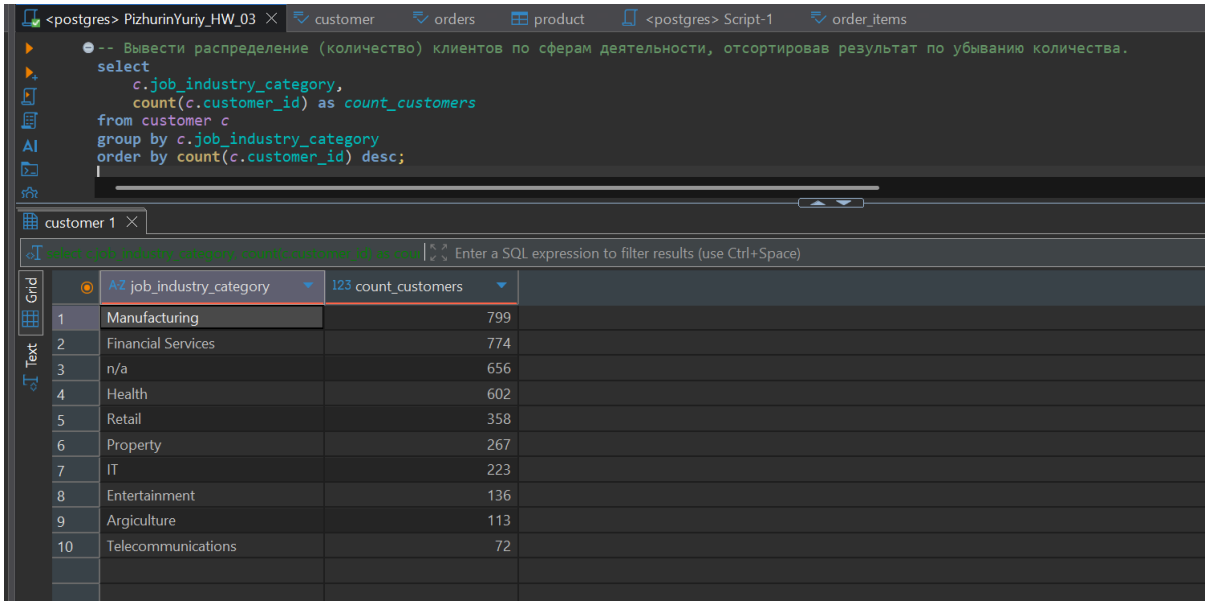
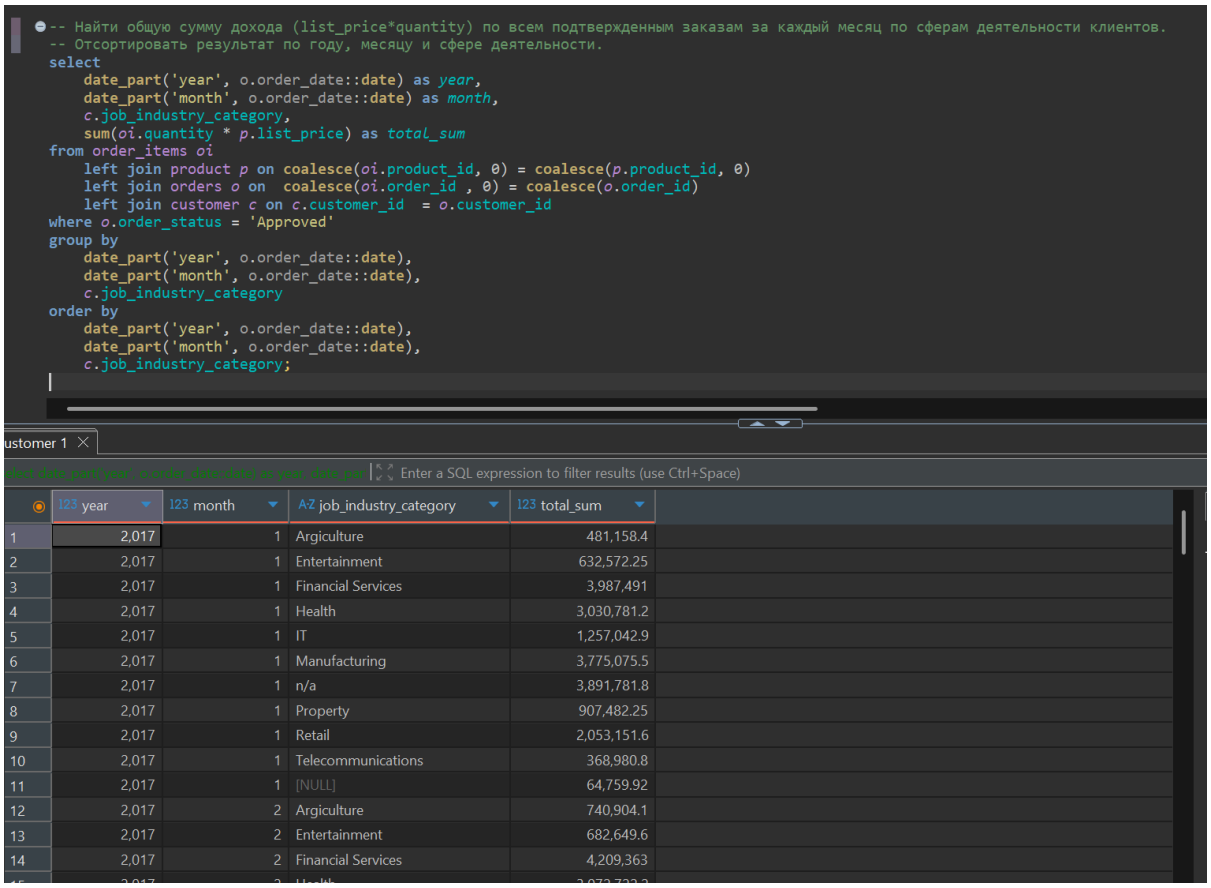

<postgres> PizhurinYuriy_HW_03
customer
orders
product
<postgres> Script-1
order_items
-- Вывести распределение (количество) клиентов по сферам деятельности, отсортировав результат по убыванию количества.
select
c.job_industry_category,
count(c.customer_id) as count_customers
from customer c
group by c.job_industry_category
order by count(c.customer_id) desc;
customer 1
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
job_industry_category
count_customers
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


-- 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.
select
date_part('year', o.order_date::date) as year,
date_part('month', o.order_date::date) as month,
c.job_industry_category,
sum(oi.quantity * p.list_price) as total_sum
from order_items oi
left join product p on coalesce(oi.product_id, 0) = coalesce(p.product_id, 0)
left join orders o on coalesce(oi.order_id , 0) = coalesce(o.order_id)
left join customer c on c.customer_id = o.customer_id
where o.order_status = 'Approved'
group by
date_part('year', o.order_date::date),
date_part('month', o.order_date::date),
c.job_industry_category
order by
date_part('year', o.order_date::date),
date_part('month', o.order_date::date),
c.job_industry_category;
ustomer 1
Enter a SQL expression to filter results (use Ctrl+Space)
year
month
job_industry_category
total_sum
1 2,017 1 Argiculture 481,158.4
2 2,017 1 Entertainment 632,572.25
3 2,017 1 Financial Services 3,987,491
4 2,017 1 Health 3,030,781.2
5 2,017 1 IT 1,257,042.9
6 2,017 1 Manufacturing 3,775,075.5
7 2,017 1 n/a 3,891,781.8
8 2,017 1 Property 907,482.25
9 2,017 1 Retail 2,053,151.6
10 2,017 1 Telecommunications 368,980.8
11 2,017 1 [NULL] 64,759.92
12 2,017 2 Argiculture 740,904.1
13 2,017 2 Entertainment 682,649.6
14 2,017 2 Financial Services 4,209,363

```sql
-- 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.
with brands as (
    select distinct p.brand
    from product p
    where p.brand is not null
),
online_orders_it as (
    select
        p.brand,
        count(distinct o.order_id) as order_count
    from orders o
        join customer c on o.customer_id = c.customer_id
        join order_items oi on o.order_id = oi.order_id
        join product p on oi.product_id = p.product_id
    where c.job_industry_category = 'IT'
      and o.online_order = true
      and o.order_status = 'Approved'
    group by p.brand
)
select
    b.brand,
    coalesce(oit.order_count, 0) as online_order_count
from brands b
left join online_orders_it oit on b.brand = oit.brand
order by b.brand;
```

product 1

Enter a SQL expression to filter results (use Ctrl+Space)

	brand	online_order_count
1		44
2	Giant Bicycles	194
3	Norco Bicycles	173
4	OHM Cycles	174
5	Solex	202
6	Trek Bicycles	160
7	WeareA2B	171

```sql
-- 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.
-- Через Group By
select
    c.customer_id,
    c.first_name,
    c.last_name,
    sum(oi.quantity * oi.item_list_price_at_sale) as sum_sales,
    max(oi.quantity * oi.item_list_price_at_sale) as max_sales,
    min(oi.quantity * oi.item_list_price_at_sale) as min_sales,
    count(distinct o.order_id) as count_ordes,
    avg(oi.quantity * oi.item_list_price_at_sale) as avg_sales
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
    order by sum_sales, count_ordes
```

customer 1

Enter a SQL expression to filter results (use Ctrl+Space)

	customer_id	first_name	last_name	sum_sales	max_sales	min_sales	count_ordes	avg_s
1	3,292	Hamlen	Slograve	60.34	60.34	60.34	1	60.3
2	2,532	Milli	Hubbert	71.49	71.49	71.49	1	71.4
3	301	Hildy	Bilbrook	432.94	290.62	142.32	2	216.4
4	2,089	Erie	Ballston	598	416.98	181.02	2	299.0
5	784	Frederik	Duckett	796.20996	748.17	48.04	2	398.1
6	2,021	Ulrica	Abelwhite	802.88	742.54	60.34	2	40
7	1,325	Ty	Belliard	833.96	833.96	833.96	1	833.9
8	2,274	Nada	Reinert	857.87994	714.89996	142.98	2	428.9
9	1,846	Lorette	Daspar	882.98	882.98	882.98	1	882.9
10	1,796	Cayla	Marcus	942.52	706.89	235.63	2	471.2
11	2,394	Ferne	Reese	980.37	980.37	980.37	1	980.3
12	3,189	Abbott	Knaggs	1,003.5	702.45	301.05	2	
13	1,284	Alexis	Aronov	1,037.53	527.67	48.04	3	345.8
14	191	Christopher	Heining	1,065.03	1,065.03	1,065.03	1	1,065.0

Value

3292

```sql
-- 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_total_amount as (
    select
        c.customer_id,
        c.first_name,
        c.last_name,
        coalesce(sum(oi.quantity * oi.item_list_price_at_sale), 0) as total_amount
    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
```

Results 1

Enter a SQL expression to filter results (use Ctrl+Space)

	first_name	last_name	total_amount
1	Vale	Dagnan	0
2	Carlyle	Frape	0
3	Kelvin	Oldford	0
4	Tye	Doohan	129,789.945
5	Jeffry	Slowly	133,657.06
6	Jillie	Fyndon	136,632.45

Value

Text

Vale

```sql
-- 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 numbered_orders as (
    select
        c.customer_id,
        c.first_name,
        c.last_name,
        o.order_id,
        o.order_date,
        sum(oi.quantity * oi.item_list_price_at_sale) as sum_sales,
        row_number() over (partition by c.customer_id order by o.order_date, o.order_id) as rn
    from customer c
        join orders o on c.customer_id = o.customer_id
        join order_items oi on o.order_id = oi.order_id
    group by
        c.customer_id,
        c.first_name,
        c.last_name,
        o.order_id,
        o.order_date
)
select
    customer_id,
    first_name,
    last_name,
    order_id,
    order_date,
    sum_sales
from numbered_orders
where rn = 2;
```

customer(+) 1

Enter a SQL expression to filter results (use Ctrl+Space)

	customer_id	first_name	last_name	order_id	order_date	sum_sales
1	1	Laraine	Medendorp	13,424	2017-02-21	714.89996
2	2	Eli	Bockman	6,743	2017-06-11	5,268.48
3	3	Arlin	Dearle	15,188	2017-03-24	2,297.28
4	4	Talbot		14,648	2017-06-18	2,868.96
5	5	Sheila-kathryn	Calton	19,993	2017-04-28	11,638.9
6	6	Curr	Duckhouse	8,204	2017-02-06	14,122.4
7	7	Fina	Merali	18,549	2017-02-24	60.34
8	8	Rod	Inder	19,844	2017-01-28	2,091.47
9	9	Mala	Lind	2,979	2017-03-06	5,268.48

Value

1

```sql
-- 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 customer_order_date as (
    select
        c.customer_id,
        c.first_name,
        c.last_name,
        c.job_title,
        o.order_id,
        o.order_date::date as order_date
    from customer c
        join orders o on c.customer_id = o.customer_id
),
interval_between_orders as (
    select
        customer_id,
        first_name,
        last_name,
```

customer 1

Enter a SQL expression to filter results (use Ctrl+Space)

	first_name	last_name	job_title	max_interval
1	Susanetta		Legal Assistant	357
2	Royall	Terris	Geological Engineer	330
3	Stoddard	Giacomoni	Structural Analysis Engineer	330
4	Gregorius	Cockram	Data Coordiator	330
5	Bearnard	Letixier		329
6	Caralie	Sellors	Senior Editor	321
7	Debee	Martynov	Senior Editor	320
8	Genni	Larway	Environmental Specialist	314
9	Timmie	Lenden		310
10	Carolynn	Samsin	Pharmacist	310
11	Franz	Craddy		310
12	Jodee	Judkins	Recruiting Manager	306
13	Ashia	Muzzi	Mechanical Systems Engineer	306
14	Heywood	Sollett	Tax Accountant	305
15	Cleveland	Islep	Software Engineer II	299

Value

Text

Susanetta

```sql
-- 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_sum_sales as (
    select
        c.customer_id,
        c.first_name,
        c.last_name,
        c.wealth_segment,
        coalesce(sum(oi.quantity * oi.item_list_price_at_sale), 0) as sum_sales
    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,
        c.wealth_segment
),
customers_rn_segment as (
    select
```

customer 1

Enter a SQL expression to filter results (use Ctrl+Space)

	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	Wendy	Randlesome	Mass Customer	101,439.06
14	Charis	Maas	Mass Customer	100,891.35
15	Cordelia		Mass Customer	99,880.69