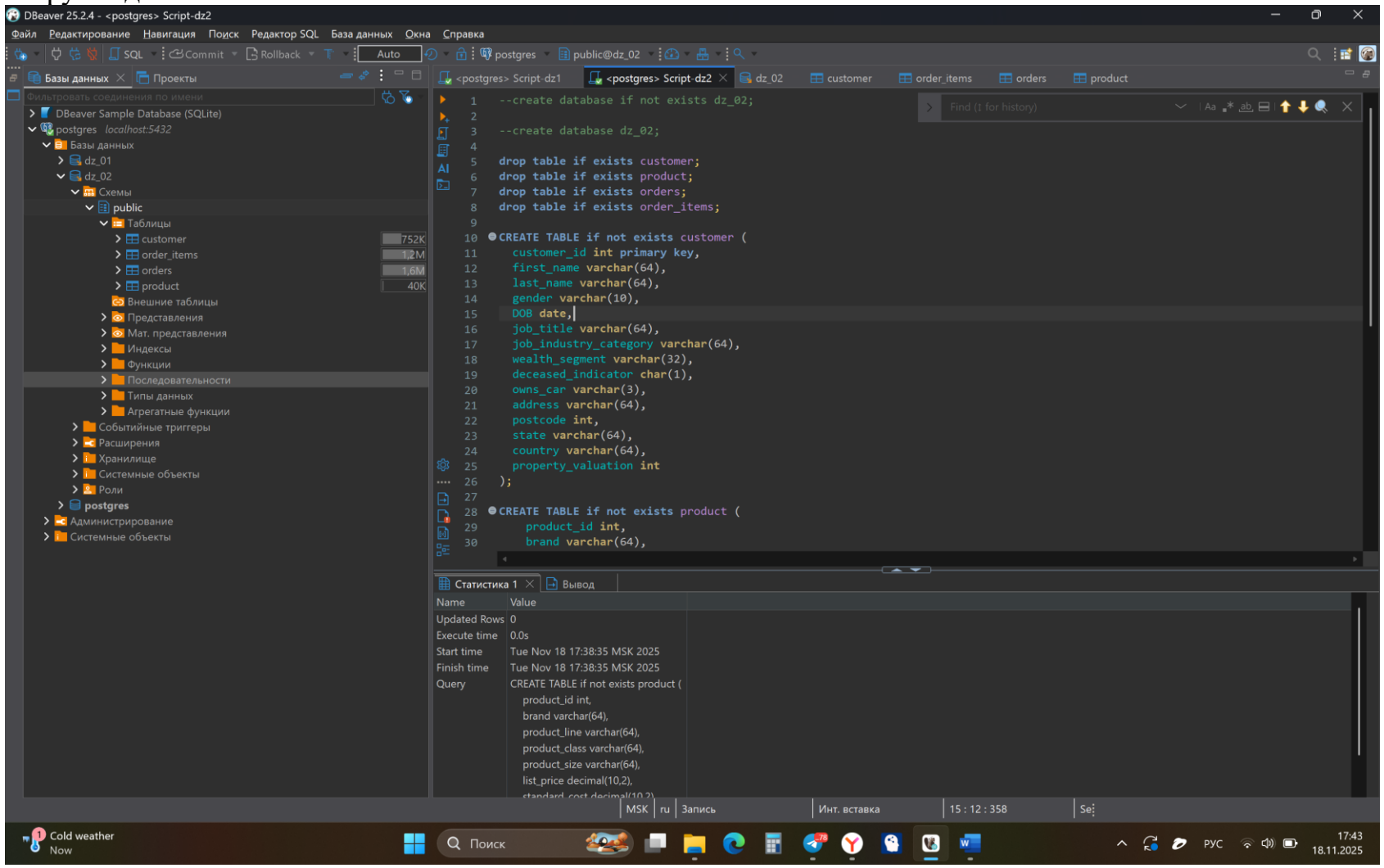
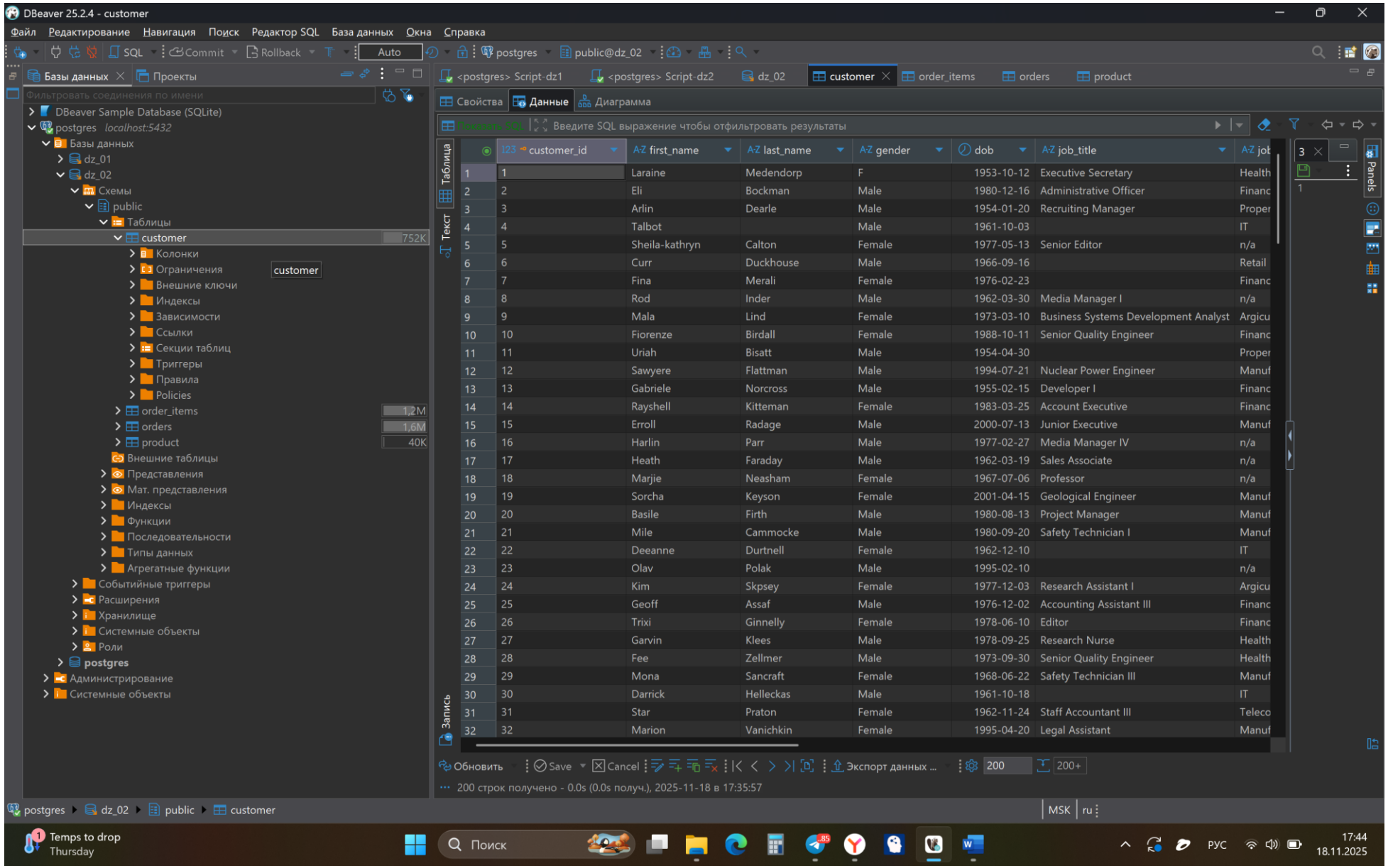
Никулин К.А. ДЗ №3

Загрузка данных



Никулин К.А. ДЗ №3



1. Вывести распределение (количество) клиентов по сферам деятельности, отсортировав результат по убыванию количества.

```sql
-- Домашнее задание №3.
-- №1.
--Вывести распределение (количество) клиентов по сферам деятельности,
--отсортировав результат по убыванию количества.
select
    job_industry_category,
    count(*) as customers_count
from
    customer
group by
    job_industry_category
order by
    customers_count desc;
```

customer 1

Введите SQL выражение чтобы отфильтровать результаты

	job_industry_category	customers_count
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

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.

```sql
select
    extract (year from o.order_date) as year,
    extract (month from o.order_date) as month,
    c.job_industry_category,
    sum(p_c.list_price*o_i.quantity) as total_revenue
from customer c
join orders o on c.customer_id = o.customer_id
join order_items o_i on o.order_id = o_i.order_id
join product_cor p_c on o_i.product_id = p_c.product_id
where o.order_status = 'Approved'
group by year, month, c.job_industry_category
order by year, month, c.job_industry_category;
```

ustomer 1

Введите SQL выражение чтобы отфильтровать результаты

	year	month	job_industry_category	total_revenue
1	2 017	1	Argiculture	299 741,43
2	2 017	1	Entertainment	416 965,33
3	2 017	1	Financial Services	2 443 496,22
4	2 017	1	Health	1 921 608,44
5	2 017	1	IT	735 074,61
6	2 017	1	Manufacturing	2 413 462,43
7	2 017	1	n/a	2 278 578
8	2 017	1	Property	575 902,88
9	2 017	1	Retail	1 217 372,32
10	2 017	1	Telecommunications	211 919,63
11	2 017	2	Argiculture	420 197,65
12	2 017	2	Entertainment	399 603,17
13	2 017	2	Financial Services	2 541 873,95
14	2 017	2	Health	1 811 851,88
15	2 017	2	IT	699 719,96
16	2 017	2	Manufacturing	2 807 982,8
17	2 017	2	n/a	1 781 138,61
18	2 017	2	Property	870 581,34
19	2 017	2	Retail	1 098 580,25
20	2 017	2	Telecommunications	237 096,43
21	2 017	3	Argiculture	320 236,94
22	2 017	3	Entertainment	488 610,62
23	2 017	3	Financial Services	2 305 750,75
24	2 017	3	Health	1 909 695,04
25	2 017	3	IT	637 157,44
26	2 017	3	Manufacturing	2 680 021,33
27	2 017	3	n/a	2 004 002,22
28	2 017	3	Property	864 363,39
29	2 017	3	Retail	1 163 784,73
30	2 017	3	Telecommunications	193 538,84
31	2 017	4	Argiculture	334 741,6
32	2 017	4	Entertainment	445 760,41

Обновить | Save | Cancel | Экспорт данных ... | 200

120 | 120 строк получено - 0.0s, 2025-11-30 в 15:16:10

3. Вывести количество уникальных онлайн-заказов для всех брендов в рамках подтвержденных заказов клиентов из сферы IT. Включить бренды, у которых нет онлайн-заказов от IT-клиентов, — для них должно быть указано количество 0.

```sql
-- №3.
--Вывести количество уникальных онлайн-заказов для всех брендов
--в рамках подтвержденных заказов клиентов из сферы IT.
--Включить бренды, у которых нет онлайн-заказов от IT-клиентов,
--— для них должно быть указано количество 0.

select
    p_c.brand,
    count(distinct o.order_id) as online_order_brand
from product_cor p_c
left join order_items o_i on p_c.product_id = o_i.product_id
left join orders o on o_i.order_id = o.order_id
left join customer c
    on o.customer_id = c.customer_id
    and c.job_industry_category = 'IT'
    and o.order_status = 'Approved'
    and o.online_order = 'True'
group by p_c.brand
order by online_order_brand desc, p_c.brand;
```

oduct_cor 1

Введите SQL выражение чтобы отфильтровать результаты

brand	online_order_brand
OHM Cycles	3 899
Solex	3 885
Giant Bicycles	3 755
WeareA2B	3 685
Trek Bicycles	2 568
Norco Bicycles	2 208

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.

```sql
select
    c.first_name,
    c.last_name,
    sum(o_i.item_list_price_at_sale) as total_revenue,
    max(o_i.item_list_price_at_sale) as max_order_amount,
    min(o_i.item_list_price_at_sale) as min_order_amount,
    count(distinct o.order_id) as orders_count,
    avg(o_i.item_list_price_at_sale) as avg_order_amount
from
    customer c
    join orders o  on c.customer_id = o.customer_id
    join order_items o_i on o.order_id = o_i.order_id
group by
    c.customer_id,
    c.first_name,
    c.last_name
order by
    total_revenue desc,
    orders_count desc;
```

customer 1

Введите SQL выражение чтобы отфильтровать результаты

	first_name	last_name	total_revenue	max_order_amount	min_order_amount	orders_count	avg_order_amount
1	Jillie	Fyndon	19 071,32	2 005,66	230,91	14	1 362,23714285
2	Hercule		18 349,27	1 992,93	290,62	13	1 411,48230769
3	Jeffry	Slowly	18 052,68	2 091,47	360,4	12	1 504,
4	Tye	Doohan	17 898,46	2 091,47	1 057,51	10	1 789,8
5	Melantha	Pickburn	17 258,94	2 083,94	183,86	11	1 568,99454545
6	Raffaello	Godleman	17 160,24	2 005,66	183,86	12	1 430,
7	Kynthia	Purcer	17 133,93	2 091,47	688,63	11	1 557,
8	Ericka	Eggers	17 035,83	1 977,36	71,16	13	1 310,44846153
9	Deana	Rathbourne	16 199,24	2 083,94	183,86	13	1 246,09538461
10	Herc	McIlhone	16 122,34	2 091,47	290,62	12	1 343,52833333
11	Ammamaria	Standbridge	15 826	2 091,47	586,45	10	1 583

```sql
with order_level as (
    select
        c.customer_id,
        c.first_name,
        c.last_name,
        o.order_id,
        sum(o_i.item_list_price_at_sale) as order_amount
    from customer c
    join orders o on c.customer_id = o.customer_id
    join order_items o_i on o.order_id = o_i.order_id
    group by
        c.customer_id, c.first_name, c.last_name , o.order_id
),
customer_agr as (
    select
        o_l.customer_id,
        o_l.first_name,
        o_l.last_name,
        o_l.order_id,
        o_l.order_amount,
        sum(o_l.order_amount) over (partition by o_l.customer_id) as total_revenue,
        max(o_l.order_amount) over (partition by o_l.customer_id) as max_order_amount,
        min(o_l.order_amount) over (partition by o_l.customer_id) as min_order_amount,
        count(o_l.order_amount) over (partition by o_l.customer_id) as orders_count,
        avg(o_l.order_amount) over (partition by o_l.customer_id) as avg_order_amount
    from order_level o_l
)
select distinct
    c_a.customer_id,
    c_a.first_name,
    c_a.last_name,
    c_a.total_revenue,
    c_a.max_order_amount,
    c_a.min_order_amount,
    c_a.orders_count,
    c_a.avg_order_amount
from customer_agr c_a
order by
    total_revenue desc,
    orders_count desc;
```

Введите SQL выражение чтобы отфильтровать результаты

	first_name	last_name	total_revenue	max_order_amount	min_order_amount	orders_count	avg_order_amount
1	Jillie	Fyndon	19 071,32	2 005,66	230,91	14	1 362,23714285
2	Hercule		18 349,27	1 992,93	290,62	13	1 411,48230769
3	Jeffry	Slowly	18 052,68	2 091,47	360,4	12	1 504,
4	Tye	Doohan	17 898,46	2 091,47	1 057,51	10	1 789,8
5	Melantha	Pickburn	17 258,94	2 083,94	183,86	11	1 568,99454545
6	Raffaello	Godleman	17 160,24	2 005,66	183,86	12	1 430,
7	Kynthia	Purcer	17 133,93	2 091,47	688,63	11	1 557,
8	Ericka	Eggers	17 035,83	1 977,36	71,16	13	1 310,44846153
9	Deana	Rathbourne	16 199,24	2 083,94	183,86	13	1 246,09538461
10	Herc	McIlhone	16 122,34	2 091,47	290,62	12	1 343,52833333
11	Ammamaria	Standbridge	15 826	2 091,47	586,45	10	1 58
12	Glynnis	Sailor	15 447,92	1 977,36	230,91	12	1 287,32666666
13	Barrett	Lindley	15 370,81	2 091,47	569,56	9	1 707,86777777
14	Delores	Ashcroft	15 091,91	2 005,66	544,05	11	1 371,99181818
15	Rozamond	Franceschino	15 071,26	1 997,68	183,86	10	1 507,1
16	Keeley	Kruger	14 949,91	1 810	441,49	12	1 245,82583333

Результирующие таблицы идентичны

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).

```sql
-- №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_sum as (
    select
        c.customer_id,
        c.first_name,
        c.last_name,
        sum(oi.item_list_price_at_sale) as total_revenue
    from customer c
    left join orders o      on c.customer_id = o.customer_id
    left join order_items oi on o.order_id   = oi.order_id
    group by
        c.customer_id, c.first_name, c.last_name
),
ranked as (
    select
        customer_id,
        first_name,
        last_name,
        total_revenue,
        rank() over (order by total_revenue asc nulls last)  as r_min,
        rank() over (order by total_revenue desc nulls last) as r_max
    from customer_sum
)
select
    customer_id,
    first_name,
    last_name,
    total_revenue
from ranked
where r_min <= 3
   or r_max <= 3
order by total_revenue, customer_id;
```

customer 1

WITH customer_sum AS (SELECT c.customer_id, Введите SQL выражение чтобы отфильтровать результаты

Таблица | Текст

	customer_id	first_name	last_name	total_revenue
1	3 292	Hamlen	Slograve	60,34
2	2 532	Milli	Hubbert	71,49
3	2 274	Nada	Reinert	142,98
4	1 597	Jeffry	Slowly	18 052,68
5	1 129	Hercule		18 349,27
6	2 183	Jillie	Fyndon	19 071,32

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.

```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 ordered_orders as (
    select
        c.customer_id,
        c.first_name,
        c.last_name,
        o.order_id,
        o.order_date,
        row_number() over (
            partition by c.customer_id
            order by o.order_date
        ) as row_num
    from customer c
    join orders o on c.customer_id = o.customer_id
)
select
    customer_id,
    first_name,
    last_name,
    order_id,
    order_date
from ordered_orders
where row_num = 2;
```

	customer_id	first_name	last_name	order_id	order_date
1	1	Laraine	Medendorp	13 424	2017-02-21
2	2	Eli	Bockman	6 743	2017-06-11
3	3	Arlin	Dearle	15 188	2017-03-24
4	4	Talbot		14 648	2017-06-18
5	5	Sheila-kathryn	Calton	19 993	2017-04-28
6	6	Curr	Duckhouse	8 204	2017-02-06
7	7	Fina	Merali	18 549	2017-02-24
8	8	Rod	Inder	19 844	2017-01-28
9	9	Mala	Lind	2 979	2017-03-06
10	10	Fiorenze	Birdall	10 250	2017-07-13
11	11	Uriah	Bisatt	16 846	2017-06-02
12	12	Sawyere	Flattman	12 242	2017-07-23
13	13	Gabriele	Norcross	8 905	2017-02-16
14	14	Rayshell	Kitteman	8 486	2017-08-16
15	15	Erroll	Radage	434	2017-03-10
16	16	Harlin	Parr	5 083	2017-05-10
17	17	Heath	Faraday	10 775	2017-05-01
18	18	Marjie	Neasham	3 777	2017-05-11
19	19	Sorcha	Keyson	14 850	2017-03-25

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.

```sql
-- №7.
-- Вывести имена, фамилии и профессии клиентов,
--а также длительность максимального интервала (в днях) между двумя последовательными заказа
--Исключить клиентов, у которых только один или меньше заказов.
with ordered_orders as (
    select
        c.customer_id,
        c.first_name,
        c.last_name,
        c.job_title,
        o.order_date,
        lag(o.order_date) over (
            partition by c.customer_id
            order by o.order_date
        ) as prev_order_date
    from customer c
    join orders o on c.customer_id = o.customer_id
),
intervals as (
    select
        customer_id,
        first_name,
        last_name,
        job_title,
        (order_date - prev_order_date) as diff_days
    from ordered_orders
    where prev_order_date is not null
)
select
    customer_id,
    first_name,
    last_name,
    job_title,
    max(diff_days) as max_interval_days
from intervals
group by
    customer_id,
    first_name,
    last_name,
    job_title
order by
    max_interval_days desc,
    customer_id;
```

Введите SQL выражение чтобы отфильтровать результаты

	customer_id	first_name	last_name	job_title	max_interval_days
1	1 584	Susanetta		Legal Assistant	357
2	1 810	Royall	Terris	Geological Engineer	330
3	2 128	Gregorius	Cockram	Data Coordiator	330
4	3 316	Stoddard	Giacomoni	Structural Analysis Engineer	330
5	3 156	Bearnard	Letixier		329
6	3 222	Caralie	Sellors	Senior Editor	321
7	335	Debee	Martynov	Senior Editor	320
8	316	Genni	Larway	Environmental Specialist	314
9	2 085	Carolynn	Samsin	Pharmacist	310
10	2 146	Timmie	Lenden		310
11	3 024	Franz	Craddy		310
12	92	Jodee	Judkins	Recruiting Manager	306
13	1 633	Ashia	Muzzi	Mechanical Systems Enginee	306
14	2 586	Heywood	Sollett	Tax Accountant	305
15	2 541	Cleveland	Islep	Software Engineer II	299
16	510	Sheilah	Blackmore		297
17	520	Jazmin	Neumann	VP Quality Control	297
18	2 108	Michel	O'Halligan	Software Test Engineer IV	297
19	3 159	Jesus	MacShirie		296
20	2 172	Emmery	Angrock	Information Systems Manag	292
21	2 329	Donny	Stiven	Computer Systems Analyst II	292
22	2 122	Osborne	Nawton	Registered Nurse	291
23	2 500	Micky	Livings	Senior Developer	291
24	2 106	Antons	Ley	Project Manager	290
25	2 218	Madelena	Blincoe	Senior Sales Associate	286

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.

```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_revenue as (
    select
        c.customer_id,
        c.first_name,
        c.last_name,
        c.wealth_segment,
        sum(o_i.item_list_price_at_sale) as total_revenue
    from customer c
    join orders o on c.customer_id = o.customer_id
    join order_items o_i on o.order_id = o_i.order_id
    group by
        c.customer_id,
        c.first_name,
        c.last_name,
        c.wealth_segment
),
ranked as (
    select
        customer_id,
        first_name,
        last_name,
        wealth_segment,
        total_revenue,
        rank() over (
            partition by wealth_segment
            order by total_revenue desc
        ) as rnk
    from customer_revenue
)
select
    customer_id,
    first_name,
    last_name,
    wealth_segment,
    total_revenue
from ranked
where rnk <= 5
order by
    wealth_segment,
    rnk,
    customer_id;
```

customer 1

with customer_revenue as (select c.custom | Введите SQL выражение чтобы отфильтровать результаты

Таблица | Текст

	customer_id	first_name	last_name	wealth_segment	total_revenue
1	1 597	Jeffry	Slowly	Affluent Customer	18 052,68
2	941	Tye	Doohan	Affluent Customer	17 898,46
3	2 788	Melantha	Pickburn	Affluent Customer	17 258,94
4	1 887	Kynthia	Purcer	Affluent Customer	17 133,93
5	1 302	Ericka	Eggers	Affluent Customer	17 035,83
6	936	Raffaello	Godleman	High Net Worth	17 160,24
7	1 103	Glynnis	Sailor	High Net Worth	15 447,92
8	999	Dido	Leyburn	High Net Worth	14 662,14
9	1 460	Morley	Shutt	High Net Worth	14 633,24
10	2 476	Hal	Braddon	High Net Worth	14 578,69
11	2 183	Jillie	Fyndon	Mass Customer	19 071,32
12	1 129	Hercule		Mass Customer	18 349,27
13	1 140	Deana	Rathbourne	Mass Customer	16 199,24
14	1 317	Barrett	Lindley	Mass Customer	15 370,81
15	2 762	Rozamond	Franceschino	Mass Customer	15 071,26