Домашние задание 3

Гриднев К.А.

1. Создать таблицы со следующими структурами и загрузить данные из csv-файлов. Детали приведены ниже.

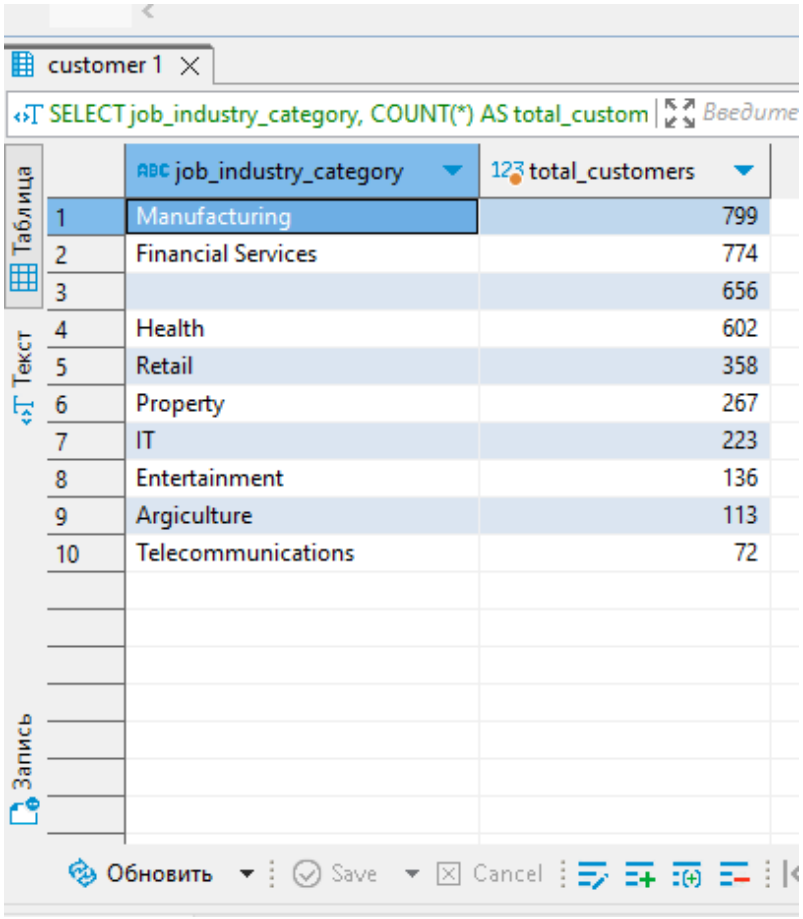
```
CREATE TABLE customer (
customer_id int4
 ,first_name varchar(50)
 ,last_name varchar(50)
 ,gender varchar(30)
 ,DOB varchar(50)
 ,job_title varchar(50)
 ,job_industry_category varchar(50)
 ,wealth_segment varchar(50)
 ,deceased_indicator varchar(50)
 ,owns_car varchar(30)
 ,address varchar(50)
 ,postcode varchar(30)
 ,state varchar(30)
 ,country varchar(30)
 ,property_valuation int4
 ,UNIQUE("customer_id")
)
```

```
CREATE TABLE transaction (
transaction_id int4
 ,product_id int4
 ,customer_id int4
 ,transaction_date varchar(30)
 ,online_order varchar(30)
 ,order_status varchar(30)
 ,brand varchar(30)
 ,product_line varchar(30)
 ,product_class varchar(30)
 ,product_size varchar(30)
 ,list_price float4
 ,standard_cost float4
 ,UNIQUE("transaction_id")
 ,FOREIGN KEY ("customer_id") REFERENCES customer("customer_id")
)
```

2. Выполнить следующие запросы:

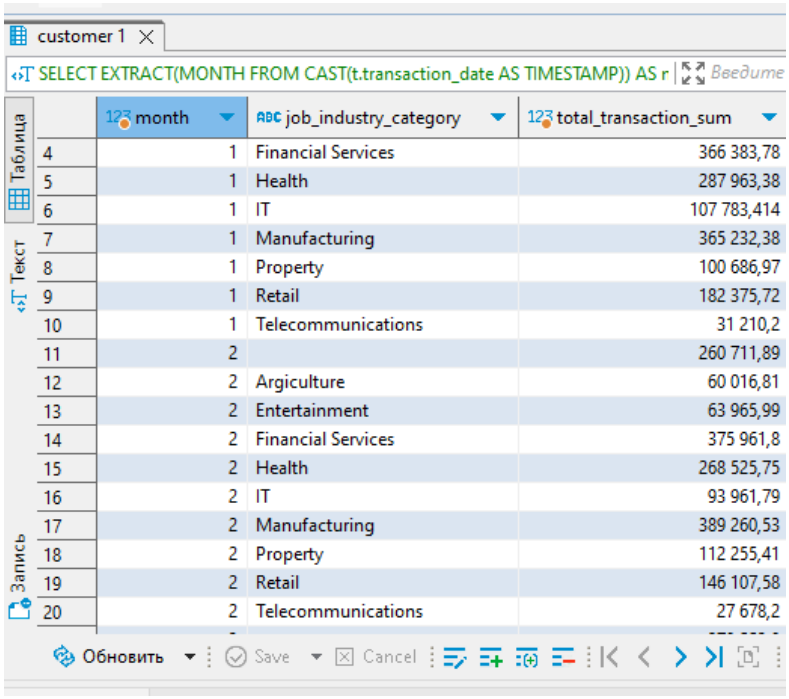
Вывести распределение (количество) клиентов по сферам деятельности, отсортировав результат по убыванию количества.

```
SELECT job_industry_category, COUNT(*) AS total_customers
FROM customer
GROUP BY job_industry_category
ORDER BY total_customers DESC;
```



Найти сумму транзакций за каждый месяц по сферам деятельности, отсортировав по месяцам и по сфере деятельности.

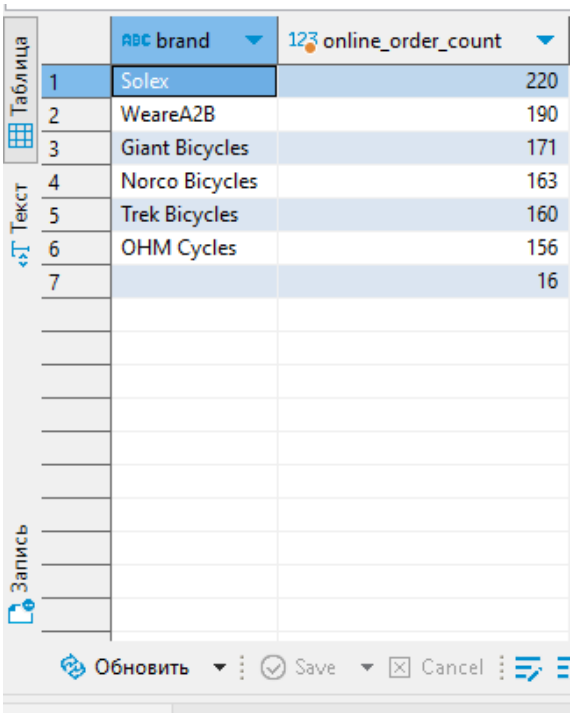
```
SELECT EXTRACT(MONTH FROM CAST(t.transaction_date AS TIMESTAMP)) AS
month,
 c.job_industry_category,
 SUM(t.list_price) AS total_transaction_sum
FROM transaction t
INNER JOIN customer c ON t.customer_id = c.customer_id
GROUP BY EXTRACT(MONTH FROM CAST(t.transaction_date AS TIMESTAMP)),
 c.job_industry_category
ORDER BY month, c.job_industry_category;
```



	month	job_industry_category	total_transaction_sum
4	1	Financial Services	366 383,78
5	1	Health	287 963,38
6	1	IT	107 783,414
7	1	Manufacturing	365 232,38
8	1	Property	100 686,97
9	1	Retail	182 375,72
10	1	Telecommunications	31 210,2
11	2		260 711,89
12	2	Argiculture	60 016,81
13	2	Entertainment	63 965,99
14	2	Financial Services	375 961,8
15	2	Health	268 525,75
16	2	IT	93 961,79
17	2	Manufacturing	389 260,53
18	2	Property	112 255,41
19	2	Retail	146 107,58
20	2	Telecommunications	27 678,2

Вывести количество онлайн-заказов для всех брендов в рамках подтвержденных заказов клиентов из сферы IT

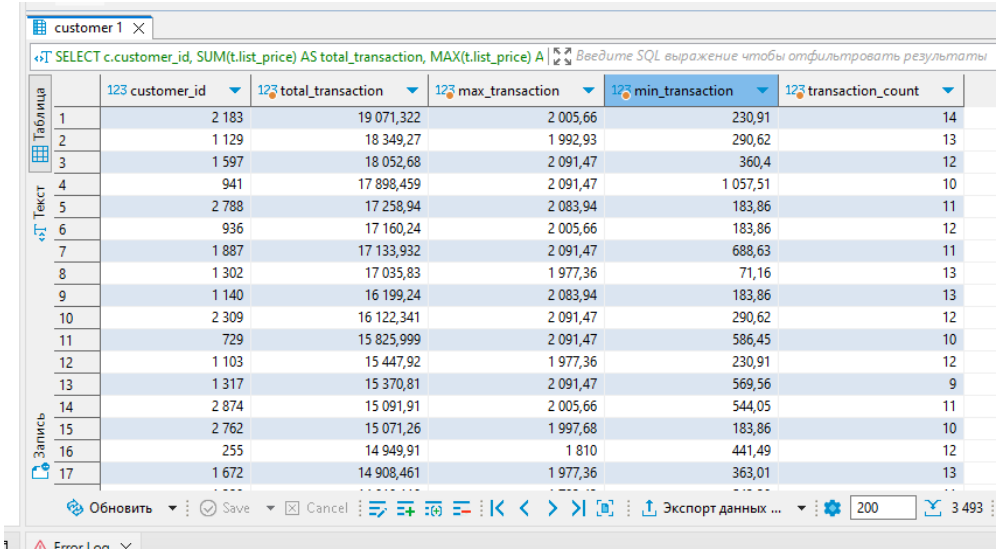
```sql
SELECT t.brand, COUNT(*) AS online_order_count
FROM transaction t
INNER JOIN customer c ON t.customer_id = c.customer_id
WHERE t.order_status = 'Approved' AND c.job_industry_category = 'IT'
GROUP BY t.brand
ORDER BY online_order_count DESC;
```



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

a) Group by

```
SELECT
 c.customer_id,
 SUM(t.list_price) AS total_transaction,
 MAX(t.list_price) AS max_transaction,
 MIN(t.list_price) AS min_transaction,
 COUNT(t.transaction_id) AS transaction_count
FROM customer c
INNER JOIN transaction t ON c.customer_id = t.customer_id
GROUP BY c.customer_id
ORDER BY total_transaction DESC, transaction_count DESC;
```

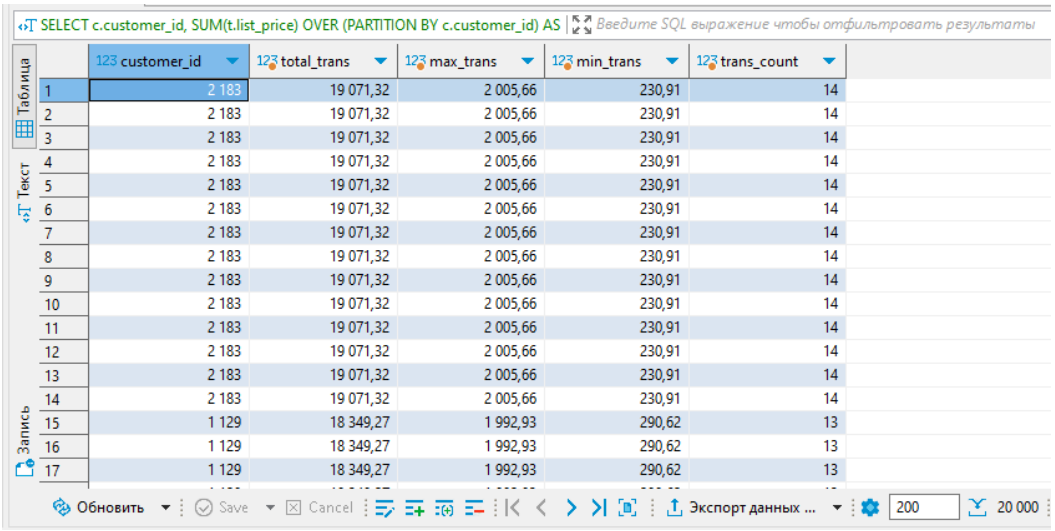


	customer_id	total_transaction	max_transaction	min_transaction	transaction_count
1	2 183	19 071,322	2 005,66	230,91	14
2	1 129	18 349,27	1 992,93	290,62	13
3	1 597	18 052,68	2 091,47	360,4	12
4	941	17 898,459	2 091,47	1 057,51	10
5	2 788	17 258,94	2 083,94	183,86	11
6	936	17 160,24	2 005,66	183,86	12
7	1 887	17 133,932	2 091,47	688,63	11
8	1 302	17 035,83	1 977,36	71,16	13
9	1 140	16 199,24	2 083,94	183,86	13
10	2 309	16 122,341	2 091,47	290,62	12
11	729	15 825,999	2 091,47	586,45	10
12	1 103	15 447,92	1 977,36	230,91	12
13	1 317	15 370,81	2 091,47	569,56	9
14	2 874	15 091,91	2 005,66	544,05	11
15	2 762	15 071,26	1 997,68	183,86	10
16	255	14 949,91	1 810	441,49	12
17	1 672	14 908,461	1 977,36	363,01	13

В случае (a) было получено 3493 записи так как в таблице транзакций есть несколько записей с одинаковым customer_id.

Б)

```
SELECT
 c.customer_id,
 SUM(t.list_price) OVER (PARTITION BY c.customer_id) AS total_trans,
 MAX(t.list_price) OVER (PARTITION BY c.customer_id) AS max_trans,
 MIN(t.list_price) OVER (PARTITION BY c.customer_id) AS min_trans,
 COUNT(*) OVER (PARTITION BY c.customer_id) AS trans_count
FROM
 customer c
 JOIN transaction t ON c.customer_id = t.customer_id
ORDER BY total_trans DESC, trans_count DESC;
```



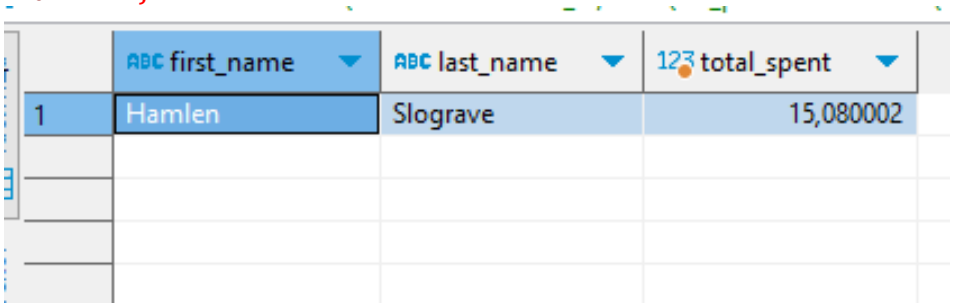
	customer_id	total_trans	max_trans	min_trans	trans_count
1	2 183	19 071,32	2 005,66	230,91	14
2	2 183	19 071,32	2 005,66	230,91	14
3	2 183	19 071,32	2 005,66	230,91	14
4	2 183	19 071,32	2 005,66	230,91	14
5	2 183	19 071,32	2 005,66	230,91	14
6	2 183	19 071,32	2 005,66	230,91	14
7	2 183	19 071,32	2 005,66	230,91	14
8	2 183	19 071,32	2 005,66	230,91	14
9	2 183	19 071,32	2 005,66	230,91	14
10	2 183	19 071,32	2 005,66	230,91	14
11	2 183	19 071,32	2 005,66	230,91	14
12	2 183	19 071,32	2 005,66	230,91	14
13	2 183	19 071,32	2 005,66	230,91	14
14	2 183	19 071,32	2 005,66	230,91	14
15	1 129	18 349,27	1 992,93	290,62	13
16	1 129	18 349,27	1 992,93	290,62	13
17	1 129	18 349,27	1 992,93	290,62	13

В случае (б) было получено 20 000 записей

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

A)- минимальной суммы

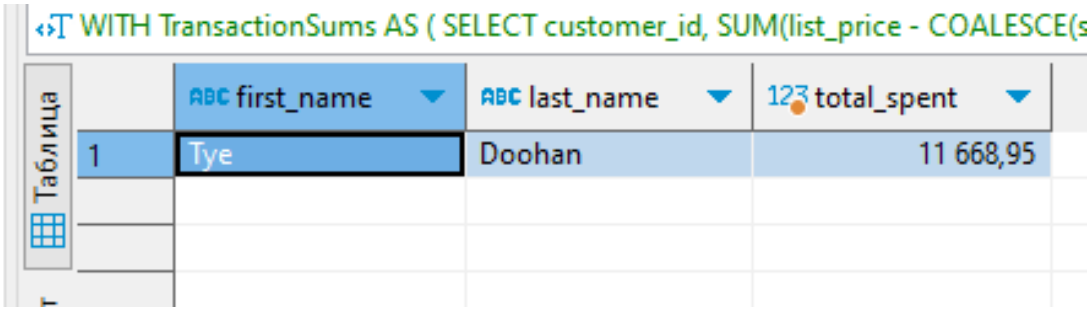
```
WITH TransactionSums AS (
 SELECT
 customer_id,
 SUM(list_price - COALESCE(standard_cost, 0)) AS total_spent
 FROM
 transaction
 GROUP BY
 customer_id
), RankCustomers AS (
 SELECT
 c.first_name,
 c.last_name,
 ts.total_spent,
 RANK() OVER (ORDER BY ts.total_spent ASC) AS rank
 FROM
 TransactionSums ts
 JOIN
 customer c ON ts.customer_id = c.customer_id
)
SELECT
 first_name,
 last_name,
 total_spent
FROM
 RankCustomers
WHERE
 rank = 1;
```



	first_name	last_name	total_spent
1	Hamlen	Slograve	15,080002

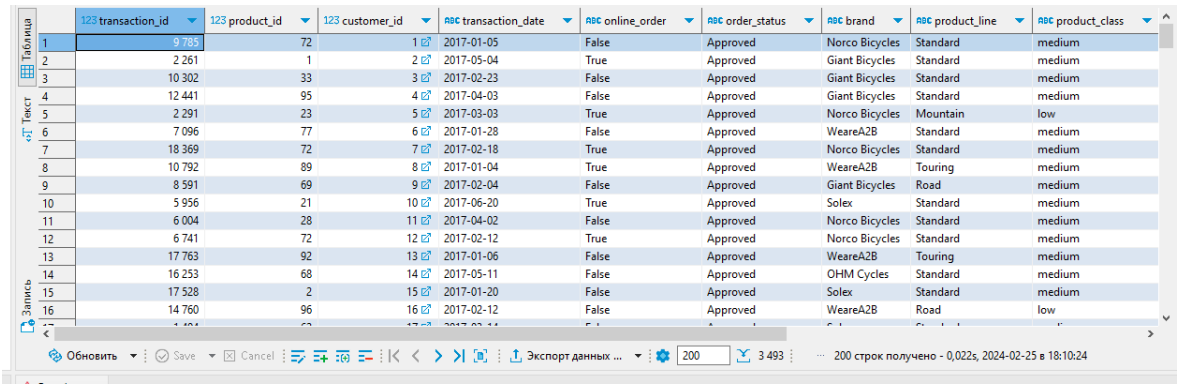
Б) максимальной суммы

```
WITH TransactionSums AS (
 SELECT
 customer_id,
 SUM(list_price - COALESCE(standard_cost, 0)) AS total_spent --
 FROM
 transaction
 GROUP BY
 customer_id
), RankCustomers AS (
 SELECT
 c.first_name,
 c.last_name,
 ts.total_spent,
 RANK() OVER (ORDER BY ts.total_spent DESC) AS rank
 FROM
 TransactionSums ts
 JOIN
 customer c ON ts.customer_id = c.customer_id
)
SELECT
 first_name,
 last_name,
 total_spent
FROM
 RankCustomers
WHERE
 rank = 1;
```



Вывести только самые первые транзакции клиентов. Решить с помощью оконных функций

```
WITH RankTransactions AS (
 SELECT
 t.*,
 ROW_NUMBER() OVER (PARTITION BY t.customer_id ORDER BY
t.transaction_date ASC) AS rk
 FROM
 transaction t
)
SELECT
 *
FROM
 RankTransactions
WHERE
 rk = 1;
```



	transaction_id	product_id	customer_id	transaction_date	online_order	order_status	brand	product_line	product_class
1	9 785	72	1	2017-01-05	False	Approved	Norco Bicycles	Standard	medium
2	2 261	1	2	2017-05-04	True	Approved	Giant Bicycles	Standard	medium
3	10 302	33	3	2017-02-23	False	Approved	Giant Bicycles	Standard	medium
4	12 441	95	4	2017-04-03	False	Approved	Giant Bicycles	Standard	medium
5	2 291	23	5	2017-03-03	True	Approved	Norco Bicycles	Mountain	low
6	7 096	77	6	2017-01-28	False	Approved	WeareA2B	Standard	medium
7	18 369	72	7	2017-02-18	True	Approved	Norco Bicycles	Standard	medium
8	10 792	89	8	2017-01-04	True	Approved	WeareA2B	Touring	medium
9	8 591	69	9	2017-02-04	False	Approved	Giant Bicycles	Road	medium
10	5 956	21	10	2017-06-20	True	Approved	Solex	Standard	medium
11	6 004	28	11	2017-04-02	False	Approved	Norco Bicycles	Standard	medium
12	6 741	72	12	2017-02-12	True	Approved	Norco Bicycles	Standard	medium
13	17 763	92	13	2017-01-06	False	Approved	WeareA2B	Touring	medium
14	16 253	68	14	2017-05-11	False	Approved	OHM Cycles	Standard	medium
15	17 528	2	15	2017-01-20	False	Approved	Solex	Standard	medium
16	14 760	96	16	2017-02-12	False	Approved	WeareA2B	Road	low

Вывести имена, фамилии и профессии клиентов, между транзакциями которых был максимальный интервал (интервал вычисляется в днях)

```
WITH TransactionIntervals AS (
    SELECT
        customer_id,
        transaction_date,
        LEAD(transaction_date::date) OVER (PARTITION BY customer_id ORDER BY
transaction_date::date) - transaction_date::date AS interval_days
    FROM
        transaction
), MaxIntervals AS (
    SELECT
        customer_id,
        MAX(interval_days) AS max_interval
    FROM
        TransactionIntervals
    GROUP BY
        customer_id
), MaxInterval AS (
    SELECT
        MAX(max_interval) AS max_interval
    FROM
        MaxIntervals
)
SELECT
    c.first_name,
    c.last_name,
    c.job_title
FROM
    customer c
JOIN
    MaxIntervals mi ON c.customer_id = mi.customer_id
JOIN
    MaxInterval mi2 ON mi.max_interval = mi2.max_interval;
```