**Министерство науки и высшего образования Российской Федерации** федеральное государственное автономное образовательное учреждение высшего образования

# «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

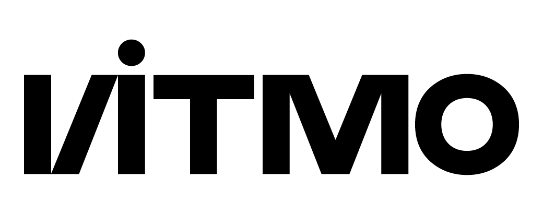
**Отчет**

по Лабораторной Работе № 4 по дисциплине **«Базы Данных**»

Автор: Белов Г.П

Факультет: ИКТ Группа: К3239

Преподаватель: Говорова Марина Михайловна



Санкт-Петербург, 2023

# Содержание работы

**Цель работы:**

Овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

# Практическое задание:

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и просмотреть историю запросов.
4. Создать простой и составной индексы для двух произвольных запросов и сравнить время выполнения запросов без индексов и с индексами. Для получения плана запроса использовать команду EXPLAIN.

# Вариант 19. БД «Банк»

# Выполнение работы:

**1. Запросы к базе данных**

1. Найти вкладчика, имеющего на текущий день несколько вкладов.

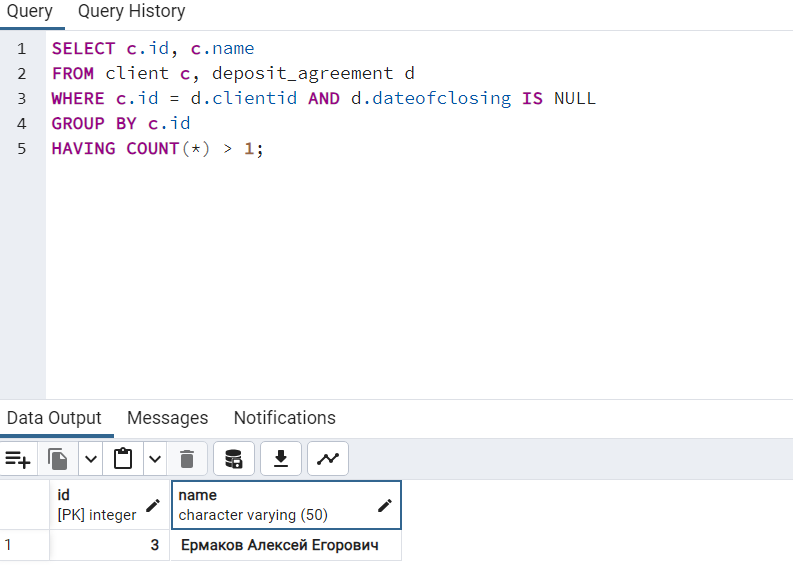
SELECT c.id, c.name

FROM client c, deposit\_agreement d

WHERE c.id = d.clientid AND d.dateofclosing IS NULL

GROUP BY c.id

HAVING COUNT(\*) > 1;

\*

1. Найти вкладчика, имеющего вклады во всех видах валюты на текущий день.

SELECT c.name, d.clientid

FROM client c, deposit\_agreement d

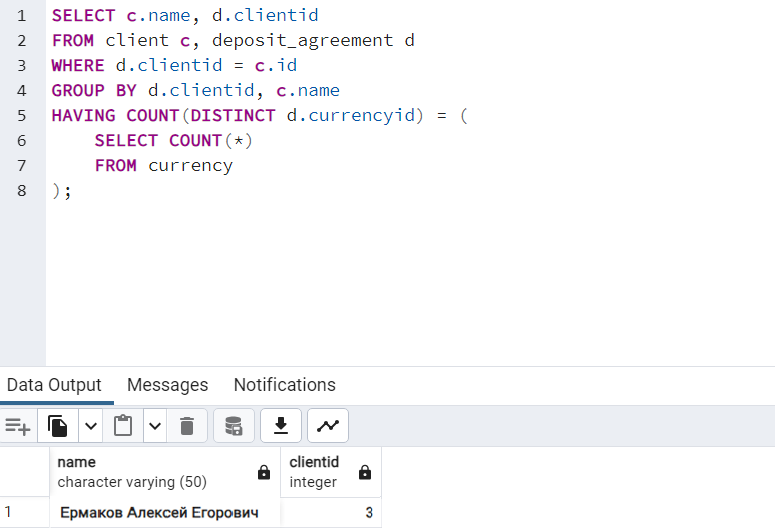
WHERE d.clientid = c.id

GROUP BY d.clientid, c.name

HAVING COUNT(DISTINCT d.currencyid) = (

SELECT COUNT(\*)

FROM currency

);

Вывести данные вкладчика, имеющего максимальный вклад в английский фунтах.

SELECT

c.id,

c.name

FROM

client c, deposit\_agreement d

WHERE

c.id = d.clientid

AND d.currencyid = (

SELECT

id

FROM

currency

WHERE

name = 'Фунт'

)

AND d.initialamount = (

SELECT

MAX(d2.initialamount)

FROM

deposit\_agreement d2

WHERE

d2.currencyid = (

SELECT

id

FROM

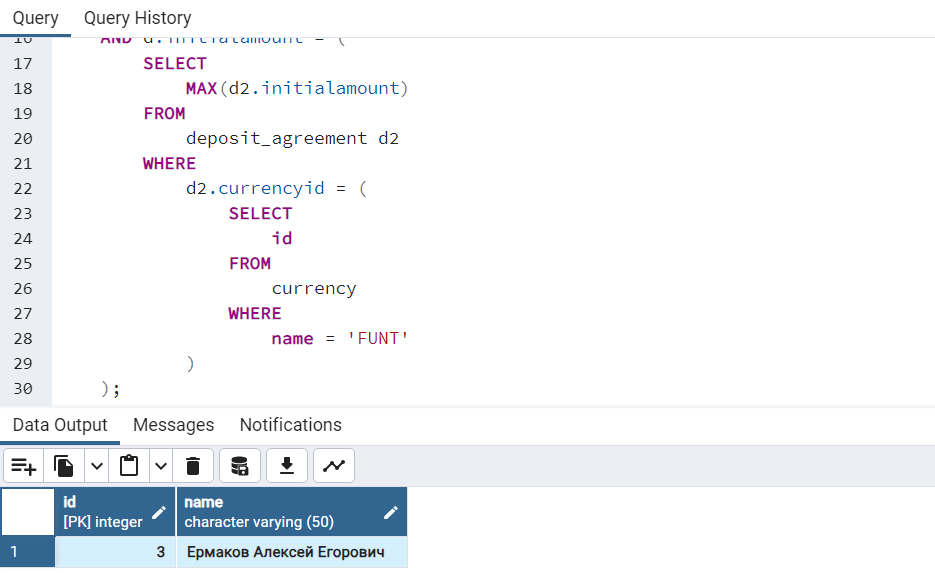
currency

WHERE

name = 'Фунт'

)

);



1. Какой из вкладов пользовался наибольшей популярностью за истекший год.

SELECT

d.name,

da.depositid,

COUNT(da.\*) as depositcount

FROM

deposit\_agreement da,

deposit d

WHERE

d.id = da.depositid

AND da.dateofopening >= CURRENT\_DATE - INTERVAL '1 year'

GROUP BY

da.depositid,

d.name

HAVING

COUNT(\*) = (

SELECT

MAX(cnt)

FROM

(

SELECT

COUNT(\*) as cnt

FROM

deposit\_agreement

WHERE

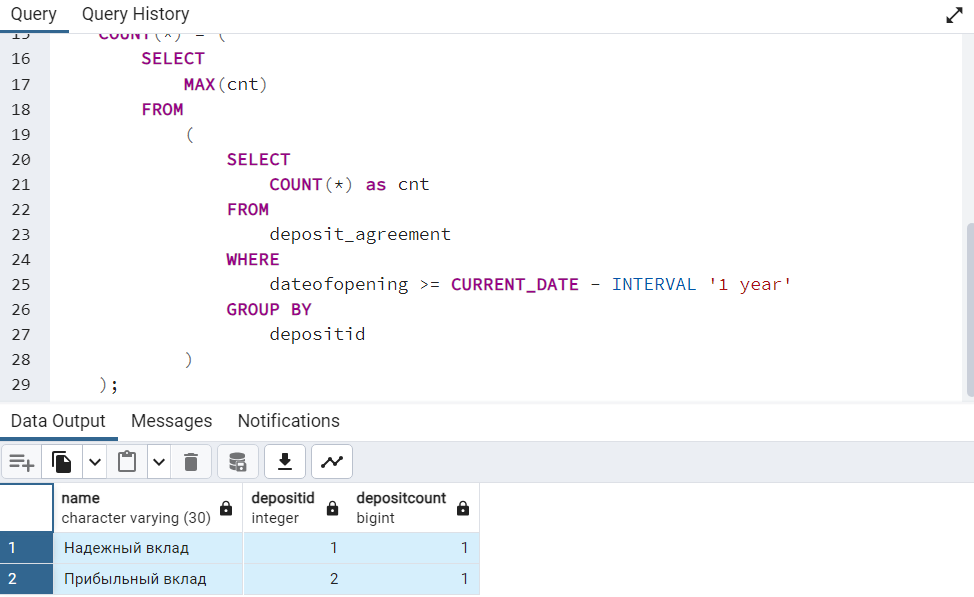
dateofopening >= CURRENT\_DATE - INTERVAL '1 year'

GROUP BY

depositid

)

)



1. Кто из сотрудников заключил максимальное число договоров по кредитам за последний месяц.

SELECT

e.name,

ca.employeeid,

COUNT(ca.\*) AS creditcount

FROM

credit\_agreement ca,

employee e

WHERE

ca.employeeid = e.id

AND ca.dateofopening >= CURRENT\_DATE - INTERVAL '1 month'

GROUP BY

ca.employeeid,

e.name

HAVING

COUNT(\*) = (

SELECT

MAX(cnt)

FROM

(

SELECT

COUNT(\*) as cnt

FROM

credit\_agreement

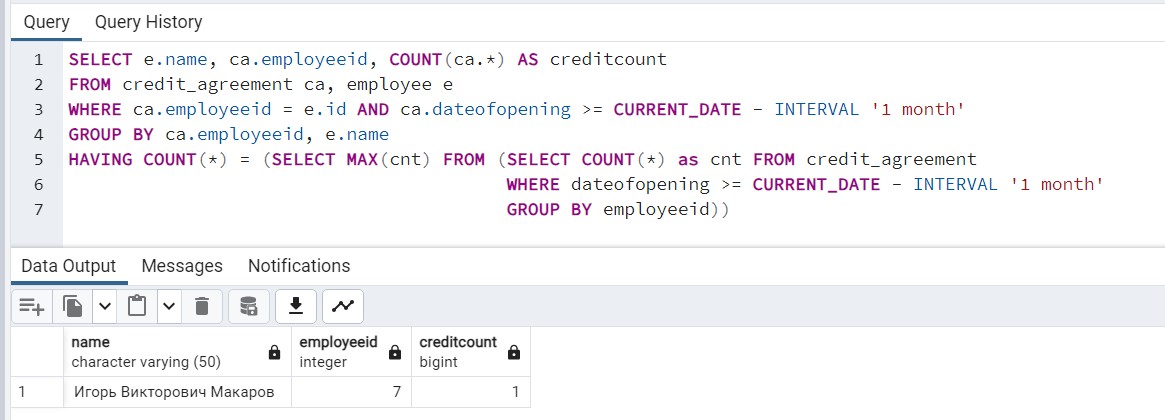
WHERE

dateofopening >= CURRENT\_DATE - INTERVAL '1 month'

GROUP BY

employeeid

)



# 2. Представления

1. Содержащее сведения обо всех сотрудниках банка и заключенных ими договорах по кредитам за прошедший месяц.

CREATE VIEW employeeandcredits AS

SELECT

e.id,

e.name,

e.passportid,

e.phonenumber,

e.email,

e.address,

e.dateofbirth,

e.salary,

e.categoryid,

ca.id as creditagreementid,

ca.currencyid,

ca.creditid,

ca.clientid,

ca.dayofaccrual,

ca.plannedclosingdate,

ca.interestamount,

ca.currentdebt,

ca.initialamount,

ca.dateofopening,

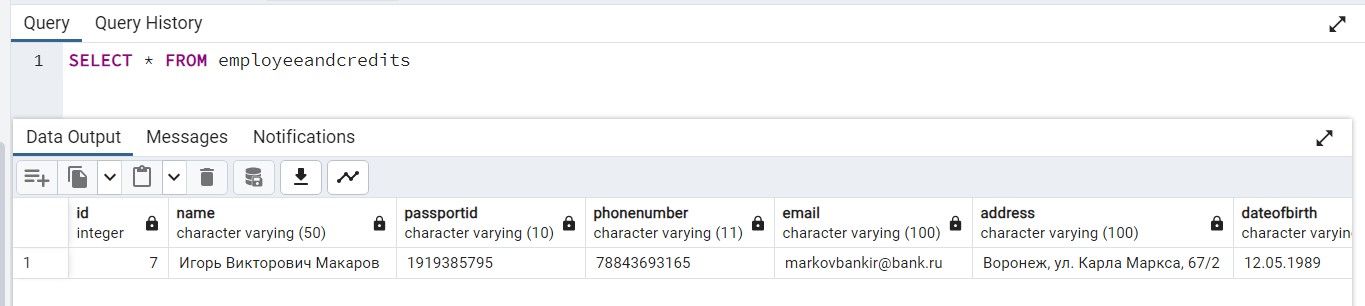
ca.dateofclosing

FROM credit\_agreement ca, employee e

WHERE

ca.dateofopening >= (CURRENT\_DATE - INTERVAL '1 month') AND

ca.employeeid = e.id;



1. Найти клиентов банка, имеющих задолженности по кредитам.

CREATE VIEW clientdebt AS

SELECT cl.\*

FROM credit\_agreement ca, client cl, credit\_payment\_schedule cps WHERE ca.id = cps.creditagreementid

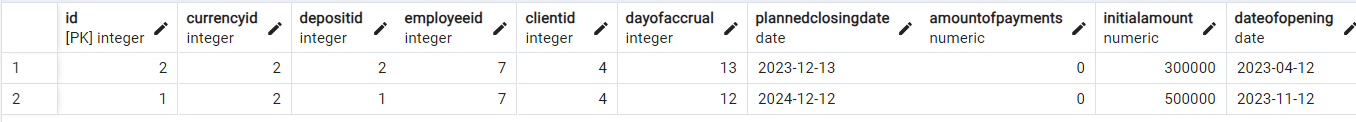
AND ca.clientid = cl.id

AND cps.plannedpaymentdate < CURRENT\_DATE AND cps.actualpaymentdate IS NULL

# Запросы на модификацию данных

1. INSERT - Создать договор о кредите на 670000 руб. на клиента с номером паспорта 2809346819.

INSERT INTO credit\_agreement(currencyid, creditid, clientid, employeeid, dayofaccrual, plannedclosingdate, interestamount, currentdebt, initialamount, dateofopening, interestrate)

VALUES((SELECT id FROM currency WHERE name = 'rub'), 2, (SELECT id FROM client WHERE passportid = '2809346819'), 7, EXTRACT(DAY FROM CURRENT\_DATE), CURRENT\_DATE + INTERVAL '1 YEAR', 0, 0, 670000, CURRENT\_DATE, 7.8) 

1. UPDATE - Обновить сумму начисленных процентов у всех договоров о вкладе

UPDATE deposit\_agreement da

SET amountofpayments = (SELECT COALESCE(SUM(paymentamount), 0) FROM deposit\_payment\_schedule dps

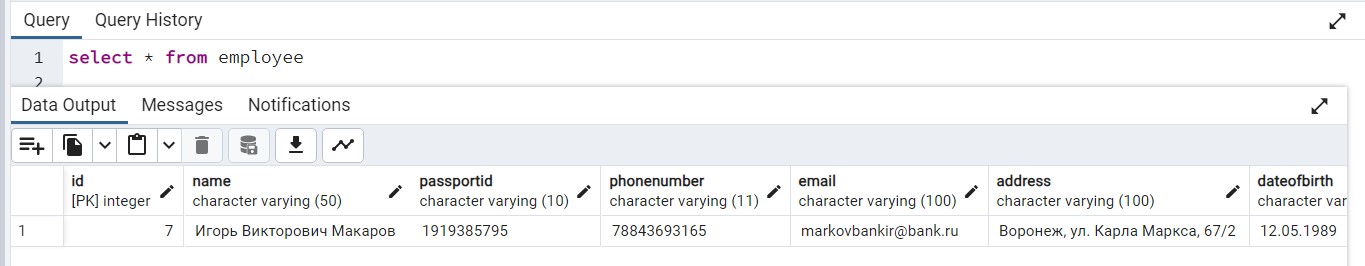
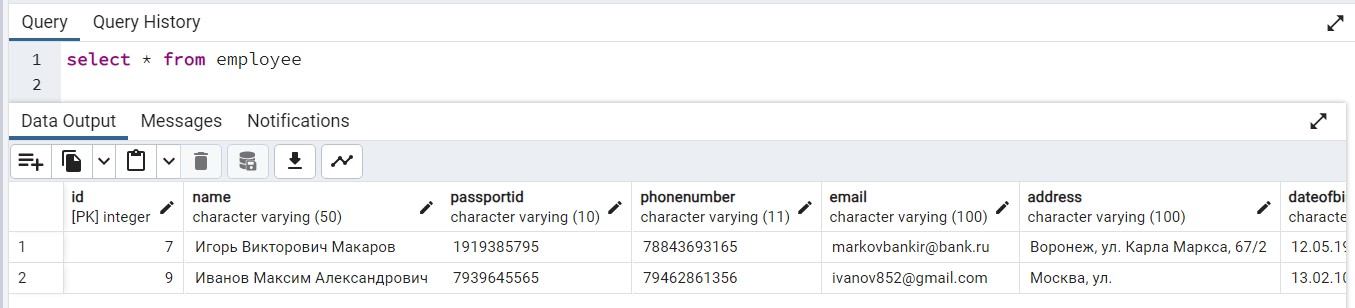
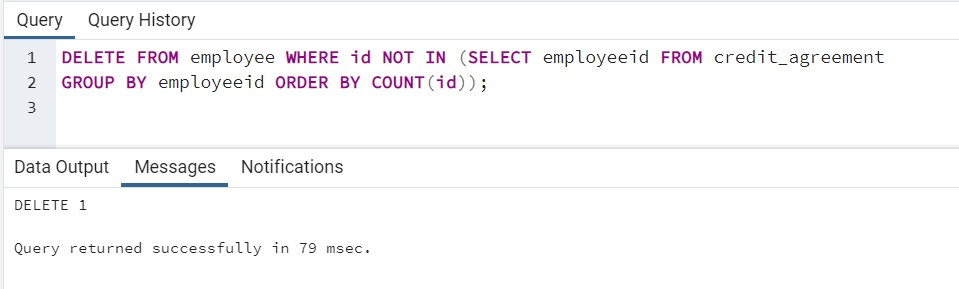
WHERE dps.depositagreementid = da.id AND dps.actualpaymentdate IS NOT NULL)



1. DELETE - Удалить сотрудников, которые не заключили ни одного договора о кредите

DELETE FROM employee WHERE id NOT IN (SELECT employeeid FROM credit\_agreement

GROUP BY employeeid ORDER BY COUNT(id));



# Создание индексов

Найти номера, имена, номера телефона и текущие долги клиентов по кредиту:

SELECT ca.clientid, cl.name, cl.phonenumber, SUM(dps.creditpayment + dps.interestpayment) as payment

FROM client cl, credit\_agreement ca, credit\_payment\_schedule dps WHERE cl.id = ca.clientid AND dps.creditagreementid = ca.id

AND dps.actualpaymentdate IS NULL AND dps.plannedpaymentdate < CURRENT\_DATE GROUP BY(ca.clientid, cl.name, cl.phonenumber, dps.creditpayment + dps.interestpayment)

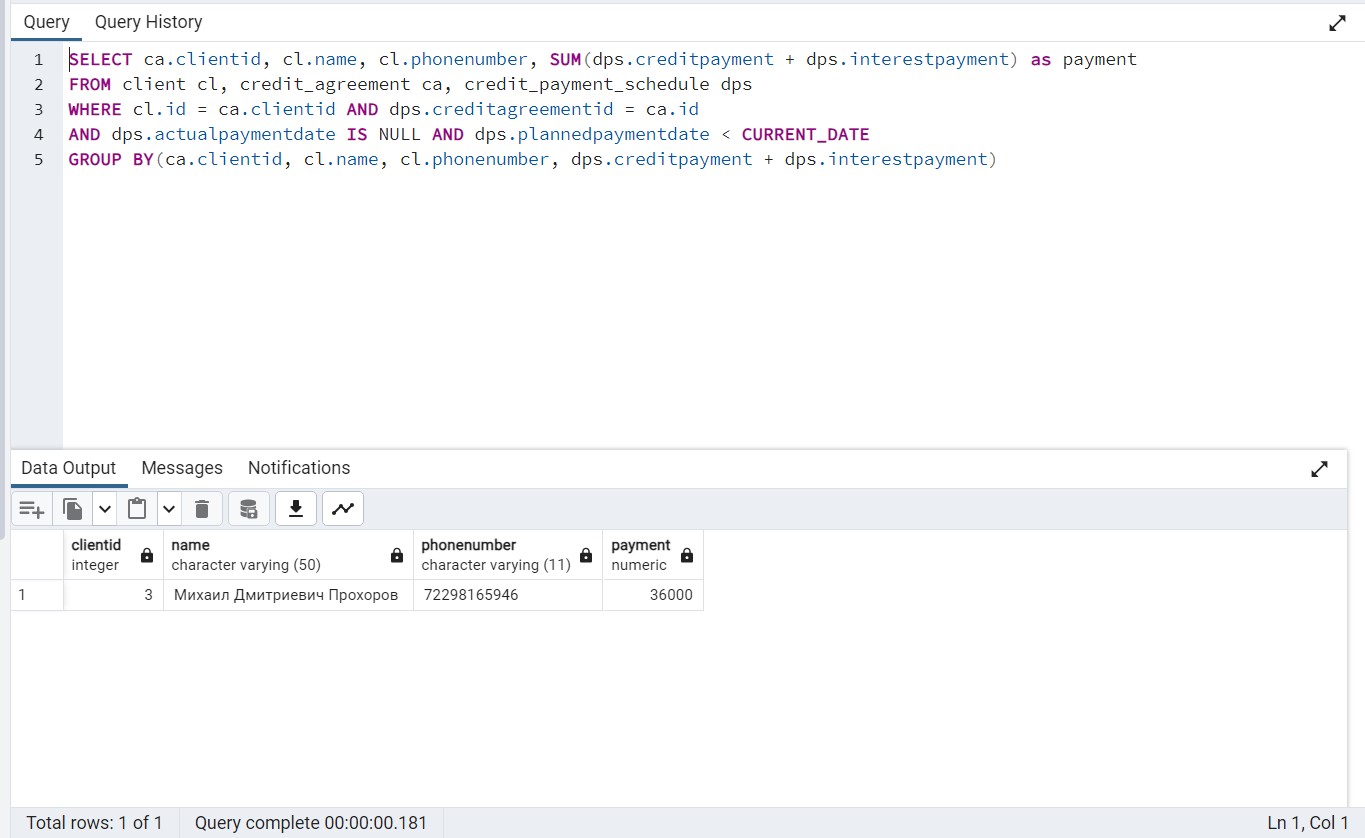
Создание индексов:

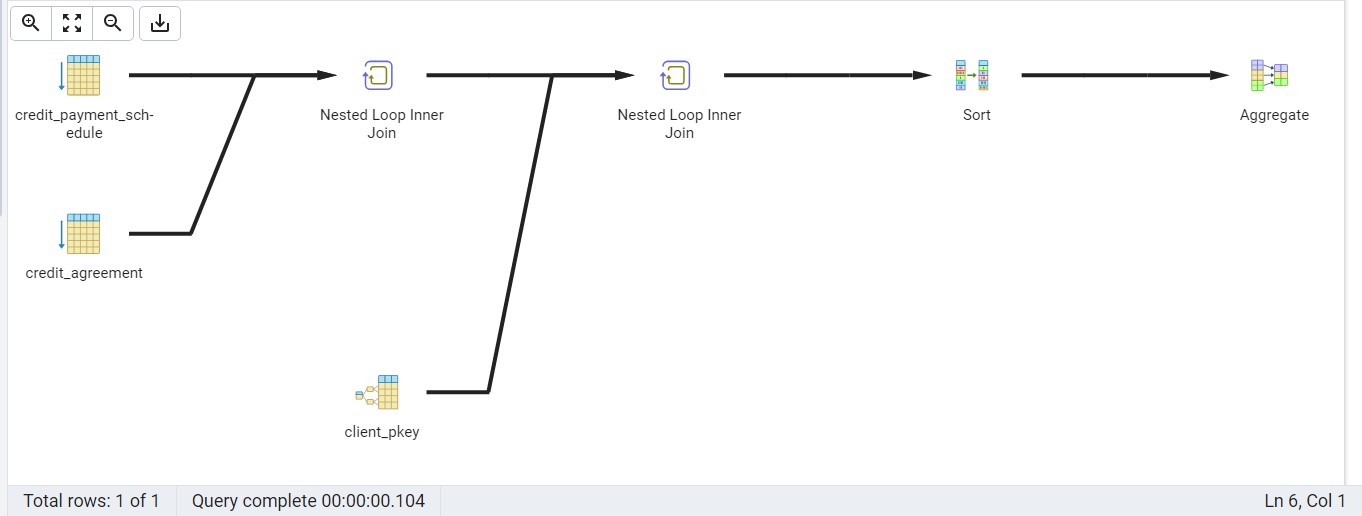
CREATE INDEX idx\_client\_id ON credit\_agreement (clientid);

CREATE INDEX idx\_payment\_dates ON credit\_payment\_schedule (plannedpaymentdate, actualpaymentdate);

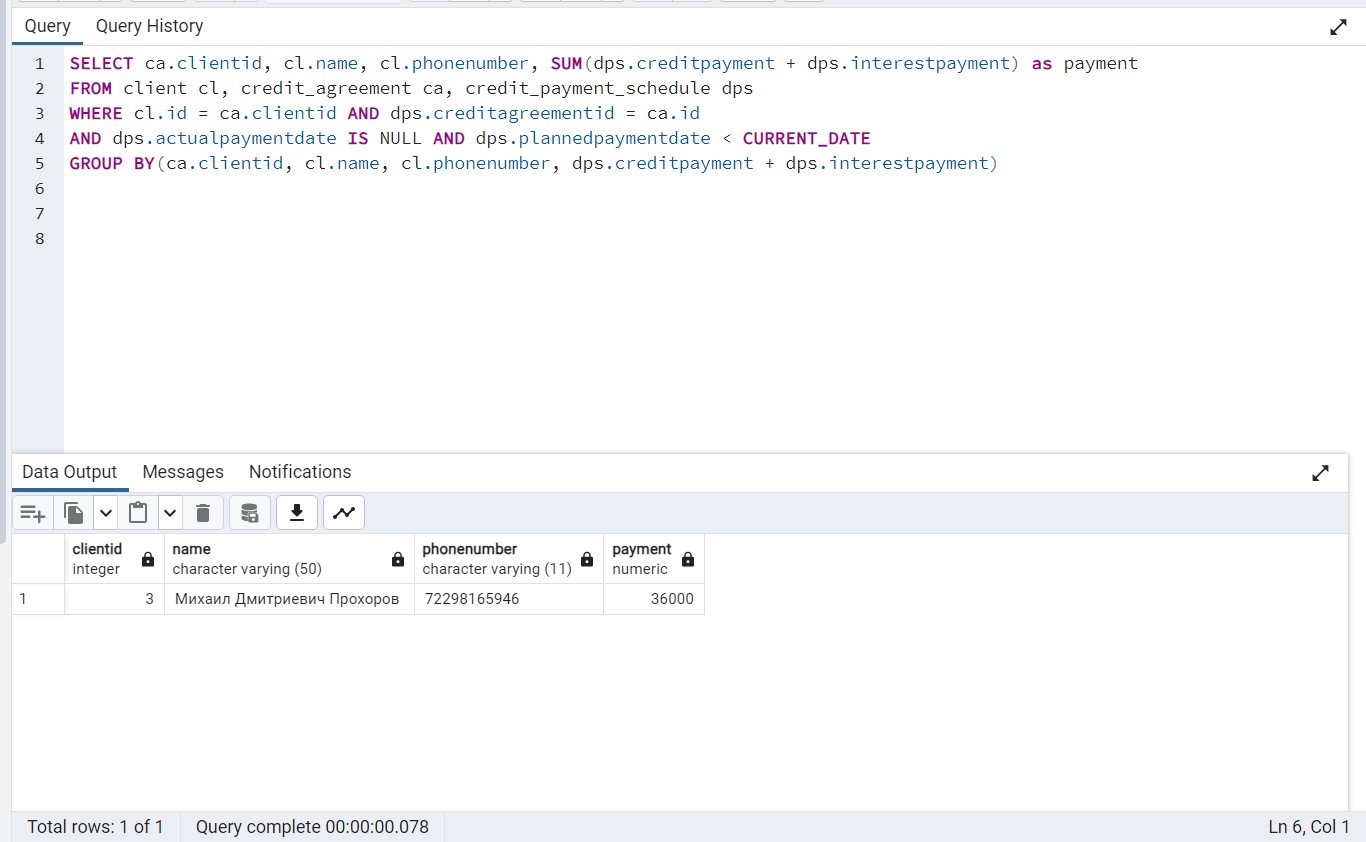
CREATE INDEX idx\_payment\_credit\_agreement\_id ON credit\_payment\_schedule (creditagreementid)

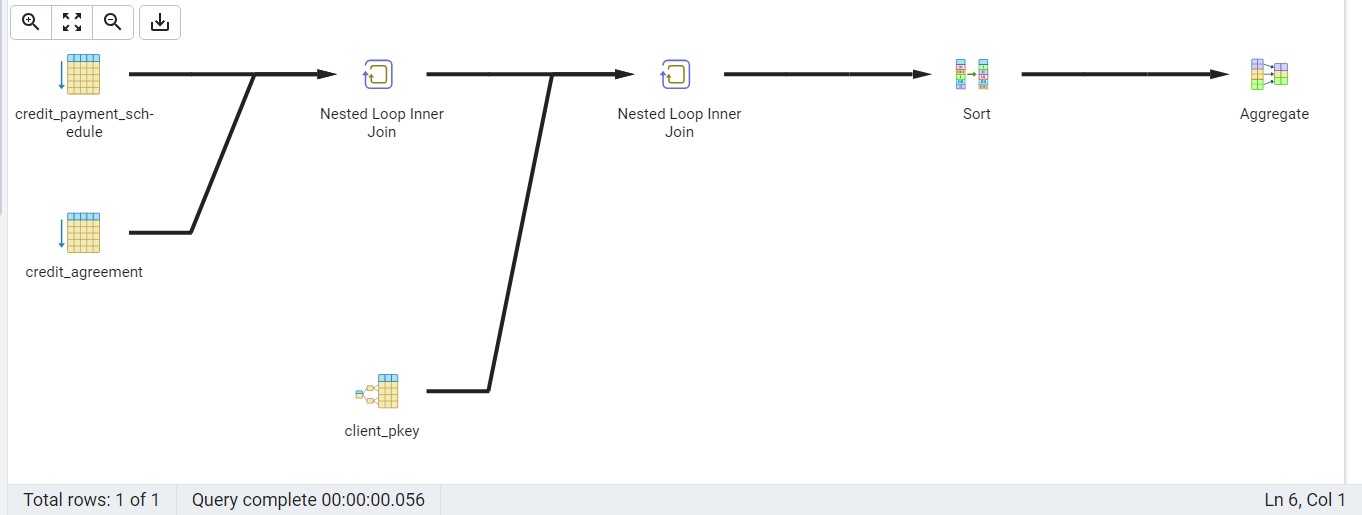
Без индекса:





С индексом:





Удаление индексов:

DROP INDEX idx\_client\_id; DROP INDEX idx\_payment\_dates;

DROP INDEX idx\_payment\_credit\_agreement\_id

# Вывод

В рамках выполнения лабораторной работы были изучены возможности написания различных запросов в PostgreSQL. В частности, мы изучили запросы на выборку данных, которые включали в себя группировку, фильтрацию, агрегацию, сортировку и присоединение. Также мы написали запросы на выбор данных из таблицы и создание представлений в соответствии с индивидуальным заданием.

Для улучшения автоматизации были использованы подзапросы для модификации данных. Это позволило сделать запросы более эффективными и быстрыми.

Для оптимизации производительности были созданы простые и составные индексы для таблиц с данными. После создания индексов мы провели сравнение скорости выполнения запросов до и после создания индексов. Результаты показали значительный прирост скорости выполнения запросов на чтение после использования индексов.

Таким образом, в результате выполнения лабораторной работы мы получили ценный опыт в написании запросов в PostgreSQL и оптимизации производительности баз данных.