

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4
«Запросы на выборку и модификацию данных. Представления.
Работа с индексами»
по дисциплине «Проектирование и реализация баз данных»

Обучающийся Недиков Михаил Олегович

Факультет прикладной информатики

Группа K3239

Направление подготовки 09.03.03 Прикладная информатика

Образовательная программа Мобильные и сетевые технологии
2025

Преподаватель Говорова Марина Михайловна

Санкт-Петербург
2025/2026

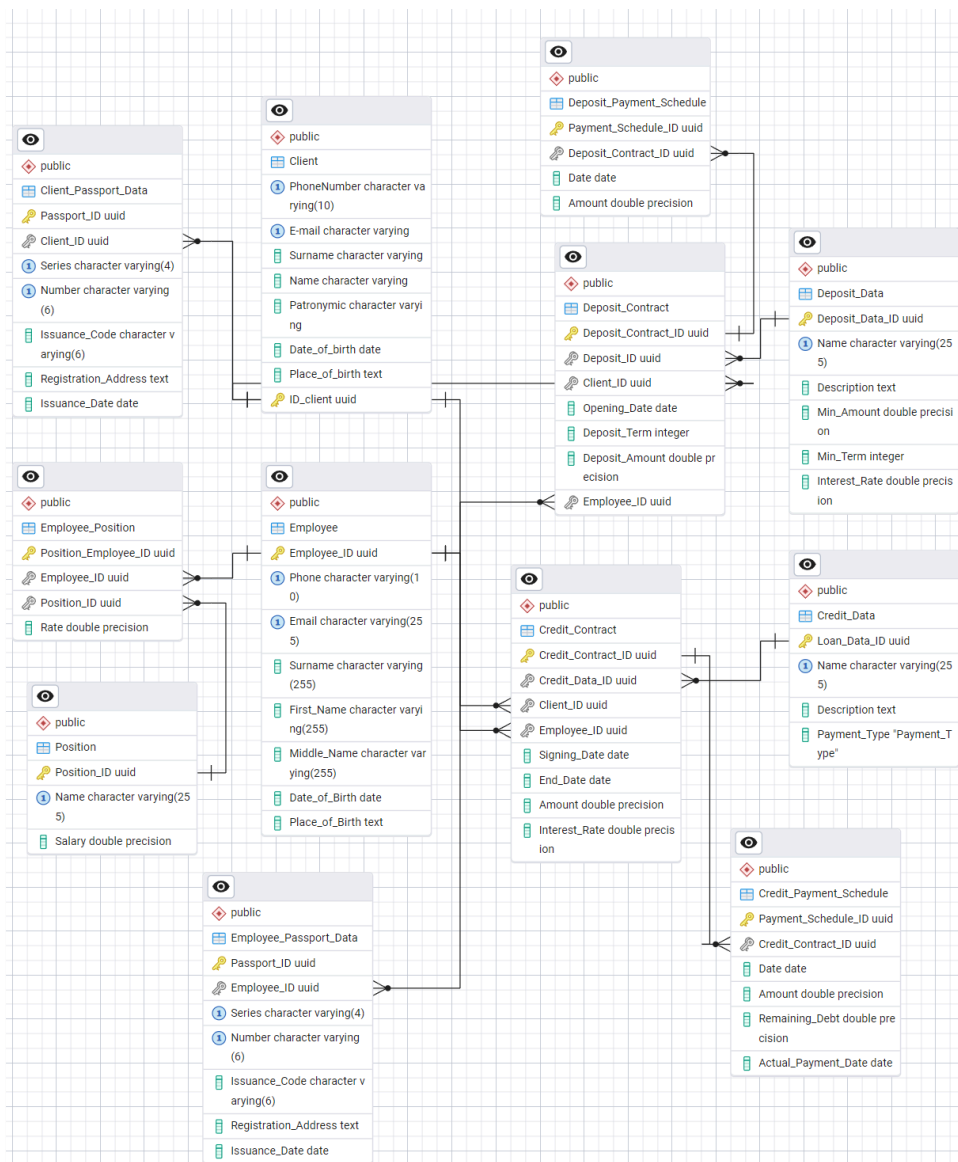
1. Цель работы

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

2. Практическое задание

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию лабораторной работы №2, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и посмотреть 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.

3. Схема базы данных (ЛР 3)



4. Выполнение

4.1 Запросы к базе данных

Запрос 1

Найти вкладчика, имеющего на текущий день несколько вкладов.

```
SELECT c."Surname", c."Name", COUNT(*) AS deposit_cnt
FROM "Client" c
JOIN "Deposit_Contract" dc ON dc."Client_ID" = c."ID_client"
```

```

WHERE dc."Opening_Date" + (dc."Deposit_Term" || ' month')::interval >=
current_date
GROUP BY c."ID_client"
HAVING COUNT(*) > 1;

```

Query Query History

```

1 SELECT c."Surname", c."Name", COUNT(*) AS deposit_cnt
2 FROM "Client" c
3 JOIN "Deposit_Contract" dc ON dc."Client_ID" = c."ID_client"
4 WHERE dc."Opening_Date" + (dc."Deposit_Term" || ' month')::interval >= current_date
5 GROUP BY c."ID_client"
6 HAVING COUNT(*) > 1;
7

```

Data Output Messages Notifications

	Surname character varying	Name character varying	deposit_cnt bigint
1	Петров	Сергей	2
2	Смирнова	Анна	2

Запрос 2

Найти вкладчика, имеющего вклады в нескольких видах валюты на текущий день.

```

SELECT c."Surname", c."Name", array_agg(DISTINCT cu."Name") AS
currencies
FROM "Client" c
JOIN "Deposit_Contract" dc ON dc."Client_ID" = c."ID_client"
JOIN "Currency" cu ON cu."Currency_ID" = dc."Currency_ID"
WHERE dc."Opening_Date" + (dc."Deposit_Term" || ' month')::interval >=

```

HAVING COUNT(DISTINCT cu."Currency_ID") > 1;

LIMIT 1;

Query Query History

1 SELECT c.*, dc."Deposit_Amount"

2 FROM "Deposit_Contract" dc

3 JOIN "Client" c ON c."ID_client" = dc."Client_ID"

4 JOIN "Currency" cu ON cu."Currency_ID" = dc."Currency_ID"

5 WHERE cu."Name" = 'GBP'

6 ORDER BY dc."Deposit_Amount" DESC

7 LIMIT 1;

Data Output Messages Notifications

SQL

	PhoneNumber character varying (10)	E-mail character varying	Surname character varying	Name character varying	Patronymic character varying	Date_of_birth date	Place_of_birth text	ID_client uuid	Deposit_Amount double precision
1	9102222222	smirnova@mail.ru	Смирнова	Анна	Викторовна	1993-05-14	Воронеж	bbbbbbb-bbbb-bbbb-bbbbbbbbb	500000

Запрос 4
Какой из вкладов пользовался наибольшей популярностью за истекший календарный год.

```
SELECT dd."Name", COUNT(*) AS opened_cnt
FROM "Deposit_Contract" dc
JOIN "Deposit_Data" dd ON dd."Deposit_Data_ID" = dc."Deposit_ID"
WHERE dc."Opening_Date" BETWEEN '2024-01-01' AND '2024-12-31'
GROUP BY dd."Name"
ORDER BY opened_cnt DESC
LIMIT 1;
```

Query
Query History

```

1 SELECT dd."Name", COUNT(*) AS opened_cnt
2 FROM "Deposit_Contract" dc
3 JOIN "Deposit_Data" dd ON dd."Deposit_Data_ID" = dc."Deposit_ID"
4 WHERE dc."Opening_Date" BETWEEN '2024-01-01' AND '2024-12-31'
5 GROUP BY dd."Name"
6 ORDER BY opened_cnt DESC
7 LIMIT 1;

```

Data Output
Messages
Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	Name character varying (255)	opened_cnt bigint
1	Сберегательный	2

Запрос 5

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

```

SELECT e."Surname", e."First_Name", COUNT(*) AS credit_cnt
FROM "Credit_Contract" cc
JOIN "Employee" e ON e."Employee_ID" = cc."Employee_ID"
WHERE cc."Signing_Date" >= date_trunc('month', current_date) - interval
'1 month'
AND cc."Signing_Date" < date_trunc('month', current_date)

```

```
GROUP BY e."Employee_ID"  
ORDER BY credit_cnt DESC  
LIMIT 1;
```

Query Query History

```
1 SELECT e."Surname", e."First_Name", COUNT(*) AS credit_cnt  
2 FROM "Credit_Contract" cc  
3 JOIN "Employee" e ON e."Employee_ID" = cc."Employee_ID"  
4 WHERE cc."Signing_Date" >= date_trunc('month', current_date) - interval '1 month'  
5 AND cc."Signing_Date" < date_trunc('month', current_date)  
6 GROUP BY e."Employee_ID"  
7 ORDER BY credit_cnt DESC  
8 LIMIT 1;
```

Data Output Messages Notifications

	Surname character varying (255)	First_Name character varying (255)	credit_cnt bigint
1	Петрова	Елена	1

Запрос 6

Вывести список вкладчиков, у которых срок вклада истекает завтра и суммы начислений, которые могут быть ими востребованы.

```
SELECT c."Surname", c."Name",  
       dc."Deposit_Amount" + dc."Accrued_Interest" AS amount_available  
FROM "Deposit_Contract" dc  
JOIN "Client" c ON c."ID_client" = dc."Client_ID"  
WHERE (dc."Opening_Date" + (dc."Deposit_Term" || '  
month')::interval)::date = current_date + 1;
```


Query
Query History

```

1 SELECT c."Surname", c."Name",
2         dc."Deposit_Amount" + dc."Accrued_Interest" AS amount_available
3 FROM "Deposit_Contract" dc
4 JOIN "Client" c ON c."ID_client" = dc."Client_ID"
5 WHERE (dc."Opening_Date" + (dc."Deposit_Term" || ' month')::interval)::date = current_date + 1;

```

Data Output
Messages
Notifications

+

📄

▼

📋

▼

🗑️

📤

⬇️

📶

SQL

	Surname character varying	Name character varying	amount_available double precision
1	Королёв	Максим	80000

Запрос 7

Вывести список сотрудников, заключивших договоры по вкладам на максимальную сумму за последний месяц.

```

WITH latest_month AS (
    SELECT *
    FROM "Deposit_Contract"
    WHERE "Opening_Date" >= date_trunc('month', current_date) - interval '1 month'
    AND "Opening_Date" < date_trunc('month', current_date)
)
SELECT e."Surname", e."First_Name",
       SUM(lm."Deposit_Amount") AS total_amount
FROM latest_month lm
JOIN "Employee" e ON e."Employee_ID" = lm."Employee_ID"
GROUP BY e."Employee_ID"
ORDER BY total_amount DESC;

```

Query
Query History

```

1 WITH latest_month AS (
2     SELECT *
3     FROM "Deposit_Contract"
4     WHERE "Opening_Date" >= date_trunc('month', current_date) - interval '1 month'
5           AND "Opening_Date" < date_trunc('month', current_date)
6 )
7 SELECT e."Surname", e."First_Name",
8        SUM(lm."Deposit_Amount") AS total_amount
9 FROM latest_month lm
10 JOIN "Employee" e ON e."Employee_ID" = lm."Employee_ID"
11 GROUP BY e."Employee_ID"
12 ORDER BY total_amount DESC;

```

Data Output
Messages
Notifications

+

📄

▼

🗑️

🔄

⬇️

📈

SQL

	Surname character varying (255) 🔒	First_Name character varying (255) 🔒	total_amount double precision 🔒
1	Иванов	Иван	300000

4.2 Представления

Представление 1

Сведения обо всех сотрудниках банка и их кредитных договорах за прошедший месяц.

```
CREATE OR REPLACE VIEW vw_employee_credit_last_month AS
```

```

SELECT e."Employee_ID",
       e."Surname",
       e."First_Name",
       cc."Credit_Contract_ID",
       cc."Client_ID",

```

```

cc."Signing_Date",

cc."Amount",

cc."Interest_Rate"

FROM "Employee" e

JOIN "Credit_Contract" cc          -- INNER JOIN

    ON cc."Employee_ID" = e."Employee_ID"

    AND cc."Signing_Date" >= date_trunc('month', current_date) - interval
'1 month'

    AND cc."Signing_Date" < date_trunc('month', current_date);

SELECT * FROM vw_employee_credit_last_month;

```

Query

Query History

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

```
CREATE OR REPLACE VIEW vw_employee_credit_last_month AS
SELECT e."Employee_ID",
       e."Surname",
       e."First_Name",
       cc."Credit_Contract_ID",
       cc."Client_ID",
       cc."Signing_Date",
       cc."Amount",
       cc."Interest_Rate"
FROM   "Employee" e
JOIN    "Credit_Contract" cc          -- INNER JOIN
ON      cc."Employee_ID" = e."Employee_ID"
AND      cc."Signing_Date" >= date_trunc('month', current_date) - interval '1 month'
AND      cc."Signing_Date" < date_trunc('month', current_date);

SELECT * FROM vw_employee_credit_last_month;
```

Data Output

Messages

Notifications

SQL

Showing rows

Employee_ID	Surname	First_Name	Credit_Contract_ID	Client_ID	Signing_Date	Amount	Interest_Rate
uid	character varying (255)	character varying (255)	uuid	uuid	date	double precision	double precision
1	22222222-2222-2222-2222-222222222222	Петрова	Елена	5e40c14b-89f8-4bae-b68b-99597bdf3309	bbbbbbbb-bbbb-bbbb-bbbb-bbbbbbbbbbbb	2025-04-05	700000

Представление 2

Клиенты банка, имеющие задолженности по кредитам.

```
CREATE OR REPLACE VIEW vw_clients_with_overdue AS
```

```
SELECT DISTINCT c.*
```

```

FROM "Credit_Payment_Schedule" ps

JOIN "Credit_Contract" cc ON cc."Credit_Contract_ID" =
ps."Credit_Contract_ID"

JOIN "Client" c      ON c."ID_client" = cc."Client_ID"

WHERE ps."Is_Overdue";

SELECT * FROM vw_clients_with_overdue;

```

Query Query History

```

1 CREATE OR REPLACE VIEW vw_clients_with_overdue AS
2 SELECT DISTINCT c.*
3 FROM "Credit_Payment_Schedule" ps
4 JOIN "Credit_Contract" cc ON cc."Credit_Contract_ID" = ps."Credit_Contract_ID"
5 JOIN "Client" c      ON c."ID_client" = cc."Client_ID"
6 WHERE ps."Is_Overdue";
7
8
9 SELECT * FROM vw_clients_with_overdue;

```

Data Output Messages Notifications

	PhoneNumber character varying (10)	E-mail character varying	Surname character varying	Name character varying	Patronymic character varying	Date_of_birth date	Place_of_birth text	ID_client uuid
1	9103333333	korolev@mail.ru	Королёв	Максим	Игоревич	1988-12-03	Самара	cccccccc-cccc-cccc-cccc-cccccccccccc
2	9102222222	smirnova@mail.ru	Смирнова	Анна	Викторовна	1993-05-14	Воронеж	bbbbbbbb-bbbb-bbbb-bbbb-bbbbbbbbbbbb

4.3 Модификация данных

Составьте и выполните по одному сложному запросу INSERT, UPDATE и DELETE с подзапросами. Ниже placeholders под формулировки, SQL и скриншоты.

INSERT-запрос

Формулировка: Добавить новый вклад 100 000 RUB тому клиенту, который уже имеет самую большую суммарную сумму вкладов в рублях. Вклад открывает сотрудник Иванов И.И., срок — 12 месяцев, тип вклада — первый в списке таблицы Deposit_Data

SQL-код:

```
INSERT INTO "Deposit_Contract" (  
    "Deposit_Contract_ID",  
    "Deposit_ID",  
    "Client_ID",  
    "Opening_Date",  
    "Deposit_Term",  
    "Deposit_Amount",  
    "Employee_ID",  
    "Currency_ID")  
SELECT gen_random_uuid(),                -- PK  
    (SELECT "Deposit_Data_ID" FROM "Deposit_Data" ORDER BY 1  
LIMIT 1),  
    best_client."ID_client",  
    CURRENT_DATE,  
    12,  
    100000,  
    '11111111-1111-1111-1111-111111111111',    -- Иванов И.И.  
    (SELECT "Currency_ID" FROM "Currency" WHERE "Name" =  
'RUB')  
FROM (  
    SELECT c."ID_client"  
    FROM "Client" c  
    JOIN "Deposit_Contract" dc ON dc."Client_ID" = c."ID_client"
```

JOIN "Currency" cu ON cu."Currency_ID" = dc."Currency_ID"

WHERE cu."Name" = 'RUB'

GROUP BY c."ID_client"

ORDER BY SUM(dc."Deposit_Amount") DESC

LIMIT 1

) AS best_client;

Скрин до:

Query

Query History

1

SELECT c."Surname", c."Name", SUM(dc."Deposit_Amount") AS total_rub

2

FROM "Client" c

3

JOIN "Deposit_Contract" dc ON dc."Client_ID" = c."ID_client"

4

JOIN "Currency" cu ON cu."Currency_ID" = dc."Currency_ID"

5

WHERE cu."Name" = 'RUB'

6

GROUP BY c."ID_client"

7

ORDER BY total_rub DESC;

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	Surname character varying 🔒	Name character varying 🔒	total_rub double precision 🔒
1	Иванова	Ольга	300000
2	Петров	Сергей	210000
3	Смирнова	Анна	200000
4	Королёв	Максим	80000

Скрин после:

Query Query History

```
1 SELECT c."Surname", c."Name", SUM(dc."Deposit_Amount") AS total_rub
2 FROM   "Client" c
3 JOIN   "Deposit_Contract" dc ON dc."Client_ID" = c."ID_client"
4 JOIN   "Currency" cu ON cu."Currency_ID" = dc."Currency_ID"
5 WHERE  cu."Name" = 'RUB'
6 GROUP BY c."ID_client"
7 ORDER BY total_rub DESC;
```

Data Output Messages Notifications

≡+ 📄 ▼ 📋 ▼ 🗑️ 🗄️ ⬇️ ⤴️ SQL

	Surname character varying 🔒	Name character varying 🔒	total_rub double precision 🔒
1	Иванова	Ольга	400000
2	Петров	Сергей	210000
3	Смирнова	Анна	200000
4	Королёв	Максим	80000

Query Query History

```

1 SELECT * FROM "Deposit_Contract"
2 WHERE "Opening_Date" = CURRENT_DATE
3 AND "Deposit_Amount" = 100000;

```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

Deposit_Contract_ID	Deposit_ID	Client_ID	Opening_Date	Deposit_Term	Deposit_Amount	Employee_ID	Currency_ID	Accrued_Interest
PK1 uuid	uuid	uuid	date	integer	double precision	uuid	uuid	numeric (14,2)
1	9be52db1-42b5-4560-9a9e-b2c7e4221b95	ab7c148a-7ae5-4ce4-96ea-db9b76e7d934	2025-05-22	12	100000	111111111-1111-1111-1111-111111111111	52bf0ee0-4361-4467-b466-c4d668947b7b	0.00

UPDATE-запрос

Формулировка: Повысить оклад на 10 % всем сотрудникам, чья текущая зарплата ниже среднего по их должности (должность определяется через таблицу Position).

SQL-код:

UPDATE "Employee_Position" ep

SET "Salary" = "Salary" * 1.10

WHERE "Salary" < (

SELECT AVG("Salary")

FROM "Employee_Position" sub

WHERE sub."Position_ID" = ep."Position_ID"

);

Скрин до:

Query Query History

```
1 SELECT e."Surname",
2       e."First_Name",
3       p."Name"           AS position,
4       p."Salary"         AS position_salary,
5       avg_pos.avg_salary
6 FROM   "Employee_Position" ep
7 JOIN   "Employee" e ON e."Employee_ID" = ep."Employee_ID"
8 JOIN   "Position" p ON p."Position_ID" = ep."Position_ID"
9 JOIN   (
10        SELECT AVG("Salary") AS avg_salary
11        FROM    "Position"
12      ) AS avg_pos ON TRUE -- одна строка с общим средним
13 WHERE  p."Salary" < avg_pos.avg_salary;
14
```

Data Output Messages Notifications

	Surname character varying (255)	First_Name character varying (255)	position character varying (255)	position_salary double precision	avg_salary double precision
1	Иванов	Иван	Менеджер по вкладам	70000	72000

Скрин после:

```

1 SELECT e."Surname",
2        e."First_Name",
3        p."Name"           AS position,
4        p."Salary"         AS position_salary,
5        avg_pos.avg_salary
6 FROM   "Employee_Position" ep
7 JOIN   "Employee" e ON e."Employee_ID" = ep."Employee_ID"
8 JOIN   "Position" p ON p."Position_ID" = ep."Position_ID"
9 JOIN   (
10        SELECT AVG("Salary") AS avg_salary
11        FROM    "Position"
12      ) AS avg_pos ON TRUE -- одна строка с общим средним
13 WHERE  p."Salary" < avg_pos.avg_salary;
14

```

DELETE-запрос

Формулировка:

Удалить из графика платежей все строки, по которым платёж просрочен (Is Overdue = true) уже больше недели.

SQL-код:

DELETE FROM "Credit Payment Schedule"

WHERE "Payment_Schedule_ID" IN (

```
SELECT "Payment Schedule ID"
```

FROM "Credit Payment Schedule"

WHERE "Is_Overdue" = true

AND "Date" < CURRENT_DATE - INTERVAL '7 days'

$$);$$

Скрин до:

Query

Query History

1

SELECT COUNT(*) AS overdue_7d

2

FROM "Credit_Payment_Schedule"

3

WHERE "Is_Overdue" = true

4

AND "Date" < CURRENT_DATE - INTERVAL '7 days';

Data Output

Messages

Notifications

≡+

▼

▼

SQL

overdue_7d

bigint

1

1

Скрин после:

Query Query History

```
1 SELECT COUNT(*) AS remain_overdue_7d
2 FROM "Credit_Payment_Schedule"
3 WHERE "Is_Overdue" = true
4 AND "Date" < CURRENT_DATE - INTERVAL '7 days';
```

Data Output Messages Notifications

	remain_overdue_7d bigint
1	0

4.4 Индексы и планы выполнения

Провести сравнение работы запроса без индекса и с индексом.

EXPLAIN без индекса

Query

Query History

1

EXPLAIN (ANALYZE, BUFFERS, FORMAT JSON)

SELECT *

FROM "Credit_Contract"

WHERE "Client_ID" = 'bbbbbbbb-bbbb-bbbb-bbbbbbbbbbbb';

Data Output

Messages

Explain

x

Notifications

Graphical

Analysis

Statistics

#	Node	Timings		Rows			Loops
		Exclusive	Inclusive	Rows X	Actual	Plan	
1.	Seq Scan on Credit_Contract as Credit_Contract (cost=0.104 rows=1 width=104) (actual=0.026... Filter ("Client_ID" = 'bbbbbbbb-bbbb-bbbb-bbbbbbbbbbbb')::uuid) Rows Removed by Filter: 1	0.028 ms	0.028 ms	2	2	1	

EXPLAIN с индексом

DROP INDEX IF EXISTS idx_credit_contract_client_id;

В ходе лабораторной работы я:

- составил 7 сложных запросов выборки, убедившись, что они возвращают корректные результаты на тестовых данных;
- создал представления, упрощающие доступ к актуальной информации о кредитах и задолженностях клиентов;
- выполнил модификацию данных (INSERT, UPDATE, DELETE) с подзапросами, подтвердив изменения скриншотами «до/после»;
- сравнил планы выполнения запросов до и после создания индекса, на практике увидев сокращение времени и переход с Seq Scan на Index Scan;
- освоил инструменты pgAdmin 4 — генератор ERD-схем, вкладку Explain/Analyze и механизм покадровых скриншотов для отчёта.