МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Челябинский государственный университет»**

**(ФГБОУ ВО «ЧелГУ»)**

Институт информационных технологий

Кафедра информационных технологий и экономической информатики

ОТЧЕТ

по лабораторной работе №5

Автор отчета: Горюнова Анастасия БИ-201

Преподаватель: Барабанщиков Игорь Витальевич

Челябинск 2022 г.

## Лабораторная работа 5. Коррелированные подзапросы. Транспонирование таблиц

**Задания для схемы MARKET:**

1. **Напишите запрос с коррелированным подзапросом, который выводит информацию о покупателях, рейтинг которых равен максимальному рейтингу в их городе.**

select raiting, cnum, cname, city

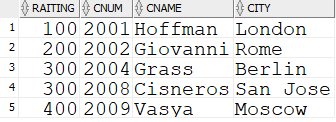
from MARKET.customers, (select max(raiting)as maxx, city AS new\_city

from MARKET.customers

group by city) max\_raiting

where raiting = max\_raiting.maxx

and city = new\_city



1. **Выведите информацию о продавцах, в чьих городах есть покупатели, которых они не обслуживают, учитывайте и тех покупателей, которым еще не назначен продавец (NULL). Решите задачу двумя способами – используя соединение или используя коррелированный подзапрос.**

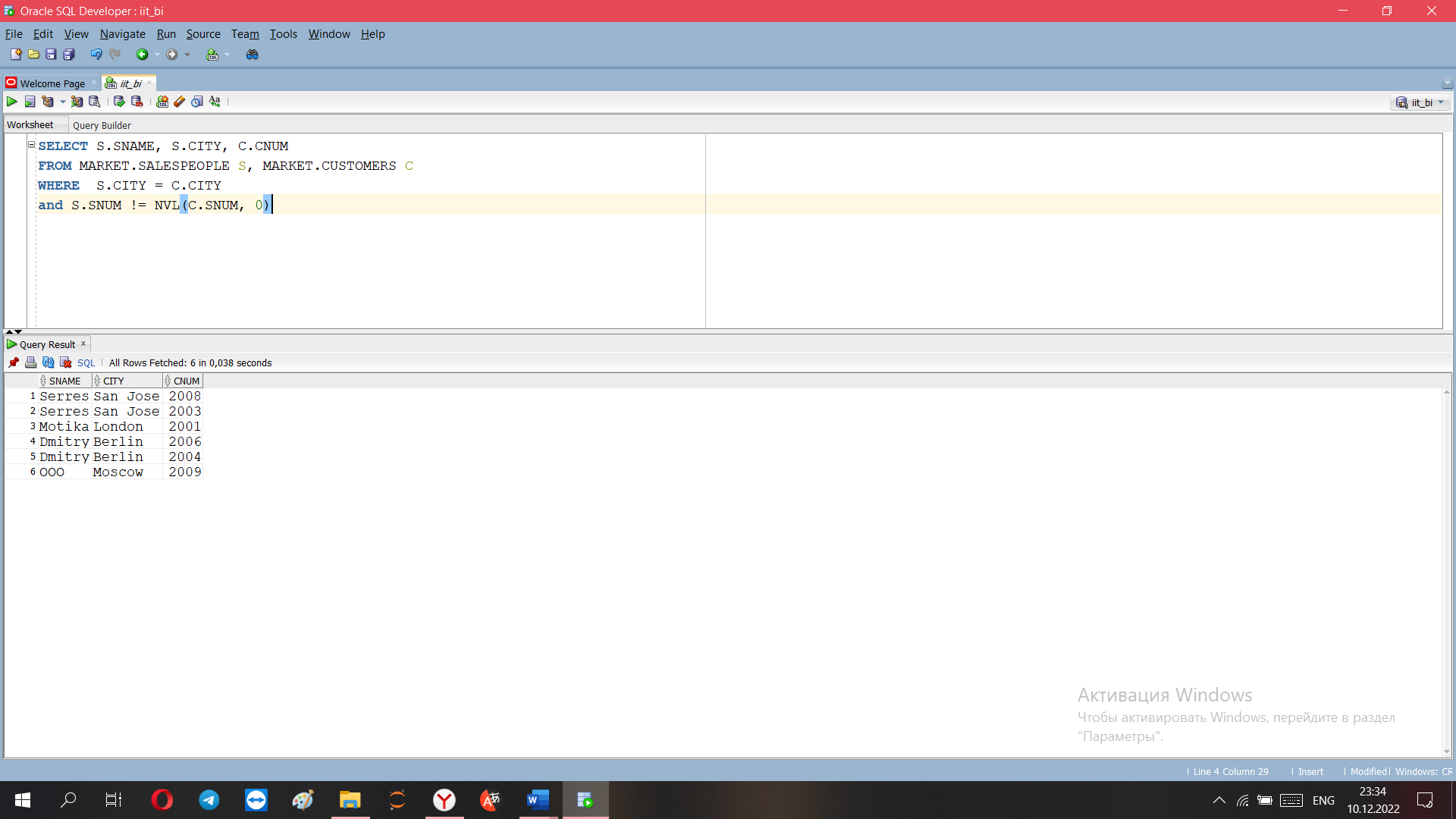
**1 способ – соединение:**

SELECT S.SNAME, S.CITY, C.CNUM

FROM MARKET.SALESPEOPLE S, MARKET.CUSTOMERS C

WHERE S.CITY = C.CITY

and S.SNUM != NVL(C.SNUM, 0)



**2 способ – подзапрос:**

SELECT S.SNAME, S.CITY, C.CNUM

FROM MARKET.SALESPEOPLE S , MARKET.CUSTOMERS C

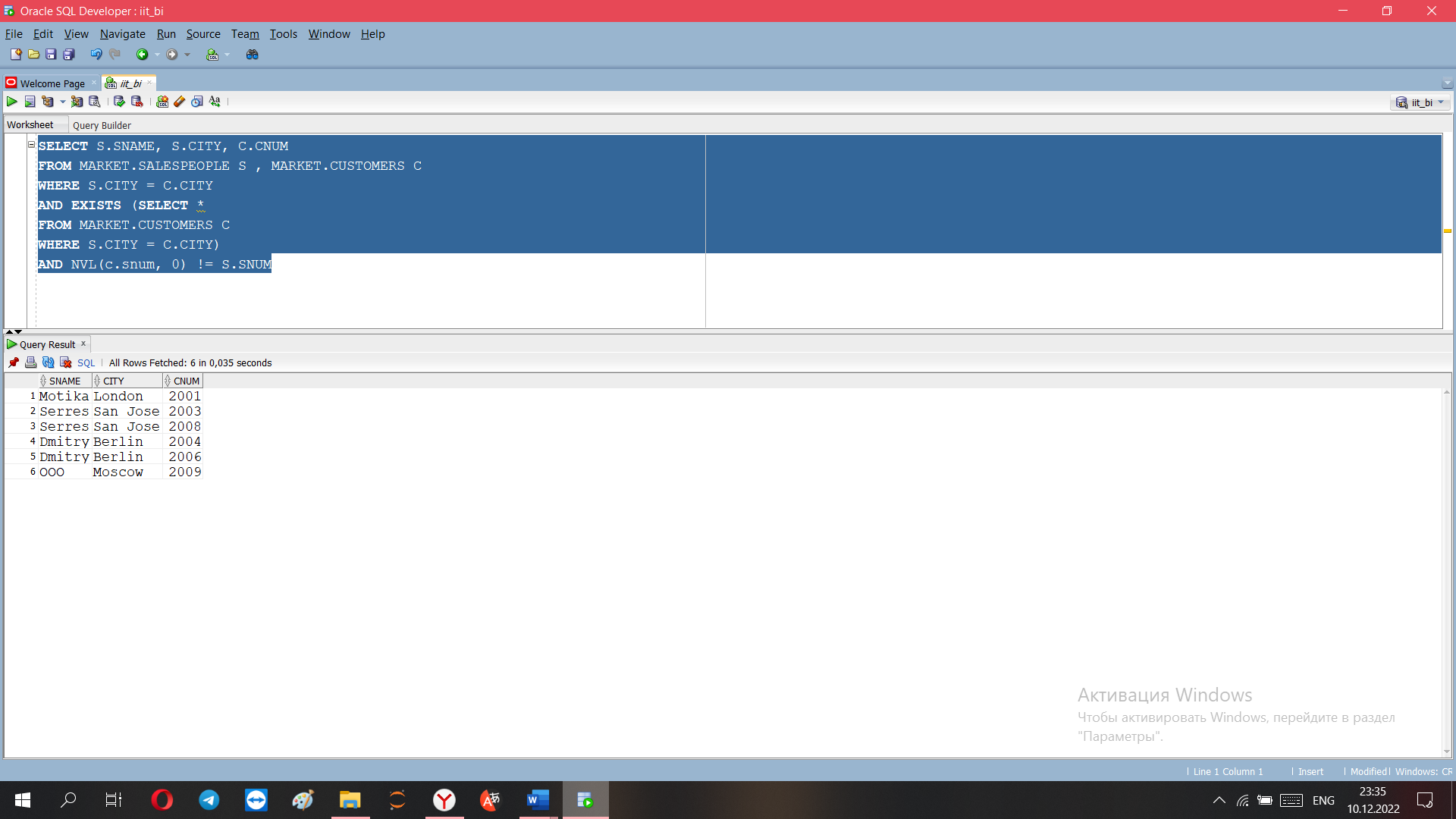
WHERE S.CITY = C.CITY

AND EXISTS (SELECT \*

FROM MARKET.CUSTOMERS C

WHERE S.CITY = C.CITY)

AND NVL(c.snum, 0) != S.SNUM



1. **Выведите информацию о продавцах, которые не выполняли заказы с марта по апрель 2000 года. Используйте коррелированный подзапрос и оператор EXISTS.**

select distinct ss.sname

from MARKET.salespeople ss, MARKET.orders ord

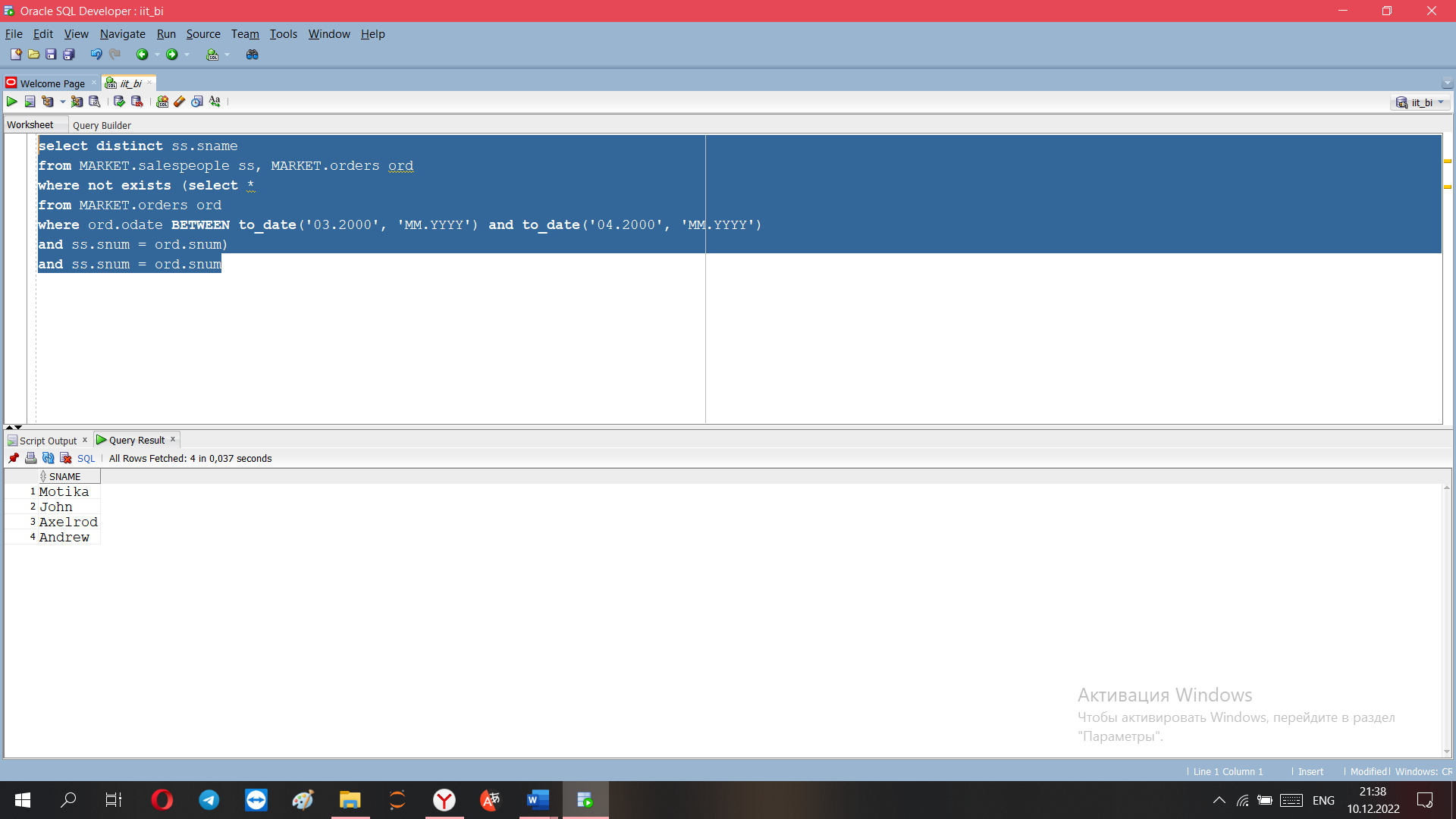
where not exists (select \*

from MARKET.orders ord

where ord.odate BETWEEN to\_date('03.2000', 'MM.YYYY') and to\_date('04.2000', 'MM.YYYY')

and ss.snum = ord.snum)

and ss.snum = ord.snum



1. **Выведите информацию о продавцах, которые в период с февраля по май 2000 года, совершали заказы со средним чеком выше среднего среди продавцов в их городе.**

select avg\_amt2, avg\_amt1, snum, name, city2

from (select round(avg(ord.amt)) AS avg\_amt1, cust.city city1

from MARKET.orders ord, MARKET.customers cust

where ord.odate > to\_date('01.00', 'MM.YY')

and ord.odate < to\_date('06.00', 'MM.YY')

and ord.cnum = cust.cnum

group by cust.city) avg\_city,

(select ord.snum AS snum, round(avg(ord.amt)) AS avg\_amt2, ss.sname AS name, ss.city city2

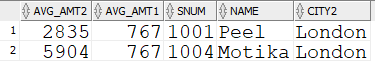
from MARKET.orders ord, MARKET.salespeople ss

where ss.snum = ord.snum

group by ord.snum, ss.sname, ss.city) avg\_orders

where avg\_orders.city2 = avg\_city.city1

and avg\_amt2 > avg\_amt1



1. **Выведите информацию о продавце и количестве выполненных им заказов (только количество – не сумму!), а также вычислите для каждого продавца, какую долю (в %) составляет количество его заказов от общего количества заказов среди продавцов в его городе. Отсортируйте результаты по городу в алфавитном порядке, а затем по доле продавца по убыванию. Используйте коррелированный подзапрос в списке выходных столбцов SELECT.**

Select s.\*,

(select count(o.snum)

from market.orders o

where o.snum = s.snum

group by o.snum) as count,

(select count(\*)

from market.salespeople s1, market.orders o

where s1.snum = o.snum

and s1.city = s.city group by s1.city) as amount\_city,

round((select count(o.snum)

from market.orders o

where o.snum = s.snum

group by o.snum)\*100/( select count(\*)

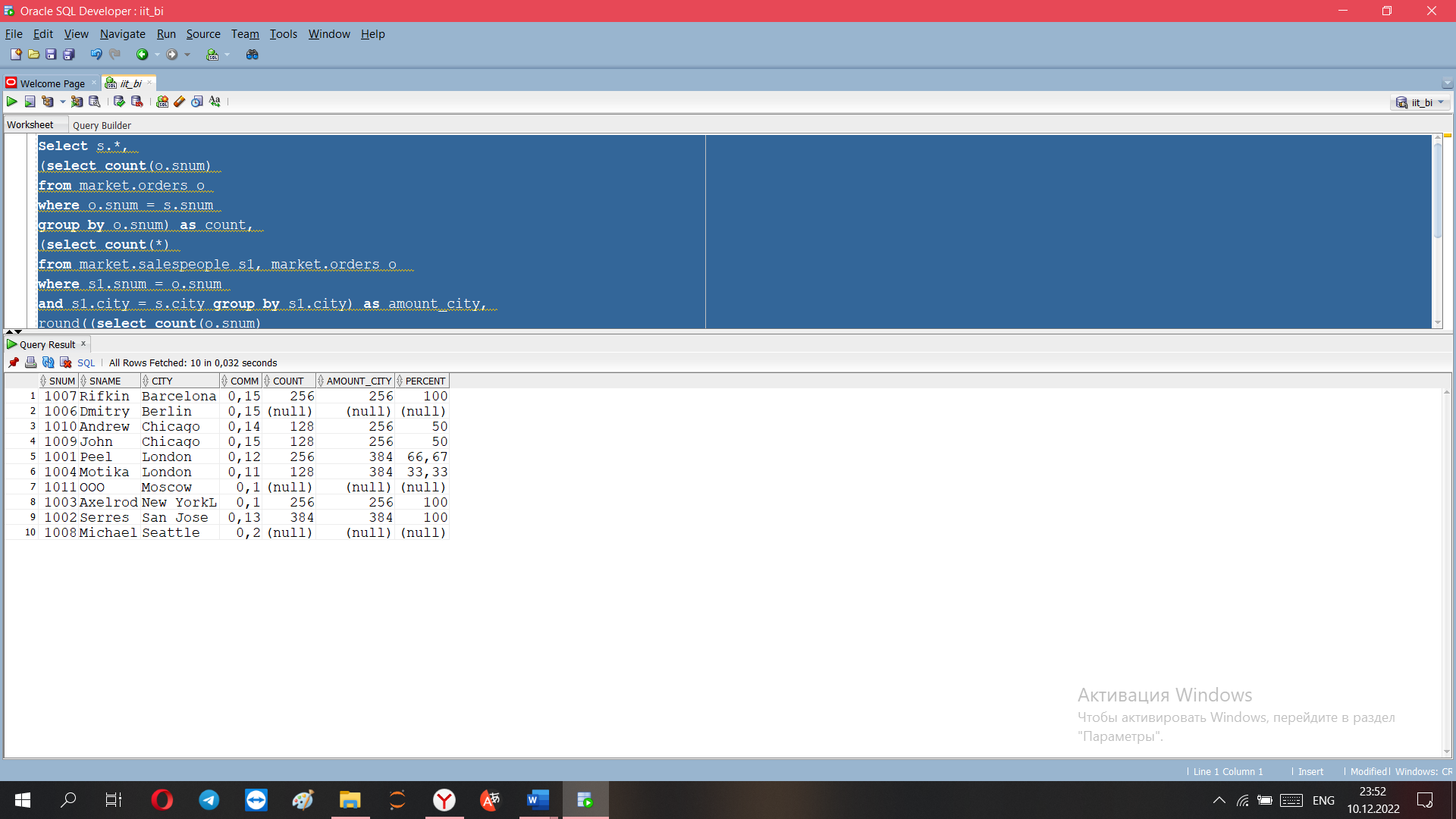
from market.salespeople s1, market.orders o

where s1.snum = o.snum

and s1.city = s.city group by s1.city),2) as percent

from market.salespeople s

order by s.city, percent desc



1. **С помощью транспонирования таблицы и подзапросов – выведите информацию о суммах заказах по каждому месяцу 2000 года (по столбцам) для каждого из продавцов (по строкам):**

select od.cnum,

sum(decode(to\_char(od.odate, 'MM'), 1, od.amt)) jan,

sum(decode(to\_char(od.odate, 'MM'), 2, od.amt)) feb,

sum(decode(to\_char(od.odate, 'MM'), 3, od.amt)) mart,

sum(decode(to\_char(od.odate, 'MM'), 4, od.amt)) apr,

sum(decode(to\_char(od.odate, 'MM'), 5, od.amt)) may,

sum(decode(to\_char(od.odate, 'MM'), 6, od.amt)) jun,

sum(decode(to\_char(od.odate, 'MM'), 7, od.amt)) july,

sum(decode(to\_char(od.odate, 'MM'), 8, od.amt)) aug,

sum(decode(to\_char(od.odate, 'MM'), 9, od.amt)) sep,

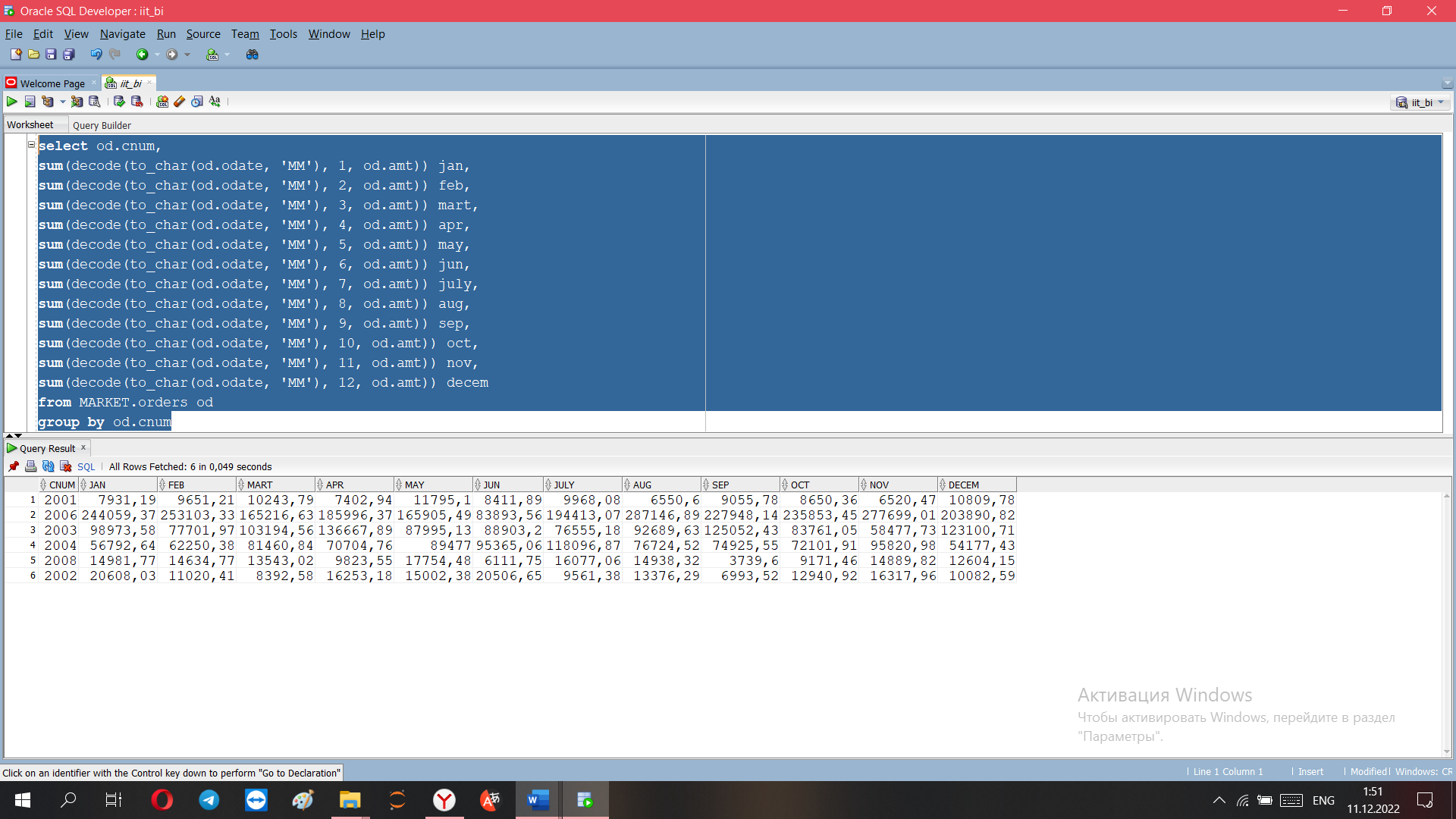
sum(decode(to\_char(od.odate, 'MM'), 10, od.amt)) oct,

sum(decode(to\_char(od.odate, 'MM'), 11, od.amt)) nov,

sum(decode(to\_char(od.odate, 'MM'), 12, od.amt)) decem

from MARKET.orders od

group by od.cnum



**Задания для схемы WATER:**

1. **Выведите номер, ФИО и адрес клиентов, которые хотя бы в один из месяцев имели долг по оплате. Решите задачу двумя способами – используя соединение или подзапрос.**

**1 способ – соединение:**

select cl.n\_client, cl.c\_last\_name, cl.c\_first\_name, cl.c\_second\_name, city.c\_city, street.c\_street, cl.c\_house, cl.n\_flat, min(com.n\_total)

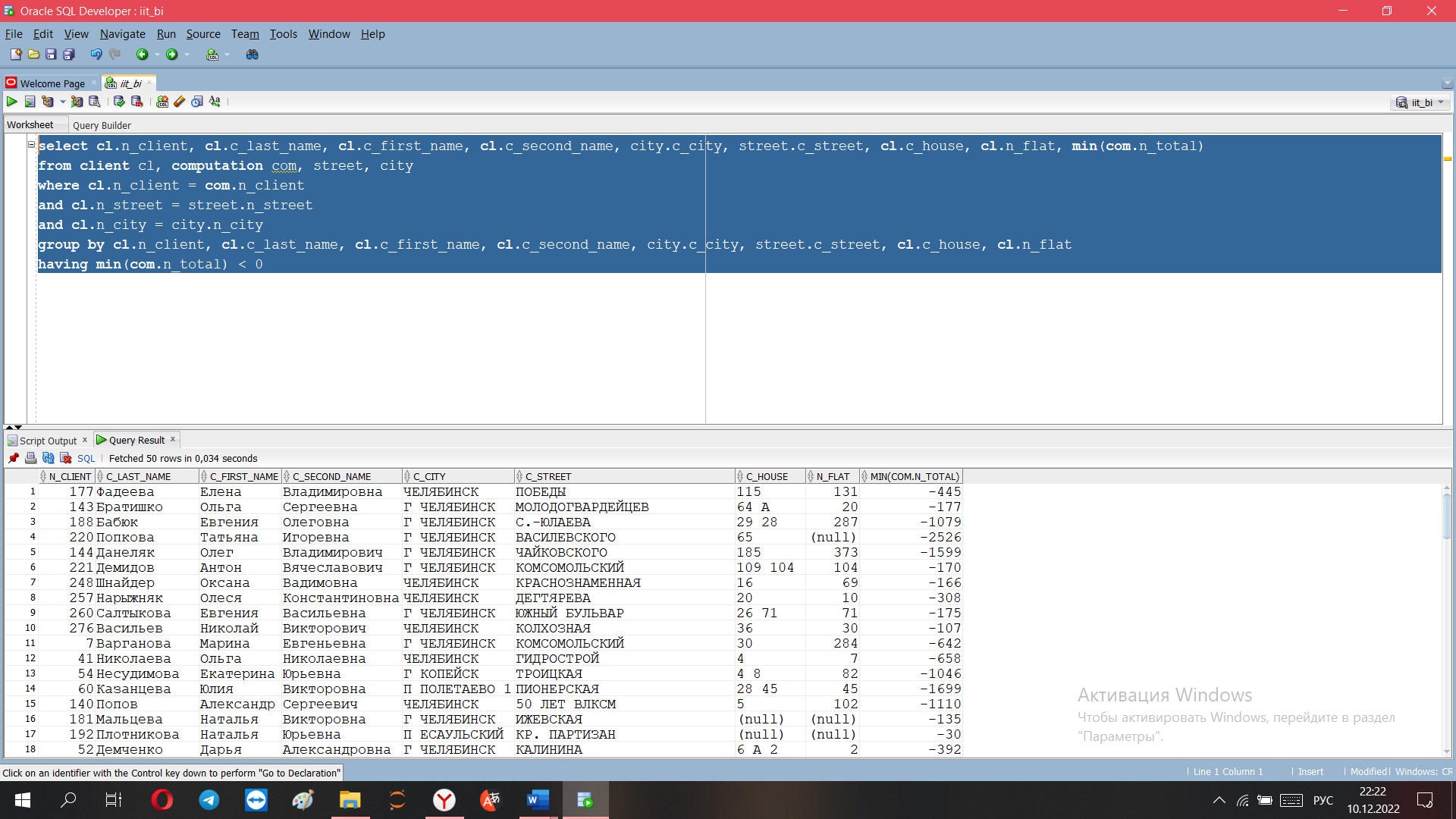
from client cl, computation com, street, city

where cl.n\_client = com.n\_client

and cl.n\_street = street.n\_street

and cl.n\_city = city.n\_city

group by cl.n\_client, cl.c\_last\_name, cl.c\_first\_name, cl.c\_second\_name, city.c\_city, street.c\_street, cl.c\_house, cl.n\_flat

having min(com.n\_total) < 0  


**2 способ – подзапрос:**

select \*

from (select cl.n\_client, cl.c\_last\_name, cl.c\_first\_name, cl.c\_second\_name, city.c\_city,street.c\_street,cl.c\_house,cl.n\_flat, min(com.n\_total) AS min\_total

from client cl, computation com, street, city

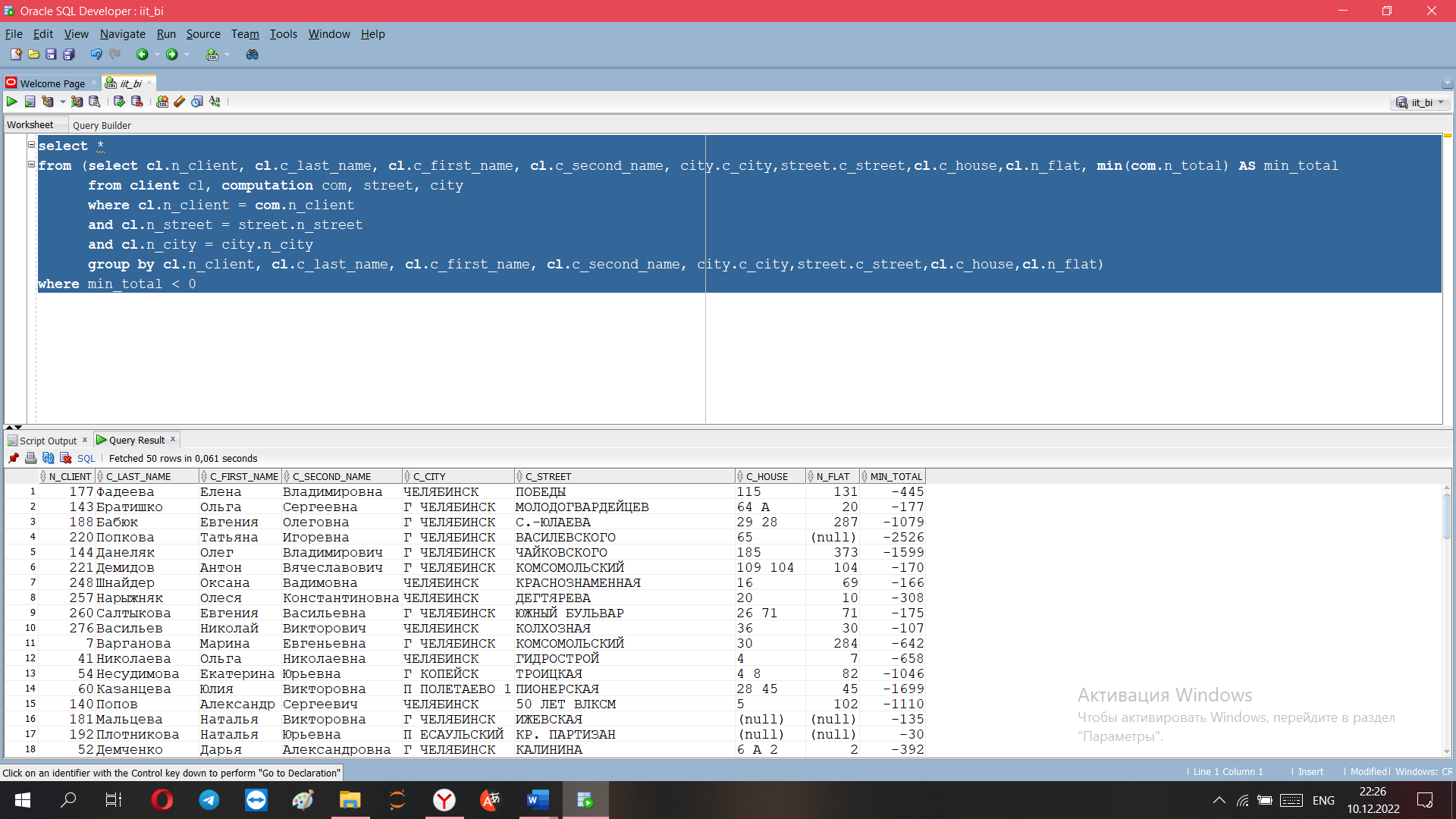
where cl.n\_client = com.n\_client

and cl.n\_street = street.n\_street

and cl.n\_city = city.n\_city

group by cl.n\_client, cl.c\_last\_name, cl.c\_first\_name, cl.c\_second\_name, city.c\_city,street.c\_street,cl.c\_house,cl.n\_flat)

where min\_total < 0



1. **Выведите информацию о клиентах, у которых средний размер ежемесячного платежа за весь период на 20% превышает средний размер ежемесячных платежей всех клиентов в их городе.**

select client, FIO, city1, avg\_client, avg\_city\*1.2

from (select cl.n\_client AS client, cl.c\_last\_name||' '||cl.c\_first\_name||' '||cl.c\_second\_name AS FIO, cl.n\_city AS city1, round(avg(p.n\_sum)) AS avg\_client

from client cl, payment p

where cl.n\_client=p.n\_client

group by cl.n\_client, cl.c\_last\_name||' '||cl.c\_first\_name||' '||cl.c\_second\_name, cl.n\_city) tabl1,

(select round(avg(p.n\_sum)) AS avg\_city, c.n\_city AS city2

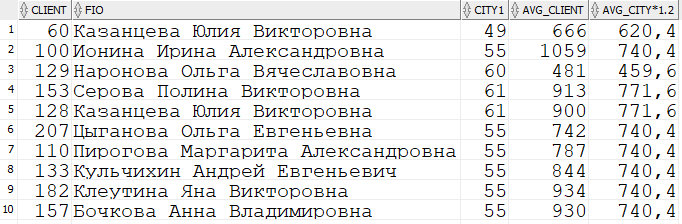
from payment p, client c

where c.n\_client=p.n\_client

group by c.n\_city) tabl2

where city1=city2

and avg\_client>(avg\_city\*(1.2))



1. **Выведите сумму долгов по оплате клиентов, проживающих по одному адресу, (не учитывая переплаты добропорядочных клиентов) по каждому месяцу (по столбцам) для каждого из адресов (город-улица-дом) (по строкам)**

select cl.n\_city,cl.n\_street, cl.c\_house,

sum(decode(to\_char(c.d\_computation, 'MM'), 1, c.n\_debt)) jan,

sum(decode(to\_char(c.d\_computation, 'MM'), 2, c.n\_debt)) feb,

sum(decode(to\_char(c.d\_computation, 'MM'), 3, c.n\_debt)) mart,

sum(decode(to\_char(c.d\_computation, 'MM'), 4, c.n\_debt)) apr,

sum(decode(to\_char(c.d\_computation, 'MM'), 5, c.n\_debt)) may,

sum(decode(to\_char(c.d\_computation, 'MM'), 6, c.n\_debt)) jun,

sum(decode(to\_char(c.d\_computation, 'MM'), 7, c.n\_debt)) july,

sum(decode(to\_char(c.d\_computation, 'MM'), 8, c.n\_debt)) aug,

sum(decode(to\_char(c.d\_computation, 'MM'), 9, c.n\_debt)) sep,

sum(decode(to\_char(c.d\_computation, 'MM'), 10, c.n\_debt)) oct,

sum(decode(to\_char(c.d\_computation, 'MM'), 11, c.n\_debt)) nov,

sum(decode(to\_char(c.d\_computation, 'MM'), 12, c.n\_debt)) decem

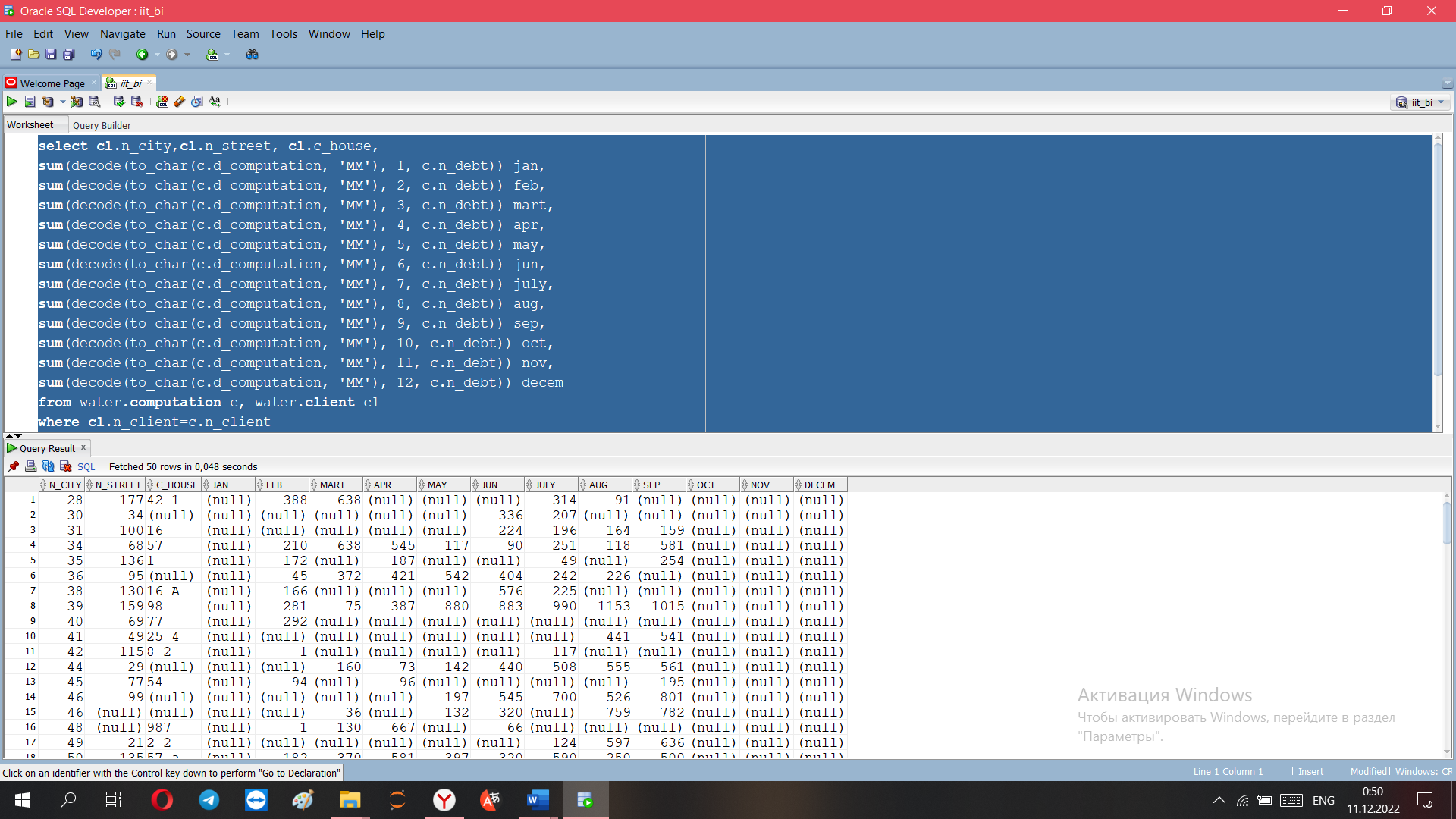
from water.computation c, water.client cl

where cl.n\_client=c.n\_client

and c.n\_debt > 0

group by cl.n\_city, cl.n\_street, cl.c\_house

order by 1



1. **\*Для каждого из клиентов выведите информацию о суммарном потреблении горячей и холодной воды за весь период и о том, какую долю это потребление составляет от потребления всех клиентов, проживающих по этому адресу. Результат отсортировать по адресу - городу, улице, дому. Исключить адреса, по которым проживает только один клиент.**

select com1.n\_client,

c1.c\_city ||' '|| s1.c\_street ||' '||cl.c\_house as adress,

sum(com1.n\_difference\_cold + com1.n\_difference\_hot) as sum\_cl,

round(sum(com1.n\_difference\_cold+com1.n\_difference\_hot)/(

select sum(com.n\_difference\_cold+com.n\_difference\_hot)

from water.computation com, water.client cll, water.city c, water.street s

where cll.n\_client = com.n\_client

and c.n\_city = cll.n\_city

and s.n\_street = cll.n\_street

and c1.c\_city ||' '|| s1.c\_street ||' '||cl.c\_house = c.c\_city ||' '||s.c\_street ||' '||cll.c\_house)\*100 ,3) as percent

from water.computation com1, water.client cl, water.city c1, water.street s1

where cl.n\_client = com1.n\_client

and c1.n\_city = cl.n\_city

and s1.n\_street = cl.n\_street

and (c1.c\_city ||' '|| s1.c\_street ||' '||cl.c\_house) in (

select c2.c\_city ||' '|| s2.c\_street ||' '||cl2.c\_house as adress

from water.computation com2, water.client cl2, water.city c2, water.street s2

where cl2.n\_client = com2.n\_client

and c2.n\_city = cl2.n\_city

and s2.n\_street = cl2.n\_street

group by c2.c\_city ||' '|| s2.c\_street ||' '||cl2.c\_house

having count(distinct com2.n\_client)>1)

group by com1.n\_client, c1.c\_city ||' '|| s1.c\_street ||' '||cl.c\_house

order by adress

