

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**»

Автор: Космач М.Р.

Факультет: ИКТ

Группа: К3239

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

Оглавление

1. Запросы к базе данных.....	3
2. Представления.....	7
3. Кастом запросы.....	8
4. Индексы.....	12
Вывод.....	14

Цель работы: овладеть практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL, использования подзапросов при модификации данных и индексов.

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL, pgadmin 4.

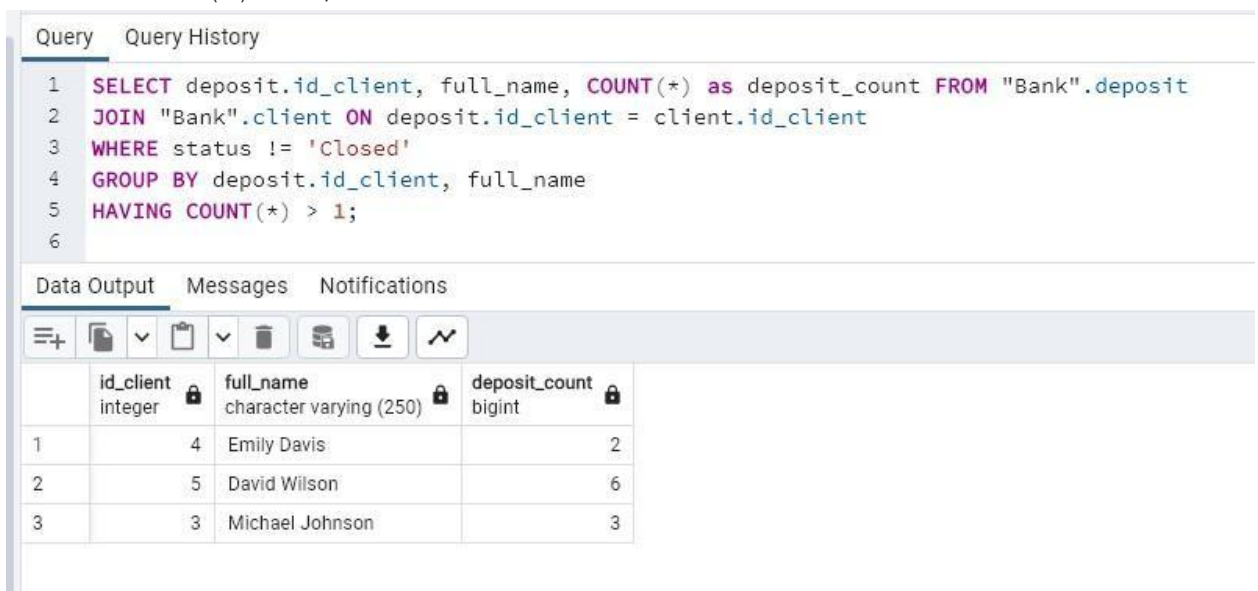
Практическое задание:

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.

1. Запросы к базе данных

1. Найти вкладчика, имеющего на текущий день несколько вкладов

```
SELECT deposit.id_client, full_name, COUNT(*) as deposit_count FROM
"Bank".deposit
JOIN "Bank".client ON deposit.id_client = client.id_client
WHERE status != 'Closed'
GROUP BY deposit.id_client, full_name
HAVING COUNT(*) > 1;
```



The screenshot shows the pgAdmin 4 interface. The 'Query' tab is active, displaying the following SQL query:

```
1 SELECT deposit.id_client, full_name, COUNT(*) as deposit_count FROM "Bank".deposit
2 JOIN "Bank".client ON deposit.id_client = client.id_client
3 WHERE status != 'Closed'
4 GROUP BY deposit.id_client, full_name
5 HAVING COUNT(*) > 1;
6
```

The 'Data Output' tab is also visible, showing the results of the query in a table with 4 columns: `id_client` (integer), `full_name` (character varying (250)), and `deposit_count` (bigint). The results are as follows:

	id_client	full_name	deposit_count
1	4	Emily Davis	2
2	5	David Wilson	6
3	3	Michael Johnson	3

2. Найти вкладчика, имеющего вклады во всех видах валюты на данный момент

```
SELECT DISTINCT deposit.id_client, full_name FROM "Bank".deposit
JOIN "Bank".client ON deposit.id_client = client.id_client WHERE
NOT EXISTS (
```

```

SELECT currency_code FROM "Bank".currency
WHERE NOT EXISTS (
    SELECT * FROM "Bank".deposit
    WHERE Deposit.currency_code = Currency.currency_code
    AND Deposit.id_client = Deposit.id_client
)
)

```

Query Query History

```

1 SELECT DISTINCT deposit.id_client, full_name FROM "Bank".deposit
2 JOIN "Bank".client ON deposit.id_client = client.id_client
3 WHERE NOT EXISTS (
4     SELECT currency_code FROM "Bank".currency
5     WHERE NOT EXISTS (
6         SELECT * FROM "Bank".deposit
7         WHERE Deposit.currency_code = Currency.currency_code
8         AND Deposit.id_client = Deposit.id_client
9     )
10 )

```

Data Output Messages Notifications

	id_client integer	full_name character varying (250)
1	1	John Smith
2	5	David Wilson
3	4	Emily Davis
4	2	Jane Doe
5	3	Michael Johnson

3. Вывести данные вкладчика, имеющего максимальный вклад в английский фунтах.

```

SELECT client.*, MAX(deposit.deposit_amount) as max_deposit FROM "Bank".deposit

```

```

JOIN "Bank".client ON deposit.id_client = client.id_client

```

```

WHERE currency_code = (SELECT currency_code FROM "Bank".currency WHERE
currency.name = 'GBP')

```

```

GROUP BY deposit.id_client, client.passport_number

```

```

HAVING MAX(deposit.deposit_amount) >= ALL (
    SELECT MAX(deposit.deposit_amount) as max_deposit FROM "Bank".deposit

```

```

JOIN "Bank".client ON deposit.id_client = client.id_client

```

```

WHERE currency_code = (SELECT currency_code FROM "Bank".currency WHERE
currency.name = 'GBP')

```

```

GROUP BY deposit.id_client, client.passport_number

```

)

Query

Query History

1

SELECT client.*, MAX(deposit.deposit_amount) as max_deposit FROM "Bank".deposit

2

3

JOIN "Bank".client ON deposit.id_client = client.id_client

4

5

WHERE currency_code = (SELECT currency_code FROM "Bank".currency WHERE currency.name = 'GBP')

6

7

GROUP BY deposit.id_client, client.passport_number

8

9

HAVING MAX(deposit.deposit_amount) >= ALL (

10

SELECT MAX(deposit.deposit_amount) as max_deposit FROM "Bank".deposit

11

12

JOIN "Bank".client ON deposit.id_client = client.id_client

13

14

WHERE currency_code = (SELECT currency_code FROM "Bank".currency WHERE currency.name = 'GBP')

15

16

GROUP BY deposit.id_client, client.passport_number

17

)

Data Output

Messages

Notifications

	passport_number [PK] character varying (10)	address character varying (250)	phone_number character varying (20)	full_name character varying (250)	email character varying (250)	id_client integer	age smallint	max_deposit integer
1	5678901234	321 Cedar St, Anytown	555-3344	David Wilson	david.wilson@example.com	5	45	5000

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

```
SELECT deposit_directory.deposit_code, COUNT(*) as num_deposits FROM
"Bank".deposit
JOIN "Bank".deposit_directory ON deposit.currency_code =
deposit_directory.deposit_code
WHERE EXTRACT(year FROM deposit.loan_date) = EXTRACT(year FROM CURRENT_DATE)
- 1
GROUP BY deposit_directory.deposit_code
HAVING COUNT(*) >= ALL (
    SELECT COUNT(*) FROM "Bank".deposit
    JOIN "Bank".deposit_directory ON deposit.currency_code =
    deposit_directory.deposit_code
    WHERE EXTRACT(year FROM deposit.loan_date) = EXTRACT(year FROM
CURRENT_DATE) - 1
    GROUP BY deposit_directory.deposit_code
)
```

Query Query History

```
1 SELECT deposit_directory.deposit_code, COUNT(*) as num_deposits FROM "Bank".deposit
2 JOIN "Bank".deposit_directory ON deposit.currency_code = deposit_directory.deposit_code
3 WHERE EXTRACT(year FROM deposit.loan_date) = EXTRACT(year FROM CURRENT_DATE) - 1
4 GROUP BY deposit_directory.deposit_code
5 HAVING COUNT(*) >= ALL (
6     SELECT COUNT(*) FROM "Bank".deposit
7     JOIN "Bank".deposit_directory ON deposit.currency_code = deposit_directory.deposit_code
8     WHERE EXTRACT(year FROM deposit.loan_date) = EXTRACT(year FROM CURRENT_DATE) - 1
9     GROUP BY deposit_directory.deposit_code
10 )
```

Data Output Messages Notifications



	deposit_code [PK] integer	num_deposits bigint
1	1	8

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

```
SELECT employee.id_employee, full_name, COUNT(*) AS num_contracts FROM
"Bank".employee
JOIN "Bank".credit ON employee.id_employee = credit.id_employee
```

```

WHERE loan_date >= current_date - interval '1 month'
GROUP BY employee.id_employee
HAVING COUNT(*) >= ALL(
SELECT COUNT(*) FROM "Bank".employee
JOIN "Bank".credit ON employee.id_employee = credit.id_employee WHERE loan_date >=
current_date - interval '1 month'
GROUP BY employee.id_employee
)

```



The screenshot shows a database query editor with a query window and a data output window. The query window contains the following SQL code:

```

1 SELECT employee.id_employee, full_name, COUNT(*) AS num_contracts FROM "Bank".employee
2 JOIN "Bank".credit ON employee.id_employee = credit.id_employee WHERE loan_date >= current_date - interval '1 month'
3 GROUP BY employee.id_employee
4 HAVING COUNT(*) >= ALL(
5     SELECT COUNT(*) FROM "Bank".employee
6     JOIN "Bank".credit ON employee.id_employee = credit.id_employee WHERE loan_date >= current_date - interval '1 month'
7     GROUP BY employee.id_employee
8 )
9

```

The data output window shows the result of the query in a table with the following columns: id_employee, full_name, and num_contracts. The table contains one row of data:

id_employee	full_name	num_contracts
2	Jane Doe	1

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

```

SELECT client.full_name, current_debt FROM "Bank".credit

JOIN "Bank".client ON credit.id_client = client.id_client





WHERE planned_loan_closing_date = CURRENT_DATE + 1

```

Query Query History

```
1 SELECT client.full_name, current_debt FROM "Bank".credit
2 JOIN "Bank".client ON credit.id_client = client.id_client
3 WHERE planned_loan_closing_date = CURRENT_DATE + 1
```

Data Output Messages Notifications

        		
	full_name character varying (250) 	current_debt integer 
1	Michael Johnson	6000

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

```
SELECT      employee.id_employee, full_name,  MAX(initial_deposit_amount)
```

```
AS deposit_amount FROM "Bank".deposit
```

```
JOIN "Bank".employee ON deposit.id_employee = employee.id_employee
```

```
WHERE loan_date >= current_date - interval '1 month'
```

```
GROUP BY employee.id_employee
```

```
HAVING MAX(initial_deposit_amount) >= ALL (
```

```
SELECT MAX(initial_deposit_amount) AS deposit_amount FROM "Bank".deposit
```

```
JOIN "Bank".employee ON deposit.id_employee = employee.id_employee
```

```
WHERE loan_date >= current_date - interval '1 month'
```

```
GROUP BY employee.id_employee
```

```
)
```

Query

Query History

```
1 SELECT employee.id_employee, full_name, MAX(initial_deposit_amount) AS deposit_amount FROM "Bank".deposit
2 JOIN "Bank".employee ON deposit.id_employee = employee.id_employee
3 WHERE loan_date >= current_date - interval '1 month'
4 GROUP BY employee.id_employee
5
6 HAVING MAX(initial_deposit_amount) >= ALL (
7     SELECT MAX(initial_deposit_amount) AS deposit_amount FROM "Bank".deposit
8     JOIN "Bank".employee ON deposit.id_employee = employee.id_employee
9     WHERE loan_date >= current_date - interval '1 month'
10    GROUP BY employee.id_employee
11 )
12 |
```

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑️

📦

⬇️

📈

	id_employee [PK] integer	full_name character varying (50)	deposit_amount integer
1	2	Jane Doe	5000

2. Представления

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

```
CREATE VIEW "Bank".employee_info AS
SELECT employee.*, COUNT(credit.contract_number) AS num_contracts FROM
"Bank".employee
LEFT JOIN "Bank".credit ON employee.id_employee = credit.id_employee AND
credit.loan_date >= current_date - interval '1 month'
GROUP BY employee.id_employee
```

Query

Query History

1

SELECT * FROM "Bank".employee_info

2

Data Output

Messages

Notifications

	id_employee integer	age smallint	code_job integer	paasport_number character varying (10)	phone_nimber character varying (20)	full_name character varying (50)	address character varying (100)	num_contracts bigint
1	1	25	101	AB1234567	123-456-7890	John Smith	123 Main St	0
2	3	28	106	EF5432109	456-789-0123	Michael Johnson	789 Elm St	0
3	5	32	107	IJ8765432	012-345-6789	David Lee	654 Pine St	0
4	4	35	100	GH0987654	789-012-3456	Emily Wilson	321 Maple Ave	0
5	2	30	104	CD9876543	987-654-3210	Jane Doe	456 Oak St	1

9. Найти клиентов банка, имеющих задолженности по кредитам.

```
CREATE VIEW "Bank".clients_debts AS
SELECT client.*, current_debt FROM "Bank".credit
JOIN "Bank".client ON credit.id_client = client.id_client
WHERE current_debt > 0
```

Query

Query History

1

2

SELECT * FROM "Bank".clients_debts

Data Output

Messages

Notifications

	passport_number character varying (10)	address character varying (250)	phone_number character varying (20)	full_name character varying (250)	email character varying (250)	id_client integer	age smallint	current_debt integer
1	1234567890	123 Main St, Anytown	555-1234	John Smith	john.smith@example.com	1	25	10000
2	1234567890	123 Main St, Anytown	555-1234	John Smith	john.smith@example.com	1	25	7000
3	0987654321	456 Elm St, Anytown	555-5678	Jane Doe	jane.doe@example.com	2	30	1000
4	1234509876	789 Oak St, Anytown	555-9101	Michael Johnson	michael.johnson@example.com	3	35	6000

3. Кастом запросы

- Внести, что в контракт по кредиту, который был оформлен сотрудником с номером , было сегодня внесено зачисление на 50000 рублей

```
INSERT INTO "Bank".accruals(id_accruals, payments,
actual_date_payments, planned_date_payments, contract_number) VALUES
(11, 50000, CURRENT_DATE, '2023-12-20', (
SELECT contract_number FROM "Bank".credit
WHERE id_employee = 4
))
```

Data Output Messages Notifications						
	payments integer	id_accruals [PK] integer	actual_date_payments date	planned_date_payments date	contract_number integer	
1	1000	1	2022-06-01	2022-06-01	2	
2	2000	2	2022-06-07	2022-06-01	3	
3	3000	3	[null]	2022-07-01	4	
4	4000	4	[null]	2022-10-01	8	
5	5000	5	2022-10-05	2022-10-10	9	
6	6000	6	[null]	2022-12-01	8	
7	7000	7	[null]	2023-03-01	4	
8	8000	8	2022-09-07	2022-09-01	3	
9	9000	9	2023-01-01	2023-01-01	2	
10	10000	10	2022-11-05	2022-11-10	9	

Query Query History						
<pre> 1 SELECT * FROM "Bank".accruals 2 ORDER BY id_accruals ASC </pre>						
Data Output Messages Notifications						
	payments integer	id_accruals [PK] integer	actual_date_payments date	planned_date_payments date	contract_number integer	
1	1000	1	2022-06-01	2022-06-01	2	
2	2000	2	2022-06-07	2022-06-01	3	
3	3000	3	[null]	2022-07-01	4	
4	4000	4	[null]	2022-10-01	8	
5	5000	5	2022-10-05	2022-10-10	9	
6	6000	6	[null]	2022-12-01	8	
7	7000	7	[null]	2023-03-01	4	
8	8000	8	2022-09-07	2022-09-01	3	
9	9000	9	2023-01-01	2023-01-01	2	
10	10000	10	2022-11-05	2022-11-10	9	
11	50000	11	2023-12-17	2023-12-20	4	

• Изменить код валюты во вкладе на
USD UPDATE "Bank".deposit
SET currency_code = (SELECT currency_code FROM "Bank".currency WHERE currency.name
= 'USD')
WHERE contract_number = 3

Query

Query History

1

SELECT * FROM "Bank".deposit

2

ORDER BY contract_number ASC

Data Output

Messages

Notifications

	contract_number [PK] integer	loan_date date	actual_date_of_closing_deposit date	planned_date_of_closing_deposit date	initial_deposit_amount integer	deposit_amount integer	currency_code integer	id_client integer
1	1	2022-01-01	2023-01-01	2023-01-01	10000	5000	1	1
2	2	2022-02-01	[null]	2023-02-01	20000	15000	1	2
3	3	2022-03-01	[null]	2023-03-01	5000	5000	2	3
4	4	2022-04-01	[null]	2023-04-01	30000	25000	1	4
5	5	2022-05-01	2023-05-01	2023-05-01	100000	50000	1	5
6	6	2022-06-01	2022-12-01	2023-06-01	15000	10000	1	1
7	7	2022-07-01	2023-01-01	2023-07-01	5000	3000	2	2
8	8	2022-08-01	[null]	2023-08-01	20000	20000	1	3
9	9	2022-09-01	[null]	2023-09-01	10000	8000	1	4
10	10	2022-10-01	[null]	2023-10-01	5000	5000	2	5
11	11	2024-03-01	[null]	2024-01-01	5000	5500	3	5
12	12	2023-12-05	[null]	2024-01-01	2500	2000	4	5
13	13	2023-12-06	[null]	2025-01-01	4000	8000	5	5
14	14	2023-12-05	[null]	2024-01-01	1200	3600	6	5
15	15	2023-11-05	[null]	2024-01-01	10000	12000	7	5
16	16	2023-11-05	[null]	2024-01-01	4000	60000	4	3

Data Output								Messages		Notifications					
<div><div><div><div></div><div></div></div><div><div></div><div></div></div><div><div></div><div></div></div><div><div></div><div></div></div><div><div></div><div></div></div><div><div></div><div></div></div><div><div></div><div></div></div><div><div></div><div></div></div></div></div>															
	contract_number [PK] integer	loan_date date	actual_date_of_closing_deposit date	planned_date_of_closing_deposit date	initial_deposit_amount integer	deposit_amount integer	currency_code integer								
1	1	2022-01-01	2023-01-01	2023-01-01	10000	5000	1								
2	2	2022-02-01	[null]	2023-02-01	20000	15000	1								
3	3	2022-03-01	[null]	2023-03-01	5000	5000	1								
4	4	2022-04-01	[null]	2023-04-01	30000	25000	1								
5	5	2022-05-01	2023-05-01	2023-05-01	100000	50000	1								
6	6	2022-06-01	2022-12-01	2023-06-01	15000	10000	1								
7	7	2022-07-01	2023-01-01	2023-07-01	5000	3000	2								
8	8	2022-08-01	[null]	2023-08-01	20000	20000	1								
9	9	2022-09-01	[null]	2023-09-01	10000	8000	1								
10	10	2022-10-01	[null]	2023-10-01	5000	5000	2								
11	11	2024-03-01	[null]	2024-01-01	5000	5500	3								
12	12	2023-12-05	[null]	2024-01-01	2500	2000	4								
13	13	2023-12-06	[null]	2025-01-01	4000	8000	5								
14	14	2023-12-05	[null]	2024-01-01	1200	3600	6								
15	15	2023-11-05	[null]	2024-01-01	10000	12000	7								
16	16	2023-11-05	[null]	2024-01-01	4000	60000	4								

- Удалить данные о расписании оплаты для кредита, у которого статус 'Finished' и имя клиента "Michael Johnson"

```
DELETE FROM "Bank".payment_timetable
```

```
WHERE contract_number = (
```

```
    SELECT contract_number FROM "Bank".credit WHERE id_client = (SELECT id_client
FROM "Bank".client WHERE full_name = 'Michael Johnson') AND status = 'Finished'
) AND status = 'Finished'
```

Query Query History

```
1 SELECT * FROM "Bank".payments_timetable
2 ORDER BY "ID_payments_timetable" ASC
```

Data Output Messages Notifications



	ID_payments_timetable [PK] integer	payment_amount integer	actual_date_payments date	planned_data_payment date	contarct_number integer
1	1	6500	2022-07-26	2022-07-26	1
2	2	30000	2022-09-09	2022-09-09	2
3	3	14000	2022-09-15	2022-09-15	3
4	4	500000	2022-10-28	2022-10-28	4
5	5	1000	2022-10-28	2022-10-28	5
6	6	6500	2022-12-26	2022-12-26	1
7	7	30000	2023-01-09	2023-01-09	2

Query		Query History				
1	SELECT * FROM "Bank".payments_timetable					
2	ORDER BY "ID_payments_timetable" ASC					
Data Output		Messages				
	ID_payments_timetable [PK] integer	payment_amount integer	actual_date_payments date	planned_data_payment date	contract_number integer	
1	1	6500	2022-07-26	2022-07-26	1	
2	2	30000	2022-09-09	2022-09-09	2	
3	3	14000	2022-09-15	2022-09-15	3	
4	4	500000	2022-10-28	2022-10-28	4	
5	6	6500	2022-12-26	2022-12-26	1	
6	7	30000	2023-01-09	2023-01-09	2	

QueryHistory

Query		Query History		
Show queries generated internally by pgAdmin?		Remove Remove All		
Today - 17.12.2023				
SELECT * FROM "Bank".payments_timetable ORDER BY "ID_payments_timetable" ASC		17.12.2023 01:47:35		
01:47:35		6 Rows affected		
DELETE FROM "Bank".payments_timetable WHERE contract_number = (SELECT contract_num...		89 msec		
01:47:25		Duration		
DELETE FROM "Bank".payments_timetable WHERE contract_number = (SELECT contract_num...				
01:47:04				
DELETE FROM "Bank".payment_timetable WHERE contract_number = (SELECT contract_numbe...				
01:46:58				
SELECT * FROM "Bank".payments_timetable ORDER BY "ID_payments_timetable" ASC				
01:43:31				
ON DELETE CASCADE FROM "Bank".credit WHERE id_client = (SELECT id_client FROM "Bank"...				
01:42:46				
DELETE CASCADE FROM "Bank".credit WHERE id_client = (SELECT id_client FROM "Bank".cl...				
01:41:45				
DELETE FROM "Bank".credit WHERE id_client = (SELECT id_client FROM "Bank".client WHE...				
01:41:13				
SELECT * FROM "Bank".client ORDER BY passport_number ASC				
01:38:37				
SELECT * FROM "Bank".credit_directory ORDER BY credit_code ASC				
01:37:18				
SELECT * FROM "Bank".payments_timetable ORDER BY "ID_payments_timetable" ASC				
01:37:09				
SELECT * FROM "Bank".credit ORDER BY contract_number ASC				
01:37:07				
SELECT * FROM "Bank".deposit ORDER BY contract_number ASC				
01:36:37				
UPDATE "Bank".deposit SET currency_code = (SELECT currency_code FROM "Bank".currency...				
01:36:24				
SELECT * FROM "Bank".currency ORDER BY currency_code ASC				

```
CREATE INDEX currency_name_idx ON "Bank".currency (name);
CREATE INDEX loan_date_idx ON "Bank".credit (loan_date);
CREATE INDEX credit_end_dates ON "Bank".credit
(loan_date, planned_loan_closing_date);
```

SELECT client.*, MAX(deposit.deposit_amount) as max_deposit FROM "Bank".deposit

JOIN "Bank".client ON deposit.id_client = client.id_client

WHERE currency_code = (SELECT currency_code FROM "Bank".currency WHERE currency.name = 'GBP')

GROUP BY deposit.id_client, client.passport_number ORDER BY max_deposit DESC

Без индекса:

Query Query History

```
1 SELECT client.*, MAX(deposit.deposit_amount) as max_deposit FROM "Bank".deposit
2 JOIN "Bank".client ON deposit.id_client = client.id_client
3 WHERE currency_code = (SELECT currency_code FROM "Bank".currency WHERE currency.name = 'GBP')
4 GROUP BY deposit.id_client, client.passport_number
5 ORDER BY max_deposit DESC
6 LIMIT 1
7
```

Data Output Messages Notifications

Successfully run. Total query runtime: 526 msec.
1 rows affected.

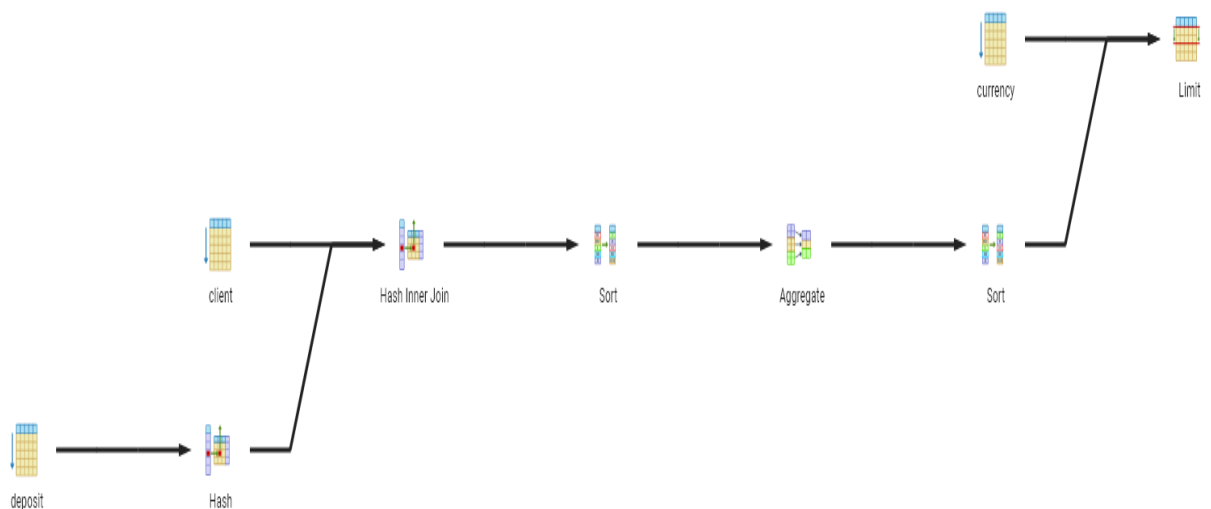
С индексом:

Query Query History

```
1 SELECT client.*, MAX(deposit.deposit_amount) as max_deposit FROM "Bank".deposit
2 JOIN "Bank".client ON deposit.id_client = client.id_client
3 WHERE currency_code = (SELECT currency_code FROM "Bank".currency WHERE currency.name = 'GBP')
4 GROUP BY deposit.id_client, client.passport_number
5 ORDER BY max_deposit DESC
6 LIMIT 1
7
```

Data Output Messages Notifications

Successfully run. Total query runtime: 65 msec.
1 rows affected.



DROP INDEX currency_name_idx;


```
DROP INDEX loan_date_idx  
DROP INDEX credit_end_dates;
```

Вывод

В ходе лабораторной работы была освоена работа с Query Tool, а именно с выборкой данных, добавлением, обновлением и их удалением. Также получены навыки создания представления данных и индексировании полей, в следствии чего были проведены оптимизационные эксперименты.