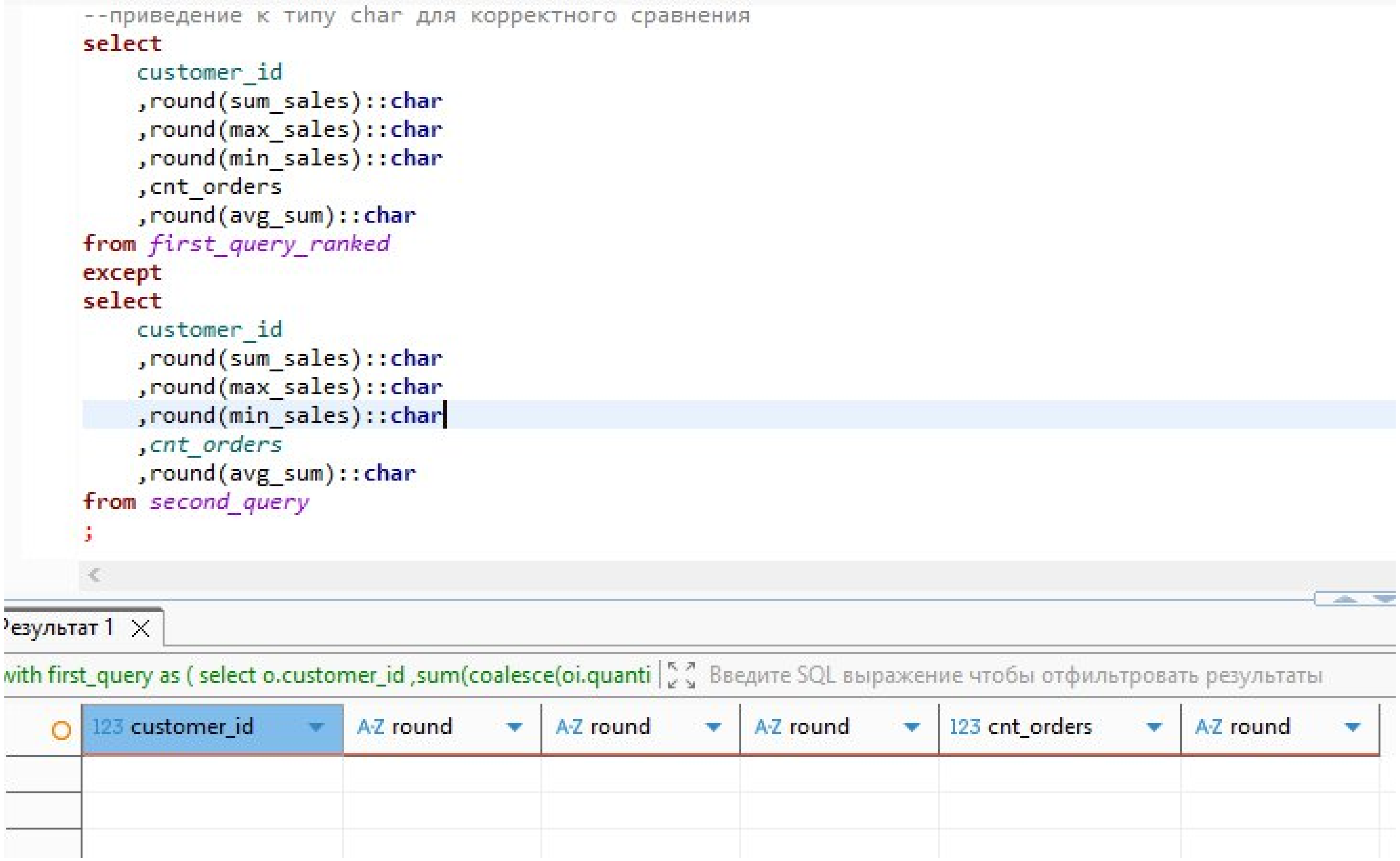
```sql
    select
        o.customer_id
        ,sum(coalesce(oi.quantity,0) * coalesce(oi.item_list_price_at_sale,0)) as sum_sales
        ,max(coalesce(oi.quantity,0) * coalesce(oi.item_list_price_at_sale,0)) as max_sales
        ,min(coalesce(oi.quantity,0) * coalesce(oi.item_list_price_at_sale,0)) as min_sales
        ,count(distinct o.order_id) as cnt_orders
        ,avg(coalesce(oi.quantity,0) * coalesce(oi.item_list_price_at_sale,0)) as avg_sum
    from orders o
    join order_items oi using (order_id)
    where o.order_status = 'Approved'
    group by o.customer_id
    order by sum_sales desc, cnt_orders desc
```

select o.customer_id ,sum(coalesce(oi.quantity,0) * coalesce(oi.it

	customer_id	sum_sales	max_sales	min_sales	cnt_orders	avg_sum
1	2 183	136 632,45	20 056,6	1 073,07	14	9 759,4614780971
2	1 597	133 657,06	20 914,7	1 720,7	12	11 138,0881958008
3	941	129 789,945	20 914,7	2 115,02	10	12 978,9938964844
4	1 129	129 189,49	19 773,6	1 743,72	13	9 937,6524376502
5	637	109 334,74	17 796,24	360,4	13	8 410,3646240234
6	2 309	107 476,69	16 353	991,44	12	8 956,3900400798
7	3 015	106 182,33	20 914,7	3 111,16	10	10 618,2329833984
8	2 046	100 891,35	18 823,23	2 667,55	9	11 210,1502549913
9	2 615	99 880,69	19 773,6	1 149,28	9	11 097,8544921875
10	2 914	98 618,77	19 929,3	471,26	12	8 218,230875651
11	151	96 678,18	17 936,371	6 498,76	8	12 084,7725219727
12	2 637	94 577,19	20 914,7	500,43	11	8 597,9262639826
13	590	94 465,02	18 755,459	357,44998	12	7 872,0850550334
14	1 517	92 630,56	16 586,281	1 172,78	11	8 420,960049159
15	1 329	92 539,875	17 934,3	543,39	10	9 253,9870117188
16	213	92 405,18	17 653	1 802	11	8 400,470925071
17	3 447	91 450,18	17 035,2	175,89	11	8 313,6526919278
18	2 999	90 899,05	17 013,51	72,06	11	8 263,5490833629
19	2 492	90 493,06	18 941,898	774,53	10	9 049,305859375
20	1 302	90 262,836	17 796,24	142,32	13	6 943,2946354793
21	1 558	90 190,37	16 731,76	2 323,59	9	10 021,1521538628
22	2 793	89 478,54	18 739,7	3 455,88	8	11 184,8173828125
23	1 482	89 040,39	18 941,898	441,49	12	7 420,0323791504
24	1 317	88 760,88	18 823,23	2 278,24	9	9 862,3201226128
25	1 374	88 632,29	20 914,7	2 178,06	8	11 079,035949707
26	1 579	88 267,12	18 429,201	200,7	10	8 826,7120834351
27	195	88 131,38	20 914,7	1 071,23	7	12 590,1984514509
28	2 138	87 555,7	18 755,459	1 075,17	8	10 944,4624786377
29	3 326	87 102,31	17 508,871	24,02	13	6 700,1776604286
30	473	87 078,97	17 508,871	1 725,81	9	9 675,4413248698
31	3 212	86 408,09	18 823,23	3 878,52	8	10 801,0114135742

200 строк получено - 0.0s, 2025-11-29 в 18:08:02

Сравнение:

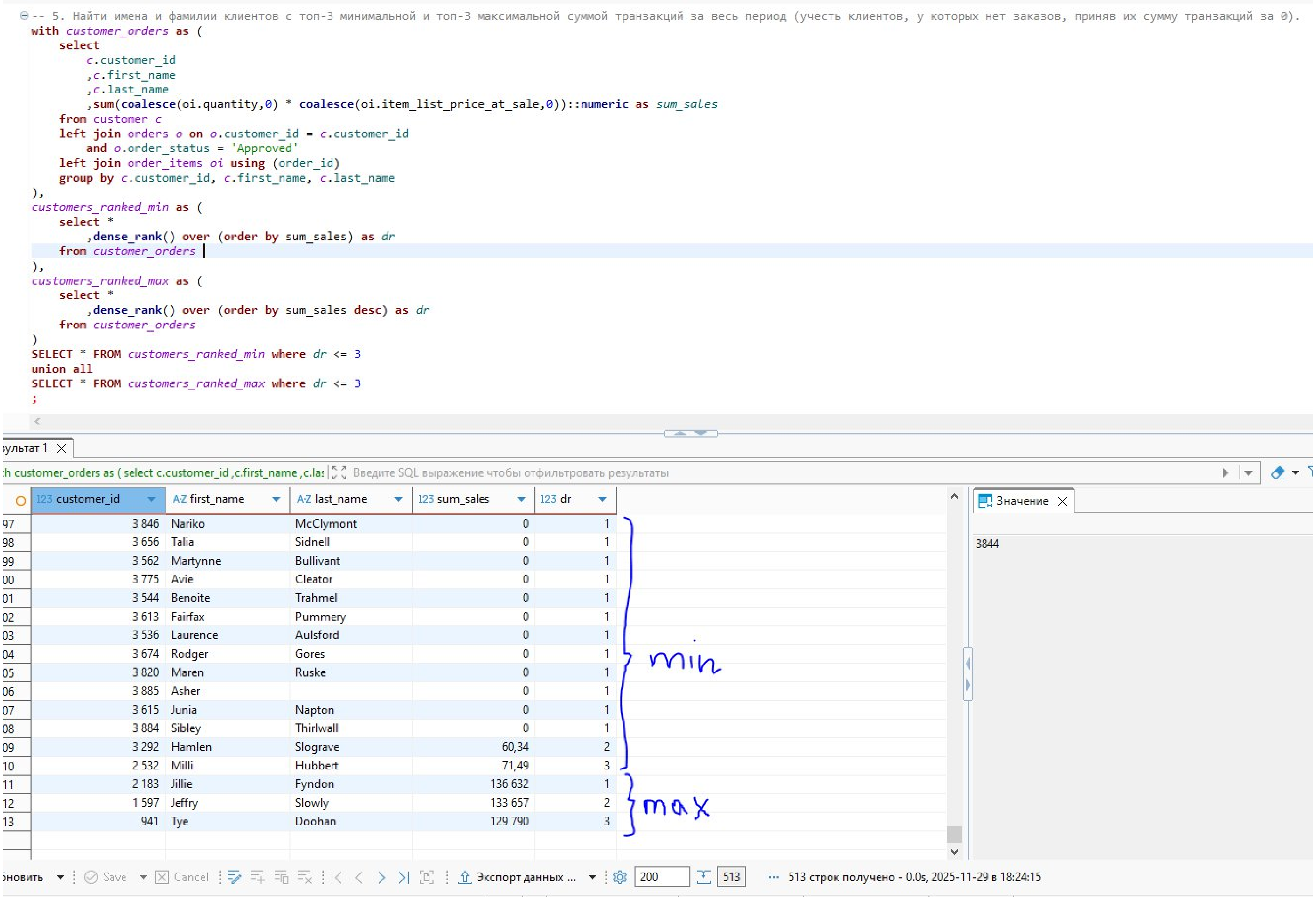


```
--приведение к типу char для корректного сравнения
select
    customer_id
    ,round(sum_sales)::char
    ,round(max_sales)::char
    ,round(min_sales)::char
    ,cnt_orders
    ,round(avg_sum)::char
from first_query_ranked
except
select
    customer_id
    ,round(sum_sales)::char
    ,round(max_sales)::char
    ,round(min_sales)::char
    ,cnt_orders
    ,round(avg_sum)::char
from second_query
;
```

езультат 1
with first_query as (select o.customer_id ,sum(coalesce(oi.quanti
Введите SQL выражение чтобы отфильтровать результаты

123 customer_id	A-Z round	A-Z round	A-Z round	123 cnt_orders	A-Z round

Задание 5



```
-- 5. Найти имена и фамилии клиентов с топ-3 минимальной и топ-3 максимальной суммой транзакций за весь период (учесть клиентов, у которых нет заказов, приняв их сумму транзакций за 0).
with customer_orders as (
    select
        c.customer_id
        ,c.first_name
        ,c.last_name
        ,sum(coalesce(oi.quantity,0) * coalesce(oi.item_list_price_at_sale,0))::numeric as sum_sales
    from customer c
    left join orders o on o.customer_id = c.customer_id
        and o.order_status = 'Approved'
    left join order_items oi using (order_id)
    group by c.customer_id, c.first_name, c.last_name
),
customers_ranked_min as (
    select *
        ,dense_rank() over (order by sum_sales) as dr
    from customer_orders
),
customers_ranked_max as (
    select *
        ,dense_rank() over (order by sum_sales desc) as dr
    from customer_orders
)
SELECT * FROM customers_ranked_min where dr <= 3
union all
SELECT * FROM customers_ranked_max where dr <= 3
;
```

h customer_orders as (select c.customer_id ,c.first_name ,c.la:

	123 customer_id	A-Z first_name	A-Z last_name	123 sum_sales	123 dr
97	3 846	Nariko	McClymont	0	1
98	3 656	Talia	Sidnell	0	1
99	3 562	Martynne	Bullivant	0	1
00	3 775	Avie	Cleator	0	1
01	3 544	Benoite	Trahmel	0	1
02	3 613	Fairfax	Pummery	0	1
03	3 536	Laurence	Aulsford	0	1
04	3 674	Rodger	Gores	0	1
05	3 820	Maren	Ruske	0	1
06	3 885	Asher		0	1
07	3 615	Junia	Napton	0	1
08	3 884	Sibley	Thirlwall	0	1
09	3 292	Hamlen	Slograve	60,34	2
10	2 532	Milli	Hubbert	71,49	3
11	2 183	Jillie	Fyndon	136 632	1
12	1 597	Jeffry	Slowly	133 657	2
13	941	Tye	Doohan	129 790	3

min

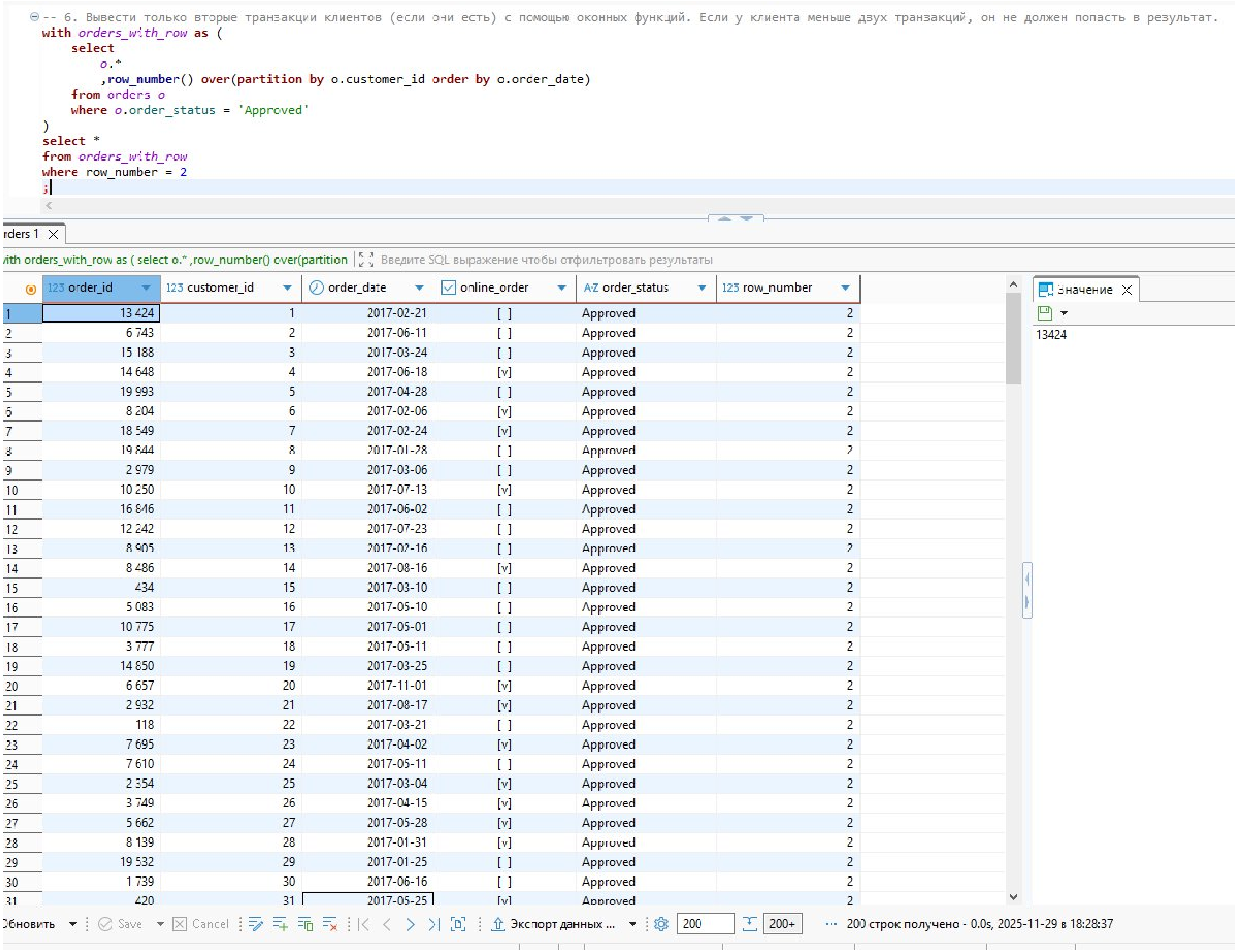
max

Значение

3844

513 строк получено - 0.0s, 2025-11-29 в 18:24:15

Задание 6



```sql
-- 6. Вывести только вторые транзакции клиентов (если они есть) с помощью оконных функций. Если у клиента меньше двух транзакций, он не должен попасть в результат.
with orders_with_row as (
    select
        o.*
        ,row_number() over(partition by o.customer_id order by o.order_date)
    from orders o
    where o.order_status = 'Approved'
)
select *
from orders_with_row
where row_number = 2
;
```

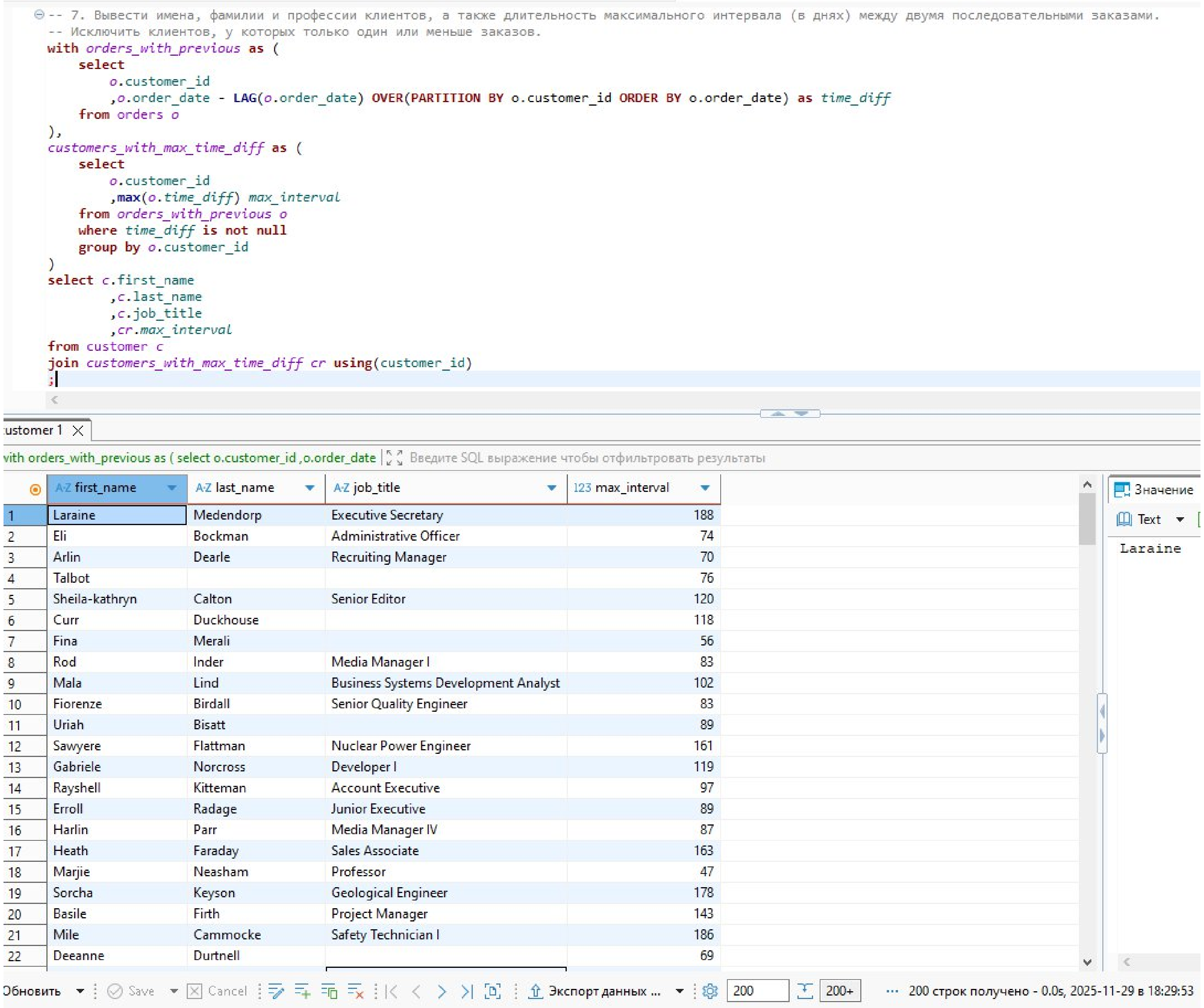
ith orders_with_row as (select o.* ,row_number() over(partition

	order_id	customer_id	order_date	online_order	order_status	row_number
1	13 424	1	2017-02-21	[ ]	Approved	2
2	6 743	2	2017-06-11	[ ]	Approved	2
3	15 188	3	2017-03-24	[ ]	Approved	2
4	14 648	4	2017-06-18	[v]	Approved	2
5	19 993	5	2017-04-28	[ ]	Approved	2
6	8 204	6	2017-02-06	[v]	Approved	2
7	18 549	7	2017-02-24	[v]	Approved	2
8	19 844	8	2017-01-28	[ ]	Approved	2
9	2 979	9	2017-03-06	[ ]	Approved	2
10	10 250	10	2017-07-13	[v]	Approved	2
11	16 846	11	2017-06-02	[ ]	Approved	2
12	12 242	12	2017-07-23	[ ]	Approved	2
13	8 905	13	2017-02-16	[ ]	Approved	2
14	8 486	14	2017-08-16	[v]	Approved	2
15	434	15	2017-03-10	[ ]	Approved	2
16	5 083	16	2017-05-10	[ ]	Approved	2
17	10 775	17	2017-05-01	[ ]	Approved	2
18	3 777	18	2017-05-11	[ ]	Approved	2
19	14 850	19	2017-03-25	[ ]	Approved	2
20	6 657	20	2017-11-01	[v]	Approved	2
21	2 932	21	2017-08-17	[v]	Approved	2
22	118	22	2017-03-21	[ ]	Approved	2
23	7 695	23	2017-04-02	[v]	Approved	2
24	7 610	24	2017-05-11	[ ]	Approved	2
25	2 354	25	2017-03-04	[v]	Approved	2
26	3 749	26	2017-04-15	[v]	Approved	2
27	5 662	27	2017-05-28	[v]	Approved	2
28	8 139	28	2017-01-31	[v]	Approved	2
29	19 532	29	2017-01-25	[ ]	Approved	2
30	1 739	30	2017-06-16	[ ]	Approved	2
31	420	31	2017-05-25	[v]	Approved	2

13424

200 строк получено - 0.0s, 2025-11-29 в 18:28:37

Задание 7



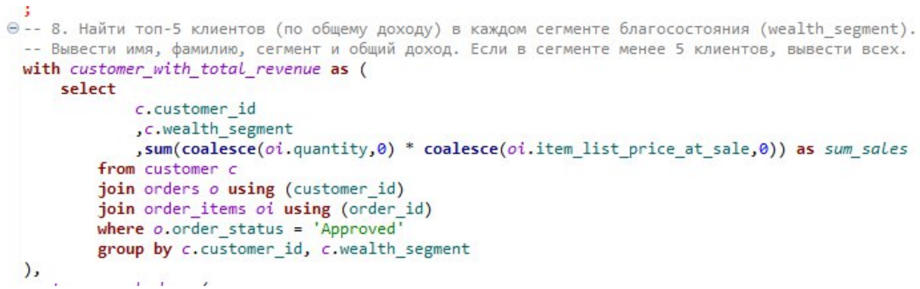
```
-- 7. Вывести имена, фамилии и профессии клиентов, а также длительность максимального интервала (в днях) между двумя последовательными заказами.
-- Исключить клиентов, у которых только один или меньше заказов.
with orders_with_previous as (
    select
        o.customer_id
        ,o.order_date - LAG(o.order_date) OVER(PARTITION BY o.customer_id ORDER BY o.order_date) as time_diff
    from orders o
),
customers_with_max_time_diff as (
    select
        o.customer_id
        ,max(o.time_diff) max_interval
    from orders_with_previous o
    where time_diff is not null
    group by o.customer_id
)
select c.first_name
        ,c.last_name
        ,c.job_title
        ,cr.max_interval
from customer c
join customers_with_max_time_diff cr using(customer_id)
;
```

vith orders_with_previous as (select o.customer_id ,o.order_date

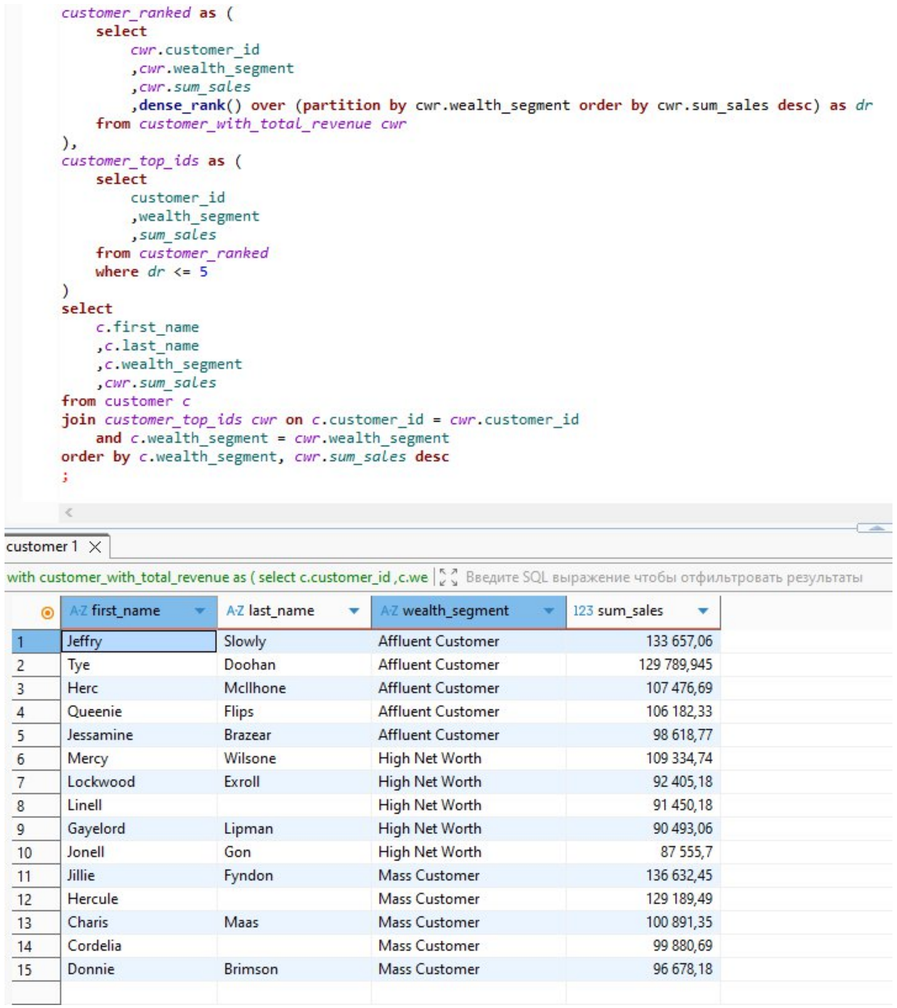
	first_name	last_name	job_title	max_interval
1	Laraine	Medendorp	Executive Secretary	188
2	Eli	Bockman	Administrative Officer	74
3	Arlin	Dearle	Recruiting Manager	70
4	Talbot			76
5	Sheila-kathryn	Calton	Senior Editor	120
6	Curr	Duckhouse		118
7	Fina	Merali		56
8	Rod	Inder	Media Manager I	83
9	Mala	Lind	Business Systems Development Analyst	102
10	Fiorenze	Birdall	Senior Quality Engineer	83
11	Uriah	Bisatt		89
12	Sawyere	Flattman	Nuclear Power Engineer	161
13	Gabriele	Norcross	Developer I	119
14	Rayshell	Kitteman	Account Executive	97
15	Erroll	Radage	Junior Executive	89
16	Harlin	Parr	Media Manager IV	87
17	Heath	Faraday	Sales Associate	163
18	Marjie	Neasham	Professor	47
19	Sorcha	Keyson	Geological Engineer	178
20	Basile	Firth	Project Manager	143
21	Mile	Cammocke	Safety Technician I	186
22	Deeanne	Durtnell		69

200 строк получено - 0.0s, 2025-11-29 в 18:29:53

Задание 8



```sql
;
-- 8. Найти топ-5 клиентов (по общему доходу) в каждом сегменте благосостояния (wealth_segment).
-- Вывести имя, фамилию, сегмент и общий доход. Если в сегменте менее 5 клиентов, вывести всех.
with customer_with_total_revenue as (
    select
            c.customer_id
            ,c.wealth_segment
            ,sum(coalesce(oi.quantity,0) * coalesce(oi.item_list_price_at_sale,0)) as sum_sales
        from customer c
        join orders o using (customer_id)
        join order_items oi using (order_id)
        where o.order_status = 'Approved'
        group by c.customer_id, c.wealth_segment
),
```



```sql
customer_ranked as (
    select
        cwr.customer_id
        ,cwr.wealth_segment
        ,cwr.sum_sales
        ,dense_rank() over (partition by cwr.wealth_segment order by cwr.sum_sales desc) as dr
    from customer_with_total_revenue cwr
),
customer_top_ids as (
    select
        customer_id
        ,wealth_segment
        ,sum_sales
    from customer_ranked
    where dr <= 5
)
select
    c.first_name
    ,c.last_name
    ,c.wealth_segment
    ,cwr.sum_sales
from customer c
join customer_top_ids cwr on c.customer_id = cwr.customer_id
    and c.wealth_segment = cwr.wealth_segment
order by c.wealth_segment, cwr.sum_sales desc
;
```

customer 1
with customer_with_total_revenue as (select c.customer_id ,c.we
Введите SQL выражение чтобы отфильтровать результаты

	first_name	last_name	wealth_segment	sum_sales
1	Jeffry	Slowly	Affluent Customer	133 657,06
2	Tye	Doohan	Affluent Customer	129 789,945
3	Herc	McIlhone	Affluent Customer	107 476,69
4	Queenie	Flips	Affluent Customer	106 182,33
5	Jessamine	Brazear	Affluent Customer	98 618,77
6	Mercy	Wilsone	High Net Worth	109 334,74
7	Lockwood	Exroll	High Net Worth	92 405,18
8	Linell		High Net Worth	91 450,18
9	Gayelord	Lipman	High Net Worth	90 493,06
10	Jonell	Gon	High Net Worth	87 555,7
11	Jillie	Fyndon	Mass Customer	136 632,45
12	Hercule		Mass Customer	129 189,49
13	Charis	Maas	Mass Customer	100 891,35
14	Cordelia		Mass Customer	99 880,69
15	Donnie	Brimson	Mass Customer	96 678,18