

**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** Маракулин Андрей Андреевич  
**Факультет** прикладной информатики  
**Группа** К3241  
**Направление подготовки** 09.03.03 Прикладная информатика  
**Образовательная программа** Мобильные и сетевые технологии 2023  
**Преподаватель** Говорова Марина Михайловна

Санкт-Петербург  
2024/2025

**Цель работы:** овладеть практическими навыками создания таблиц базы данных PostgreSQL 1X, заполнения их рабочими данными, резервного копирования и восстановления БД.

**Практическое задание:**

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию лабораторной работы №2, часть 2 и 3).

**Задание 2.** Создайте запросы:

- Найти вкладчика, имеющего на текущий день несколько вкладов.
- Найти вкладчика, имеющего вклады во нескольких видах валюты на текущий день.
- Вывести данные вкладчика, имеющего максимальный вклад в английских фунтах.
- Какой из вкладов пользовался наибольшей популярностью за истекший календарный год.
- Кто из сотрудников заключил максимальное число договоров по кредитам за последний месяц.
- Вывести список вкладчиков, у которых срок вклада истекает завтра и суммы начислений, которые могут быть ими востребованы.
- Вывести список сотрудников, заключивших договоры по вкладам на максимальную сумму за последний месяц.

**Задание 3.** Создать представление:

- содержащее сведения обо всех сотрудниках банка и заключенных ими договорах по кредитам за прошедший месяц;
  - найти клиентов банка, имеющих задолженности по кредитам.
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.

**Индивидуальное задание - Вариант 19. БД «Банк»**

Описание предметной области: Система обеспечивает работу с вкладами и кредитами клиентов банка.

Клиенты банка имеют вклады и кредиты различных видов. Для вкладов и кредитов может использоваться различная валюта.

Сотрудники банка заключают договоры с клиентами. Фиксируется сотрудник, заключивший договор.

Ежемесячно начисляется процент по вкладу, и полученная сумма добавляется к сумме вклада заказчика. Вкладчик имеет право снимать проценты по вкладу или всю сумму вклада с

процентами по истечении срока вклада. При снятии денег до истечения срока вклада процент

за текущий месяц не начисляется.

Кредит выдается на определенный срок. Формируется график выплат, который получает клиент при заключении договора, в котором ежемесячно указывается сумма выплаты по вкладу

и сумма выплаты по процентам банку. Хранится информация по своевременности ежемесячных выплат.

БД должна содержать следующий минимальный набор сведений: ФИО сотрудника. Возраст сотрудника. Адрес сотрудника. № телефона сотрудник. Паспортные данные сотрудника. Должность сотрудника. Оклад сотрудника (зависит от должности). Наименование вклада. Описание вклада. Минимальный срок вклада. Минимальная сумма вклада. Процент по вкладу. Срок вклада. Процентная ставка. Код валюты. Наименование валюты. ФИО вкладчика. Адрес вкладчика. Телефон клиента. E-mail вкладчика. Паспортные данные. Номер договора. Дата вклада. Дата возврата. Сумма вклада. Сумма возврата. Данные по кредиту. Число выплаты ежемесячно (нельзя указывать 29, 30 и 31). Должность сотрудника. Количество ставок (по штатному расписанию).

Название: БД «Bank»

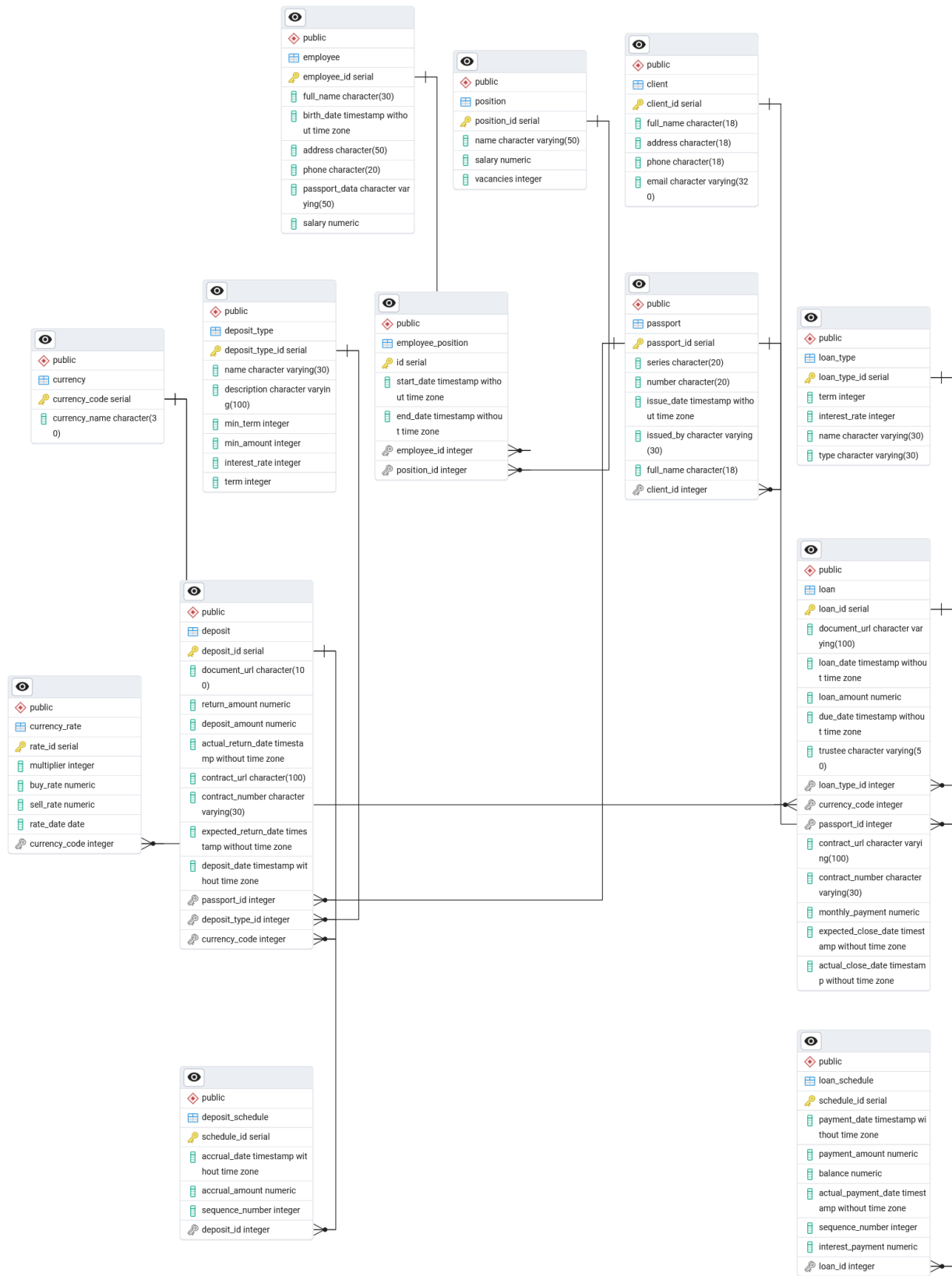


Рисунок 1 - Схема логической модели базы данных.

Выполнение:

Задание 2

- Найти вкладчика, имеющего на текущий день несколько вкладов.

QueryQuery History

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

SELECT  
c.client\_id,  
c.full\_name,  
COUNT(\*) AS cnt\_deposits  
FROM public.client AS c  
JOIN public.passport AS p  
ON p.client\_id = c.client\_id  
JOIN public.deposit AS d  
ON d.passport\_id = p.passport\_id  
WHERE  
d.deposit\_date::date <= CURRENT\_DATE  
AND d.expected\_return\_date::date >= CURRENT\_DATE  
GROUP BY c.client\_id, c.full\_name  
HAVING COUNT(\*) > 1;

Data OutputMessagesNotifications

≡+

📄

▼

📋

▼

🗑

🗄

⬇

📈

SQL

	client_id [PK] integer	full_name character (32)	cnt_deposits bigint
1	1	Иванов Иван Иванов...	2

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

Query

Query History

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

```

SELECT
  c.client_id,
  c.full_name,
  COUNT(DISTINCT d.currency_code) AS cnt_currencies
FROM public.client AS c
JOIN public.passport AS p
  ON p.client_id = c.client_id
JOIN public.deposit AS d
  ON d.passport_id = p.passport_id
WHERE
  d.deposit_date::date <= CURRENT_DATE
  AND d.expected_return_date::date >= CURRENT_DATE
GROUP BY c.client_id, c.full_name
HAVING COUNT(DISTINCT d.currency_code) > 1;

```

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	client_id [PK] integer	full_name character (32)	cnt_currencies bigint
1	1	Иванов Иван Иванович...	2

- Вывести данные вкладчика, имеющего максимальный вклад в английских фунтах.

```
Query Query History
1 WITH gbp AS (
2     SELECT currency_code
3     FROM public.currency
4     WHERE currency_name ILIKE '%Английские фунты%'
5 )
6 SELECT
7     c.client_id,
8     c.full_name,
9     c.address,
10    c.phone,
11    c.email
12 FROM public.deposit AS d
13 JOIN gbp
14     ON d.currency_code = gbp.currency_code
15 JOIN public.passport AS p
16     ON d.passport_id = p.passport_id
17 JOIN public.client AS c
18     ON d.client_id = c.client_id
```

Data Output Messages Notifications

	client_id [PK] integer	full_name character (32)	address character (32)	phone character (32)	email character varying (320)
1	2	Петров Пётр Петров...	Невский пр., 1...	+7 900000000...	petrov@example.com

- Какой из вкладов пользовался наибольшей популярностью за истекший календарный год.

Query

Query History

1

2

3

4

5

6

7

8

9

10

11

12

13

SELECT  
dt.name AS deposit\_type,  
COUNT(\*) AS cnt\_contracts  
FROM public.deposit AS d  
JOIN public.deposit\_type AS dt  
ON d.deposit\_type\_id = dt.deposit\_type\_id  
WHERE d.deposit\_date  
BETWEEN (date\_trunc('year', CURRENT\_DATE) - INTERVAL '1 year')  
AND (date\_trunc('year', CURRENT\_DATE) - INTERVAL '1 day')  
GROUP BY dt.name  
ORDER BY cnt\_contracts DESC  
LIMIT 1;

Data Output

Messages

Notifications

+

SQL

deposit\_type

character varying (30)

🔒

cnt\_contracts

bigint

🔒

1

Standard

2

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



QueryQuery History

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

SELECT  
e.employee\_id,  
e.full\_name,  
COUNT(\*) AS cnt\_loans  
FROM public.loan AS l  
JOIN public.employee\_position AS ep  
ON l.loan\_date BETWEEN ep.start\_date AND COALESCE(ep.end\_date, NOW())  
JOIN public.employee AS e  
ON ep.employee\_id = e.employee\_id  
WHERE l.loan\_date >= date\_trunc('month', CURRENT\_DATE) - INTERVAL '1 month'  
AND l.loan\_date < date\_trunc('month', CURRENT\_DATE)  
GROUP BY e.employee\_id, e.full\_name  
ORDER BY cnt\_loans DESC  
LIMIT 1;

Data OutputMessagesNotifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

S

	employee_id [PK] integer	full_name character (30)	cnt_loans bigint
1	21	Кузнецова Анна Иванов...	2

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

QueryQuery History

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

SELECT  
c.client\_id,  
c.full\_name,  
d.deposit\_id,  
d.expected\_return\_date::date AS return\_date,  
SUM(ds.accrual\_amount) AS total\_accrued  
FROM public.deposit AS d  
JOIN public.deposit\_schedule AS ds  
ON d.deposit\_id = ds.deposit\_id  
JOIN public.passport AS p  
ON d.passport\_id = p.passport\_id  
JOIN public.client AS c  
ON p.client\_id = c.client\_id  
WHERE d.expected\_return\_date::date = CURRENT\_DATE + INTERVAL '1 day'  
GROUP BY c.client\_id, c.full\_name, d.deposit\_id, d.expected\_return\_date;

Data OutputMessagesNotifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

	client_id integer	full_name character (32)	deposit_id integer	return_date date	total_accrued numeric
1	3	Сидорова Ольга Олегов...	1003	2025-05-02	180

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

QueryQuery History

```

1  WITH sums AS (
2      SELECT
3          ep.employee_id,
4          SUM(d.deposit_amount) AS total_deposit_sum
5      FROM public.deposit AS d
6      JOIN public.employee_position AS ep
7          ON d.deposit_date BETWEEN ep.start_date AND COALESCE(ep.end_date, NOW())
8      WHERE d.deposit_date >= date_trunc('month', CURRENT_DATE) - INTERVAL '1 month'
9          AND d.deposit_date < date_trunc('month', CURRENT_DATE)
10     GROUP BY ep.employee_id
11 )
12 SELECT
13     e.employee_id,
14     e.full_name,
15     s.total_deposit_sum
16 FROM sums AS s
17 JOIN public.employee AS e
18     ON s.employee_id = e.employee_id
19 WHERE s.total_deposit_sum = (
20     SELECT MAX(total_deposit_sum) FROM sums
21 );
22

```

Data OutputMessagesNotifications

+

SQL

Showing

	employee_id [PK] integer	full_name character (30)	total_deposit_sum numeric
1	21	Кузнецова Ан...	1000
2	2	Петров Алекс...	1000
3	20	Смирнов Серг...	1000
4	1	Иванова Мар...	1000

### Задание 3. Создать представление:

- содержащее сведения обо всех сотрудниках банка и заключенных ими договорах по кредитам за прошедший месяц;

Query

Query History

1

▼

CREATE VIEW v\_employee\_loans\_last\_month AS

2

SELECT

3

e.employee\_id,

4

e.full\_name,

5

l.loan\_id,

6

l.contract\_number,

7

l.loan\_date,

8

l.loan\_amount

9

FROM public.loan AS l

10

JOIN public.employee\_position AS ep

11

ON l.loan\_date BETWEEN ep.start\_date AND COALESCE(ep.end\_date, NOW())

12

JOIN public.employee AS e

13

ON ep.employee\_id = e.employee\_id

14

WHERE l.loan\_date >= date\_trunc('month', CURRENT\_DATE) - INTERVAL '1 month'

15

AND l.loan\_date < date\_trunc('month', CURRENT\_DATE);

16

Data Output

Messages

Notifications

CREATE VIEW

Query returned successfully in 79 msec.

Query

Query History

1

SELECT \* FROM public.v\_employee\_loans\_last\_month

2

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

Showing rows: 1 to 8

- найти клиентов банка, имеющих задолженности по кредитам.

Query Query History

```
1  ✓ CREATE VIEW v_clients_with_debt AS
2  SELECT DISTINCT
3      c.client_id,
4      c.full_name,
5      l.loan_id,
6      ls.balance,
7      ls.payment_date
8  FROM public.client AS c
9  JOIN public.passport AS p
10     ON p.client_id = c.client_id
11  JOIN public.loan AS l
12     ON l.passport_id = p.passport_id
13  JOIN public.loan_schedule AS ls
14     ON ls.loan_id = l.loan_id
15  WHERE (ls.actual_payment_date IS NULL AND ls.payment_date < NOW())
16         OR ls.balance > 0;
17
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 54 msec.

Query

Query History

1

2

SELECT \* FROM public.v\_clients\_with\_debt

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

	<div>client_id</div> <div>integer</div> <div>🔒</div>	<div>full_name</div> <div>character (32)</div> <div>🔒</div>	<div>loan_id</div> <div>integer</div> <div>🔒</div>	<div>balance</div> <div>numeric</div> <div>🔒</div>	<div>payment_date</div> <div>timestamp without time zone</div> <div>🔒</div>
1	11	Иванов Иван	1	83000	2025-05-01 00:00:00
2	12	Петров-Петр Петр ...	2	477000	2025-04-01 00:00:00
3	11	Иванов Иван	1	91500	2025-04-01 00:00:00
4	1	Иванов Иван Иванович ...	300	9000	2025-04-20 00:00:00
5	1	Иванов Иван Иванович ...	300	8000	2025-05-20 00:00:00
6	3	Сидорова Ольга Олегов...	302	7200	2025-03-20 00:00:00

**Задание: составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.**

Будем работать с новым клиентом

Query		Query History	
1	▼	INSERT INTO public.client (client_id, full_name, address, phone, email)	
2		VALUES	
3		(4, 'Новиков Николай Николаевич', 'Москва, ул. Пушкина, 10', '+7 9000000004', 'novikov@example.	
<div>&lt;div&gt;</div>			
Data Output Messages Notifications			
INSERT 0 1			

Query		Query History	
1	▼	INSERT INTO public.passport (passport_id, series, number, issue_date, issued_by, full_name, client_id)	
2		VALUES	
3		(13, '4503', '456789', '2021-06-01', 'ОВД Центрального района', 'Новиков Николай Николаевич', 4	
<div>&lt;div&gt;</div>			
Data Output Messages Notifications			
INSERT 0 1			
Query returned successfully in 59 msec.			

**INSERT**

Query

Query History

↗

```

1  INSERT INTO public.deposit (
2      deposit_id,
3      document_url,
4      return_amount,
5      deposit_amount,
6      actual_return_date,
7      contract_url,
8      contract_number,
9      expected_return_date,
10     deposit_date,
11     passport_id,
12     deposit_type_id,
13     currency_code
14 )
15 VALUES (
16     1100,
17     'https://bank.ru/docs/deposit1100.pdf',
18     10500,
19     10000,
20     NULL,
21     'https://bank.ru/contracts/dep1100.pdf',
22     'DEP-1100',
23     '2025-08-01 00:00:00',
24     '2025-05-01 00:00:00',
25     (SELECT passport_id
26        FROM public.passport
27        WHERE client_id = (
28            SELECT client_id
29            FROM public.client
30            WHERE full_name = 'Новиков Николай Николаевич'
31        )),
32     ),
33     (SELECT deposit_type_id
34        FROM public.deposit_type
35        WHERE name = 'Standard'
36     ),
37     (SELECT currency_code
38        FROM public.currency
39        WHERE currency_name = 'Рубль'
40     )
41 );
42

```

Data Output

Messages

Notifications

INSERT 0 1

Query returned successfully in 48 msec.

## Проверим

Query

Query History

↗

Scratch Pad X

↗

```

1  select * from deposit where deposit_id=1100

```

Data Output

Messages

Notifications

Showing rows: 1 to 1

Page No: 1

of 1

⏪ ⏩ ⏴ ⏵

deposit_id [PK] integer	document_url character (100)	return_amount numeric	deposit_amount numeric	actual_return_date timestamp without time zone	contract_url character (100)	contract_number character varying (30)	expected_return_date timestamp without time zone
1100	https://bank.ru/docs/deposit1100.p...	10500	10000	[null]	https://bank.ru/contracts/dep1100.p...	DEP-1100	2025-08-01 00:00:00

## UPDATE

Повысим сумму вкладов нашего клиента на 5%

QueryQuery History

1

▼

UPDATE public.deposit

2

SET deposit\_amount = deposit\_amount \* 1.05

3

WHERE passport\_id = (

4

SELECT passport\_id

5

FROM public.passport

6

WHERE client\_id = (

7

SELECT client\_id

8

FROM public.client

9

WHERE full\_name = 'Новиков Николай Николаевич'

10

)

11

);

Data Output

Messages

Notifications

UPDATE 1

Query returned successfully in 76 msec.

Проверим

QueryQuery History

Scratch Pad ×

1

select \* from deposit where deposit\_id=1100

Data Output

Messages

Notifications

Showing rows: 1 to 1

Page No: 1

of 1

◀◀ ▶▶ ▶

	deposit_id [PK] integer	document_url character (100)	return_amount numeric	deposit_amount numeric	actual_return_date timestamp without time zone	contract_url character (100)	contract_number character varying (30)	expiration_date timestamp
1	1100	https://bank.ru/docs/deposit1100.p...	10500	10500.00	[null]	https://bank.ru/contracts/dep1100.p...	DEP-1100	2025

DELETE

А теперь удалим депозит этого клиента

Query

Query History

1

▼

DELETE FROM public.deposit

2

WHERE passport\_id = (

3

SELECT passport\_id

4

FROM public.passport

5

WHERE client\_id = (

6

SELECT client\_id

7

FROM public.client

8

WHERE full\_name = 'Новиков Николай Николаевич'

9

)

10

);

Data Output

Messages

Notifications

DELETE 1

Query returned successfully in 65 msec.

## Проверим

Query

Query History

1

Select \* from deposit

Data Output

Messages

Notifications

Showing rows: 1 to 10

Page No: 1

of 1

	deposit_id [PK] integer	document_url character (100)	return_amount numeric	deposit_amount numeric	actual_return_date timestamp without time zone	contract_url character (100)	contract_number character varying (30)	expiration_date timestamp
1	2	https://bank.ru/docs/deposit1.pdf ...	10500	10000	[null]	https://bank.ru/contracts/d1.pdf ...	D1-001	2024-01-01
2	3	https://bank.ru/docs/deposit2.pdf ...	5250	5000	[null]	https://bank.ru/contracts/d2.pdf ...	D2-002	2024-01-01
3	4	https://bank.ru/docs/deposit3.pdf ...	53500	50000	[null]	https://bank.ru/contracts/d3.pdf ...	D3-003	2024-01-01
4	1000	https://bank.ru/docs/deposit1000.p...	1050	1000	[null]	https://bank.ru/contracts/dep1000.p...	DEP-1000	2024-01-01
5	1001	https://bank.ru/docs/deposit1001.p...	5350	5000	[null]	https://bank.ru/contracts/dep1001.p...	DEP-1001	2024-01-01
6	1002	https://bank.ru/docs/deposit1002.p...	7371	7000	[null]	https://bank.ru/contracts/dep1002.p...	DEP-1002	2024-01-01
7	1004	https://bank.ru/docs/deposit1004.p...	4200	4000	[null]	https://bank.ru/contracts/dep1004.p...	DEP-1004	2024-01-01
8	1005	https://bank.ru/docs/deposit1005.p...	6300	6000	[null]	https://bank.ru/contracts/dep1005.p...	DEP-1005	2024-01-01
9	1006	https://bank.ru/docs/deposit1006.p...	10700	10000	[null]	https://bank.ru/contracts/dep1006.p...	DEP-1006	2024-01-01
10	1003	https://bank.ru/docs/deposit1003.p...	2100	2000	[null]	https://bank.ru/contracts/dep1003.p...	DEP-1003	2024-01-01

**Задание** изучить графическое представление запросов и просмотреть историю запросов.

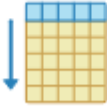


QueryQuery History

1Select \* from deposit

Data OutputMessagesExplain XNotifications

GraphicalAnalysisStatistics



deposit

QueryQuery History

Show queries generated internally by pgAdmin?☒

RemoveRemove All

Today - 01.05.2025

- ▶ insert into currency (currency\_name) values ...

15:08:53
- ▶ insert into currency (currency\_name) values ...

15:08:04
- ▶ select \* from currency

15:07:33
- ▶ select \* from currency\_rate

15:06:52
- ▶ select \* from currency

15:06:39
- ▶ select \* from currency\_rate

15:06:19
- ▶ select \* from currency

15:06:14
- ▶ insert into currency values ("Английские фут...
- ▶ insert into currency values ("Английские фут...
- ▶ select \* from currency

15:04:24

01.05.2025 15:08:58457 msec

DateRows affectedDuration

CopyCopy to Query Editor

select \* from currency

Messages

Successfully run. Total query runtime: 57 msec. 4 rows affected.

**Задание 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.**

Возьмём для примера два запроса:

Поиск всех вкладов, срок которых истекает завтра

Query

Query History

```

1  SELECT d.deposit_id, d.deposit_amount, d.expected_return_date
2  FROM public.deposit AS d
3  WHERE d.expected_return_date::date = CURRENT_DATE + INTERVAL '1 day';

```

Data Output

Messages

Explain

×

Notifications

Successfully run. Total query runtime: 76 msec.  
1 rows affected.

Data Output

Messages

Explain

×

Notifications

≡

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈










SQL

QUERY PLAN

text

🔒

1	Seq Scan on deposit d (cost=0.00..11.60 rows=1 width=44) (actual time=0.010..0.011 rows=1 loops=...
2	Filter: ((expected_return_date)::date = (CURRENT_DATE + '1 day'::interval))
3	Rows Removed by Filter: 9
4	Planning Time: 0.052 ms
5	Execution Time: 0.035 ms

Data Output	Messages	Explain	×	Notifications
				
			SQL	
	<b>QUERY PLAN</b> text 			
1	Limit (cost=29.64..29.65 rows=1 width=168) (actual time=0.045..0.046 rows=1 loops=1)			
2	InitPlan 1 (returns \$0)			
3	-> Seq Scan on currency (cost=0.00..16.50 rows=3 width=4) (actual time=0.004..0.004 rows=1 loops=1)			
4	Filter: (currency_name = 'Английские фунты'::bpchar)			
5	Rows Removed by Filter: 3			
6	-> Sort (cost=13.14..13.15 rows=1 width=168) (actual time=0.044..0.044 rows=1 loops=1)			
7	Sort Key: d.deposit_amount DESC			
8	Sort Method: quicksort Memory: 25kB			
9	-> Nested Loop (cost=0.00..13.13 rows=1 width=168) (actual time=0.026..0.028 rows=1 loops=1)			
10	Join Filter: (p.client_id = c.client_id)			
11	Rows Removed by Join Filter: 4			
12	-> Nested Loop (cost=0.00..12.07 rows=1 width=36) (actual time=0.023..0.025 rows=1 loops=1)			
13	Join Filter: (d.passport_id = p.passport_id)			
14	Rows Removed by Join Filter: 4			
15	-> Seq Scan on deposit d (cost=0.00..11.00 rows=1 width=36) (actual time=0.018..0.020 rows=1 loops=1)			
16	Filter: (currency_code = \$0)			
17	Rows Removed by Filter: 9			
18	-> Seq Scan on passport p (cost=0.00..1.03 rows=3 width=8) (actual time=0.002..0.002 rows=3 loops=1)			
19	-> Seq Scan on client c (cost=0.00..1.03 rows=3 width=136) (actual time=0.002..0.002 rows=5 loops=1)			
20	Planning Time: 0.189 ms			
21	Execution Time: 0.078 ms			

Теперь создадим индексы

Query	Query History			
1	CREATE INDEX idx_deposit_expected_return_date			
2	ON public.deposit (expected_return_date);			
Data Output	Messages	Explain	×	Notifications
CREATE INDEX				
Query returned successfully in 62 msec.				

Query

Query History

1

▼

CREATE INDEX idx\_deposit\_currency\_amount

2

ON public.deposit (currency\_code, deposit\_amount DESC);

3

Data Output

Messages

Explain

×

Notifications

CREATE INDEX

Query returned successfully in 67 msec.

Проверим время выполнения с индексами

Query

Query History

1

▼

EXPLAIN ANALYZE

2

SELECT d.deposit\_id, d.deposit\_amount, d.expected\_return\_date

3

FROM public.deposit AS d

4

WHERE d.expected\_return\_date::date = CURRENT\_DATE + INTERVAL '1 day';

Data Output

Messages

Explain

×

Notifications

≡+

▼

▼

SQL

QUERY PLAN

text

🔒

1

Seq Scan on deposit d (cost=0.00..1.20 rows=1 width=44) (actual time=0.010..0.011 rows=1 loops=...

2

Filter: ((expected\_return\_date)::date = (CURRENT\_DATE + '1 day'::interval))

3

Rows Removed by Filter: 9

4

Planning Time: 0.055 ms

5

Execution Time: 0.019 ms

Data Output	Messages	Explain	×	Notifications
<div> <div>≡</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>🗄️</div> <div>⬇️</div> <div>📈</div> <div>SQL</div> </div>				
	<b>QUERY PLAN</b> text <div>🔒</div>			
1	Limit (cost=19.77..19.77 rows=1 width=168) (actual time=0.031..0.032 rows=1 loops=1)			
2	InitPlan 1 (returns \$0)			
3	-> Seq Scan on currency (cost=0.00..16.50 rows=3 width=4) (actual time=0.004..0.005 rows=1 loop...			
4	Filter: (currency_name = 'Английские фунты'::bpchar)			
5	Rows Removed by Filter: 3			
6	-> Sort (cost=3.27..3.27 rows=1 width=168) (actual time=0.030..0.030 rows=1 loops=1)			
7	Sort Key: d.deposit_amount DESC			
8	Sort Method: quicksort Memory: 25kB			
9	-> Nested Loop (cost=0.00..3.26 rows=1 width=168) (actual time=0.023..0.025 rows=1 loops=1)			
10	Join Filter: (p.client_id = c.client_id)			
11	Rows Removed by Join Filter: 4			
12	-> Nested Loop (cost=0.00..2.19 rows=1 width=36) (actual time=0.020..0.021 rows=1 loops=1)			
13	Join Filter: (d.passport_id = p.passport_id)			
14	Rows Removed by Join Filter: 4			
15	-> Seq Scan on deposit d (cost=0.00..1.12 rows=1 width=36) (actual time=0.016..0.017 ro...			
16	Filter: (currency_code = \$0)			
17	Rows Removed by Filter: 9			
18	-> Seq Scan on passport p (cost=0.00..1.03 rows=3 width=8) (actual time=0.002..0.002 ro...			
19	-> Seq Scan on client c (cost=0.00..1.03 rows=3 width=136) (actual time=0.002..0.002 rows=...			
20	Planning Time: 0.133 ms			
21	Execution Time: 0.040 ms			

### Сравнение времени выполнения

Можно заметить, что после создания индексов время выполнения стало меньше. Следовательно, использование индексов позволяет ускорить выполнение запросов.

### Вывод:

В процессе выполнения лабораторной работы были отработаны методы модификации данных с помощью вложенных подзапросов (INSERT, UPDATE, DELETE). В pgAdmin изучено графическое представление планов выполнения запросов и история их выполнения. Для повышения производительности были созданы как простые, так и составные индексы, и с помощью EXPLAIN ANALYZE наглядно продемонстрировано, как индексы превращают последовательные сканирования в индексные проходы и сокращают время выполнения.