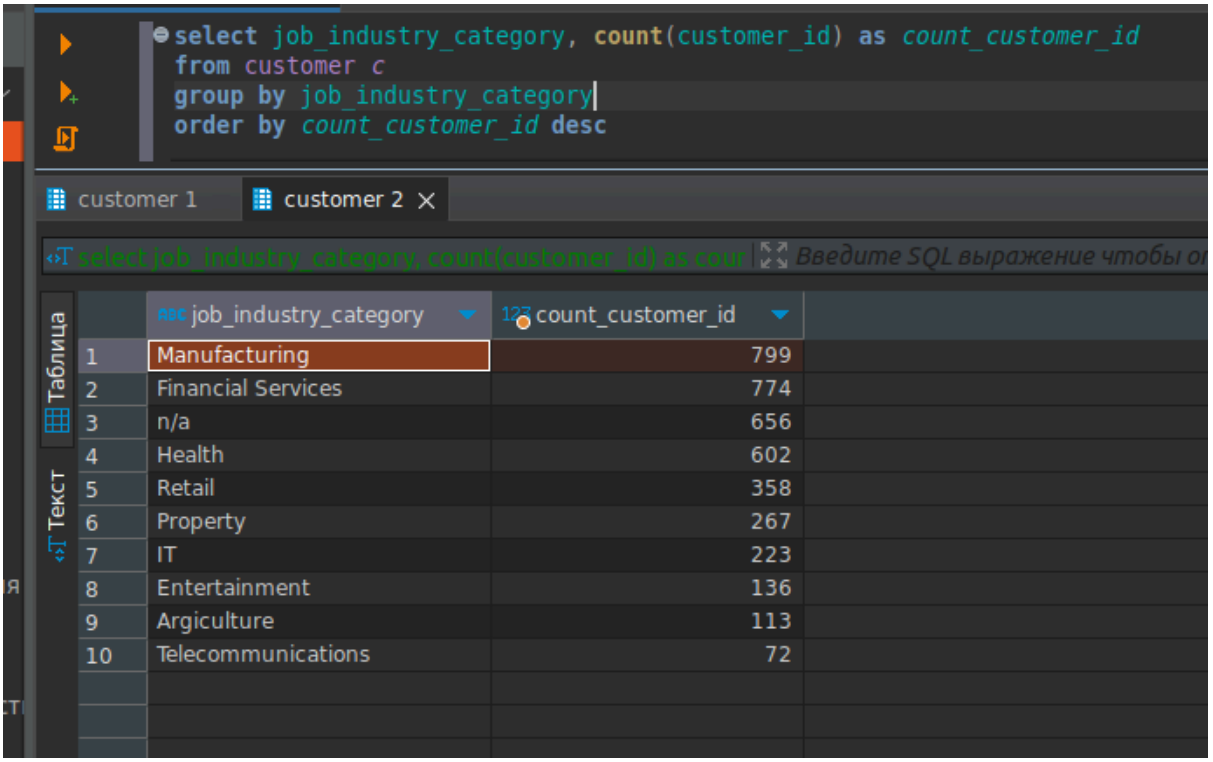
- Вывести распределение (количество) клиентов по сферам деятельности, отсортировав результат по убыванию количества. — **(1 балл)**

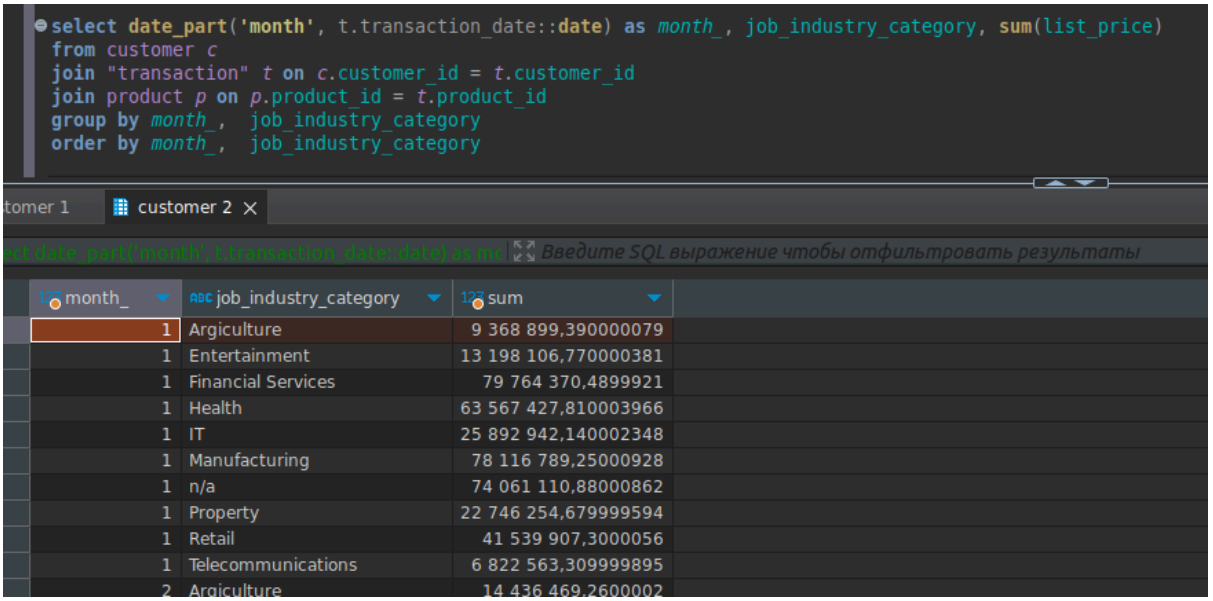


```
select job_industry_category, count(customer_id) as count_customer_id
from customer c
group by job_industry_category
order by count_customer_id desc
```

customer 1 | customer 2

	job_industry_category	count_customer_id
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 сумму транзакций за каждый месяц по сферам деятельности, отсортировав по месяцам и по сфере деятельности. — **(1 балл)**

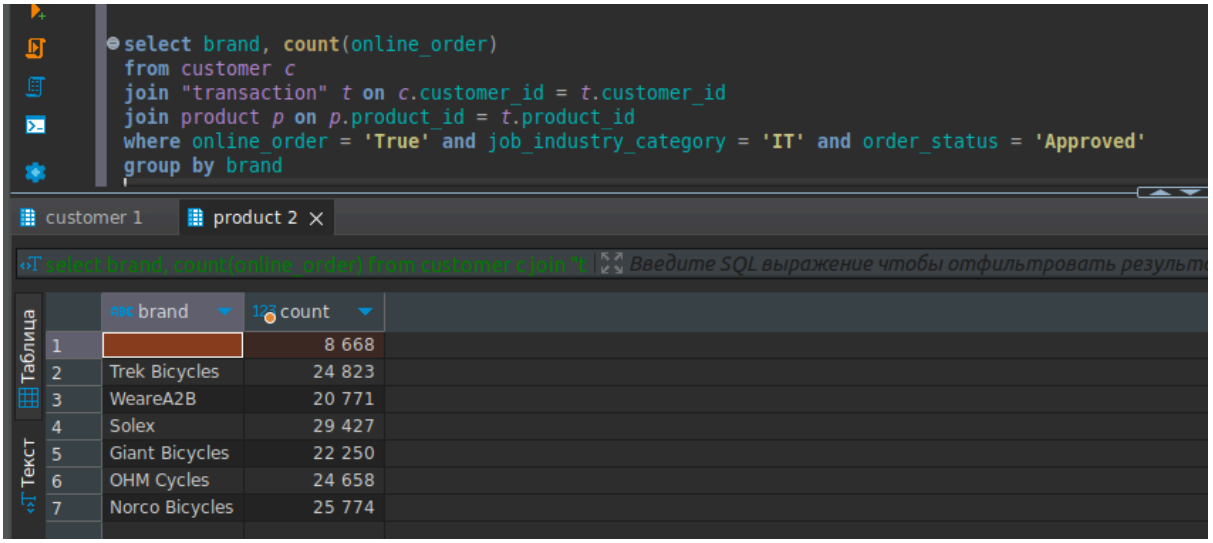


```
select date_part('month', t.transaction_date::date) as month_, job_industry_category, sum(list_price)
from customer c
join "transaction" t on c.customer_id = t.customer_id
join product p on p.product_id = t.product_id
group by month_, job_industry_category
order by month_, job_industry_category
```

customer 2

	month_	job_industry_category	sum
	1	Argiculture	9 368 899,390000079
	1	Entertainment	13 198 106,770000381
	1	Financial Services	79 764 370,4899921
	1	Health	63 567 427,810003966
	1	IT	25 892 942,140002348
	1	Manufacturing	78 116 789,25000928
	1	n/a	74 061 110,88000862
	1	Property	22 746 254,679999594
	1	Retail	41 539 907,3000056
	1	Telecommunications	6 822 563,309999895
	2	Argiculture	14 436 469,2600002

- Вывести количество онлайн-заказов для всех брендов в рамках подтвержденных заказов клиентов из сферы IT. — **(1 балл)**

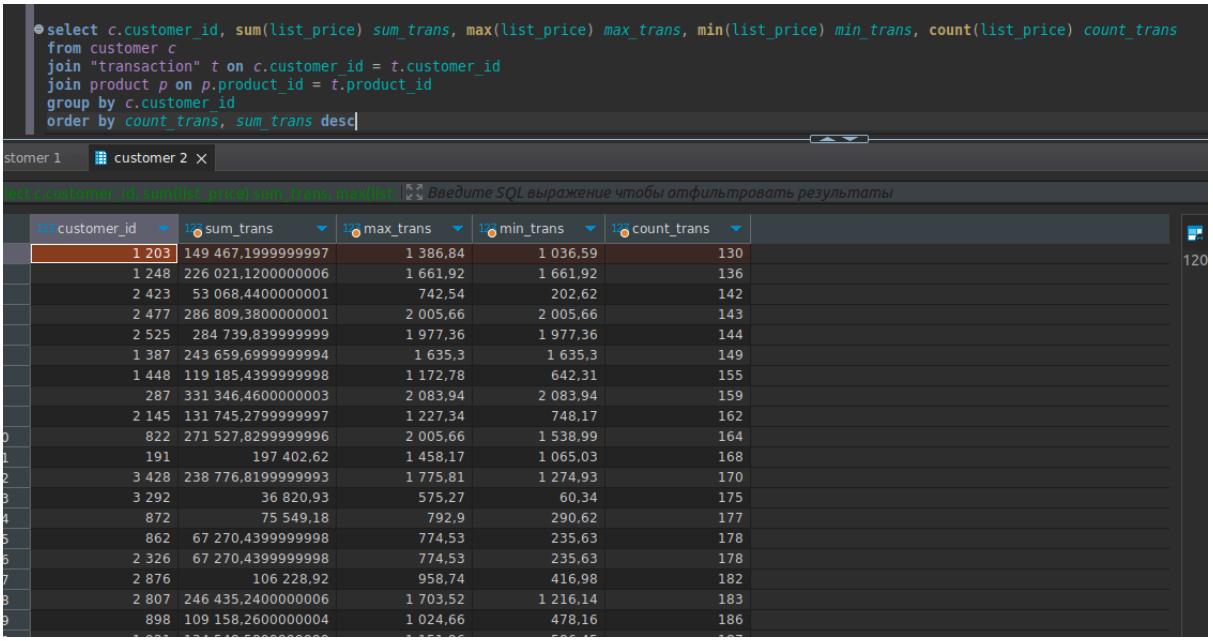


```
select brand, count(online_order)
from customer c
join "transaction" t on c.customer_id = t.customer_id
join product p on p.product_id = t.product_id
where online_order = 'True' and job_industry_category = 'IT' and order_status = 'Approved'
group by brand
```

customer 1 | product 2

	brand	count
1		8 668
2	Trek Bicycles	24 823
3	WeareA2B	20 771
4	Solex	29 427
5	Giant Bicycles	22 250
6	OHM Cycles	24 658
7	Norco Bicycles	25 774

- Найти по всем клиентам сумму всех транзакций (list_price), максимум, минимум и количество транзакций, отсортировав результат по убыванию суммы транзакций и количества клиентов. Выполните двумя способами: используя только group by и используя только оконные функции. Сравните результат. — **(2 балла)**



```
select c.customer_id, sum(list_price) sum_trans, max(list_price) max_trans, min(list_price) min_trans, count(list_price) count_trans
from customer c
join "transaction" t on c.customer_id = t.customer_id
join product p on p.product_id = t.product_id
group by c.customer_id
order by count_trans, sum_trans desc
```

customer 2

customer_id	sum_trans	max_trans	min_trans	count_trans
1 203	149 467,1999999997	1 386,84	1 036,59	130
1 248	226 021,1200000006	1 661,92	1 661,92	136
2 423	53 068,4400000001	742,54	202,62	142
2 477	286 809,3800000001	2 005,66	2 005,66	143
2 525	284 739,839999999	1 977,36	1 977,36	144
1 387	243 659,6999999994	1 635,3	1 635,3	149
1 448	119 185,4399999998	1 172,78	642,31	155
287	331 346,4600000003	2 083,94	2 083,94	159
2 145	131 745,2799999997	1 227,34	748,17	162
822	271 527,8299999996	2 005,66	1 538,99	164
191	197 402,62	1 458,17	1 065,03	168
3 428	238 776,8199999993	1 775,81	1 274,93	170
3 292	36 820,93	575,27	60,34	175
872	75 549,18	792,9	290,62	177
862	67 270,4399999998	774,53	235,63	178
2 326	67 270,4399999998	774,53	235,63	178
2 876	106 228,92	958,74	416,98	182
2 807	246 435,2400000006	1 703,52	1 216,14	183
898	109 158,2600000004	1 024,66	478,16	186

```sql
select distinct(c.customer_id),sum(list_price) over(partition by c.customer_id) sum_trans, max(list_price) over(partition by c.customer_id) max_trans
 , min(list_price) over(partition by c.customer_id) min_trans, count(list_price) over(partition by c.customer_id) count_trans
from customer c
join "transaction" t on c.customer_id = t.customer_id
join product p on p.product_id = t.product_id
order by count_trans, sum_trans desc
```

customer 2

Введите SQL выражение чтобы отфильтровать результаты

	customer_id	sum_trans	max_trans	min_trans	count_trans
	1 203	149 467,1999999997	1 386,84	1 036,59	130
	1 248	226 021,1200000006	1 661,92	1 661,92	136
	2 423	53 068,4400000001	742,54	202,62	142
	2 477	286 809,3800000001	2 005,66	2 005,66	143
	2 525	284 739,839999999	1 977,36	1 977,36	144
	1 387	243 659,6999999994	1 635,3	1 635,3	149
	1 448	119 185,4399999998	1 172,78	642,31	155
	287	331 346,4600000003	2 083,94	2 083,94	159
	2 145	131 745,2799999997	1 227,34	748,17	162
0	822	271 527,8299999996	2 005,66	1 538,99	164
1	191	197 402,62	1 458,17	1 065,03	168
2	3 428	238 776,8199999992	1 775,81	1 274,93	170
3	3 292	36 820,93	575,27	60,34	175
4	872	75 549,18	792,9	290,62	177
5	862	67 270,4399999999	774,53	235,63	178
6	2 326	67 270,4399999998	774,53	235,63	178
7	2 876	106 228,92	958,74	416,98	182
8	2 807	246 435,2400000007	1 703,52	1 216,14	183
9	898	109 158,2600000004	1 024,66	478,16	186
	1 921	134 548,5899999998	1 151,96	586,45	187

Значение

1203

- Найти имена и фамилии клиентов с минимальной/максимальной суммой транзакций за весь период (сумма транзакций не может быть null). Напишите отдельные запросы для минимальной и максимальной суммы. — **(2 балла)**

```sql
select first_name, last_name, min(list_price) min_trans
from customer c
join "transaction" t on c.customer_id = t.customer_id
join product p on p.product_id = t.product_id
where list_price is not null
group by first_name, last_name
```

customer 1 | customer 2

Таблица | Текст

	first_name	last_name	min_trans
1	Jamie	Gildersleeve	1 024,66
2	Brett	McCalister	60,34
3	Isobel	Lampett	12,01
4	Andree	Simonato	12,01
5	Joannes	Harbron	183,86
6	Byrom	Draisey	12,01
7	Melamie	Hardman	12,01
8	Terrijo	Merlin	12,01
9	Kerry	Pashenkov	795,34
10	Banky	Baudichon	586,45
11	Alair	Combe	1 061,56
12	Drake	Riguard	574,64
13	Venus	Sketch	183,86
14	Amanda	Leverington	1 129,13
15	Maressa	Kondrachenko	60,34
16	Agata	Inglesent	1 227,34
17	Alf	Farnie	499,53
18	Karlens	Chaffyn	60,34
19	Rosita	Soppeth	12,01

```sql
select first_name, last_name, max(list_price) min_trans
from customer c
join "transaction" t on c.customer_id = t.customer_id
join product p on p.product_id = t.product_id
where list_price is not null
group by first_name, last_name
```

customer 1 | customer 2

	first_name	last_name	min_trans
1	Jamie	Gildersleeve	2 091,47
2	Brett	McCalister	1 977,36
3	Isobel	Lampett	2 086,07
4	Andree	Simonato	2 086,07
5	Joannes	Harbron	2 083,94
6	Byrom	Draisey	2 086,07
7	Melamie	Hardman	2 086,07
8	Terrijo	Merlin	2 086,07
9	Kerry	Pashenkov	1 890,39
10	Banky	Baudichon	2 091,47
11	Alair	Combe	1 635,3
12	Drake	Riguard	1 977,36
13	Venus	Sketch	1 289,85
14	Amanda	Leverington	2 083,94
15	Maressa	Kondrachenko	2 091,47
16	Agata	Inglesent	2 005,66
17	Alf	Farnie	1 793,43
18	Karlens	Chaffyn	2 005,66
19	Rosita	Soppeth	2 086,07

- Вывести только самые первые транзакции клиентов. Решить с помощью оконных функций. — **(1 балл)**

```sql
select c.customer_id, transaction_id, min(transaction_date) over(partition by c.customer_id) first_trans_dt
from customer c
join "transaction" t on c.customer_id = t.customer_id
```

customer(+) 2

	customer_id	transaction_id	first_trans_dt
	1	14 486	04.06.2017
	1	18 970	04.06.2017
	1	5 157	04.06.2017
	1	9 785	04.06.2017
	1	3 765	04.06.2017
	1	15 663	04.06.2017
	1	13 644	04.06.2017
	1	94	04.06.2017
	1	16 423	04.06.2017
	1	14 931	04.06.2017
	1	13 424	04.06.2017
	2	16 411	04.05.2017
	2	2 261	04.05.2017
	2	6 743	04.05.2017
	3	19 498	04.09.2017
	3	10 302	04.09.2017
	3	16 725	04.09.2017
	3	12 954	04.09.2017
	3	11 005	04.09.2017
	3	11 784	04.09.2017
	3	13 285	04.09.2017
	3	15 188	04.09.2017

- Вывести имена, фамилии и профессии клиентов, между транзакциями которых был максимальный интервал (интервал вычисляется в днях) — **(2 балла)**.