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 №4

Автор отчета: Горюнова Анастасия БИ-201

Преподаватель: Барабанщиков Игорь Витальевич

Челябинск 2022 г.

**Задания:**

1. Вывести все платежи, с указанием даты и долга на указанную дату самого злостного неплательщика.

SELECT pm.d\_pay, pm.n\_sum, com.n\_debt

FROM client cl, computation com, payment pm

WHERE cl.n\_client in

(SELECT cl.n\_client -- находим человека с максимальным долгом

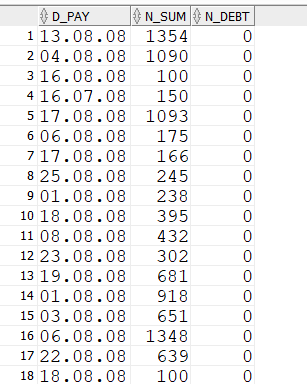
FROM client cl, computation com

WHERE com.n\_debt=

(SELECT max(n\_debt) -- максимальный долг

FROM computation)

AND com.n\_client=cl.n\_client)



1. Найти сколько клиентов живет на каждой улице (*предложить два варианта решения – с использованием группировки и с использованием подзапроса*).

SELECT ST.C\_STREET, COUNT(CL.N\_CLIENT)

FROM CLIENT CL , STREET ST

WHERE CL.N\_STREET = ST.N\_STREET

GROUP BY ST.C\_STREET



SELECT ST.C\_STREET, CL.CLIENT

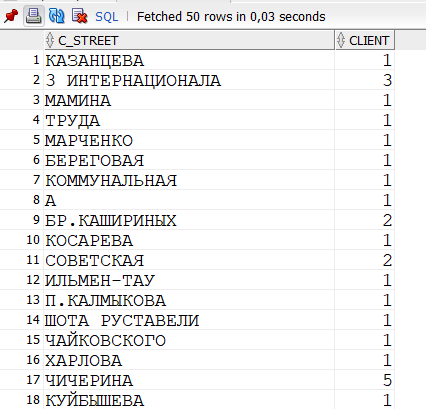
FROM STREET ST ,

(SELECT COUNT(N\_CLIENT) CLIENT, N\_STREET

FROM CLIENT

GROUP BY N\_STREET) CL

WHERE CL.N\_STREET = ST.N\_STREET



1. Вывести сумму платежей по месяцам, поступившим после даты самого большого платежа.

SELECT to\_char(pm.d\_pay,'MM.yyyy'), SUM(pm.n\_sum)

FROM payment pm

WHERE to\_char(pm.d\_pay,'MM.yyyy') >

(

SELECT to\_char(pm.d\_pay,'MM.yyyy')-- месяц и год платежа

FROM payment pm

where pm.n\_sum =

(

SELECT max(pm.n\_sum) -- максимальный платеж

FROM payment pm

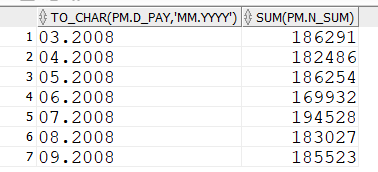
)

and ROWNUM = 1

)

GROUP BY to\_char(pm.d\_pay,'MM.yyyy')

ORDER BY to\_char(pm.d\_pay,'MM.yyyy')



1. Найти адреса людей с максимальной переплатой.

SELECT cl.n\_city, cl.n\_street, cl.c\_house, cl.n\_flat -- адрес

FROM CLIENT cl

WHERE cl.n\_client =

(

SELECT cl.n\_client -- клиент с макс. прееплатой

FROM computation cm, client cl

where cl.n\_client = cm.n\_client

having sum(cm.n\_debt) =

(

SELECT min(sum(cm.n\_debt)) -- максимальная переплата

FROM computation cm, client cl

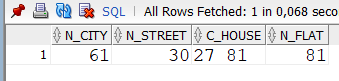
where cl.n\_client = cm.n\_client

group by cl.n\_client

)

group by cl.n\_client

)



1. Кем и когда был выполнен самый большой платеж?

SELECT cl.c\_first\_name||' '||cl.c\_last\_name, pm.d\_pay, pm.n\_sum

FROM client cl, payment pm

WHERE cl.n\_client = pm.n\_client

and pm.n\_sum = (SELECT max(pm.n\_sum) FROM payment pm)



1. Из всего периода времени найти месяц/год с максимальной оплатой по всем абонентам и вывести кто больше всех в этом месяце платил.

SELECT \*

FROM (SELECT pm.n\_client, sum(pm.n\_sum), row\_number() over (order by sum(pm.n\_sum) desc) rn -- клиент оплативший больше всех за этот период

FROM PAYMENT pm

WHERE to\_char(pm.d\_pay,'MM.yyyy') =

(

SELECT to\_char(pm.d\_pay,'MM.yyyy') -- месяц и год макс.опалаты

FROM payment pm

HAVING sum(pm.n\_sum) =

(

SELECT max(sum(pm.n\_sum)) -- максимальная оплата по всем абонентам

FROM payment pm

GROUP BY to\_char(pm.d\_pay,'MM.yyyy')

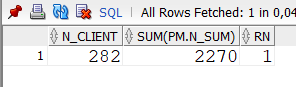
)

GROUP BY to\_char(pm.d\_pay,'MM.yyyy')

)

GROUP BY pm.n\_client)

where rn = 1



1. \*Найти Адреса 10 самых крупных потребителей холодной воды

SELECT cl.n\_client, cl.n\_city, cl.n\_street, cl.c\_house, cl.n\_flat

FROM client cl

WHERE cl.n\_client in

(SELECT cold.n\_client

FROM (

SELECT cl.n\_client, sum(cm.n\_info\_cold),

row\_number() over (order by sum(cm.n\_info\_cold) DESC) rn

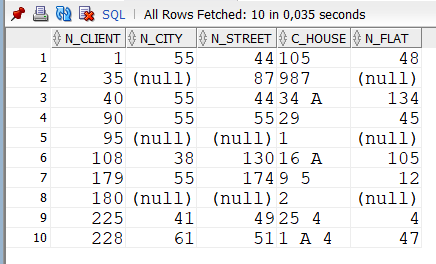
FROM client cl, computation cm

WHERE cl.n\_client = cm.n\_client

GROUP BY cl.n\_client

) cold

WHERE rn < 11)



1. \*Посчитать, сколько в среднем тратит на оплату горячей и холодной воды 10 самых крупных потребителей холодной воды.

SELECT \*

FROM (

SELECT cl.n\_client, round(avg(cm.n\_sum\_hot),2) as hot , round(avg(cm.n\_sum\_cold),2) as cold,

row\_number() over (order by sum(cm.n\_info\_cold) DESC) rn

FROM client cl, computation cm

WHERE cl.n\_client = cm.n\_client

GROUP BY cl.n\_client

)

WHERE rn < 11

