Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №2 «Запросы на выборку и модификацию данных, представления и индексы в PostgreSQL»

по дисциплине **«Проектирование и реализация баз данных»**

Автор: Афонина Н.Р.

Факультет: ИКТ

Группа: К32421

Преподаватель: Говорова М.М.

Санкт-Петербург 2023

Оглавление

Цель работы 3
Практическое задание 3
Вариант 19. БД «Банк» 3
Выполнение 4
Запросы к базе данных 4
Представления 10
Запросы на модификацию данных. 11
Создание индексов. 22
Вывод 24

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»

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. Хранится информация по своевременности ежемесячных выплат.

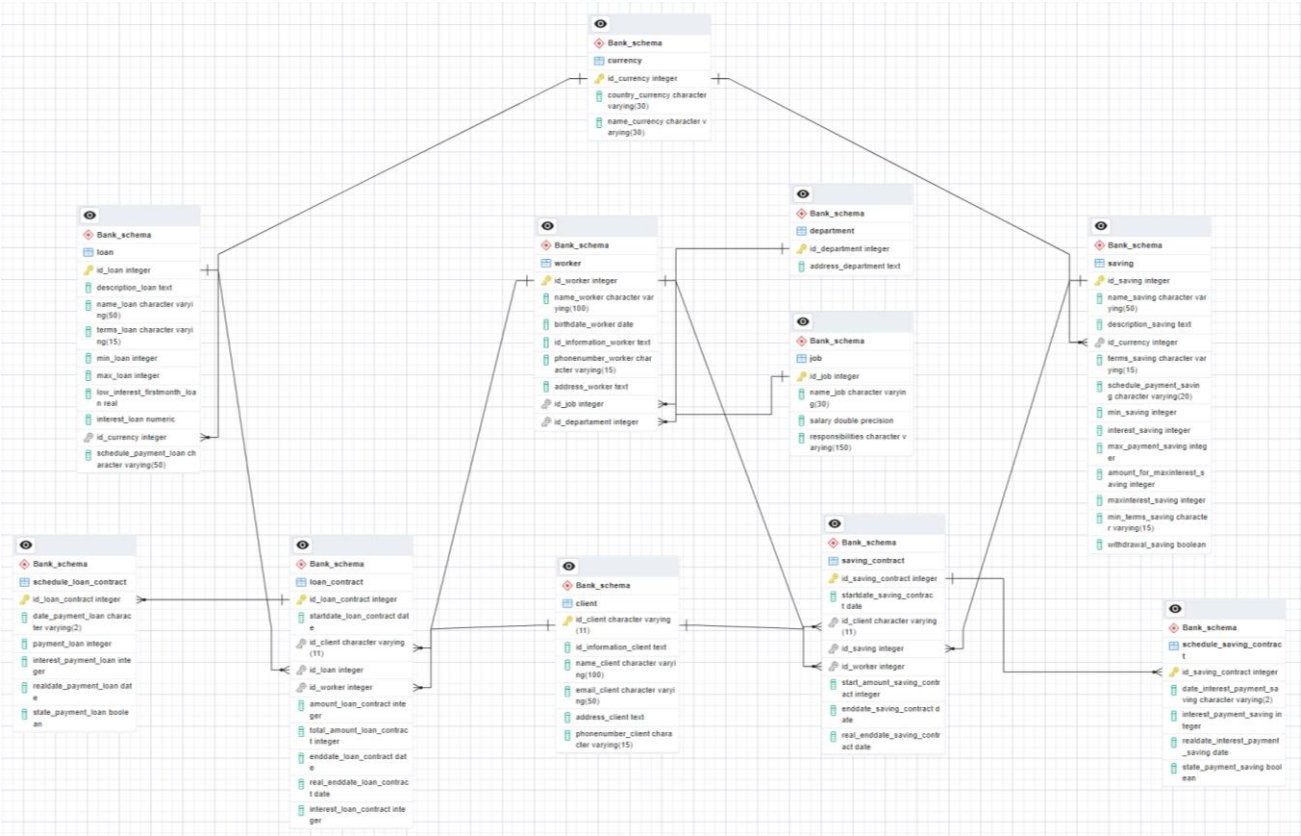
БД должна содержать следующий минимальный набор сведений: ФИО сотрудника. Возраст сотрудника. Адрес сотрудника. № телефона сотрудник. Паспортные данные сотрудника. Должность сотрудника. Оклад сотрудника (зависит от категории). Наименование вклада. Описание вклада. Минимальный срок

вклада. Минимальная сумма вклада. Процент по вкладу. Срок вклада. Процентная ставка. Код валюты. Наименование валюты. ФИО вкладчика. Адрес вкладчика. Телефон вкладчика. E-mail вкладчика. Паспортные данные. Номер договора. Дата вклада. Дата возврата. Сумма вклада. Сумма возврата. Данные по кредиту.

Выполнение

Название создаваемой БД – «Банк» («Bank»)

Схема логической модели базы данных, сгенерированная в Generate ERD



Запросы к базе данных

- Найти вкладчика, имеющего на текущий день несколько вкладов.

SELECT Cl.name_client, COUNT(Sc.id_saving_contract) AS number_of_savings

FROM ("Bank_schema".client Cl JOIN "Bank_schema".saving_contract Sc ON Cl.id_client=Sc.id_client)

WHERE Sc.startdate_saving_contract <= CURRENT_DATE AND Sc.enddate_saving_contract >= CURRENT_DATE

GROUP BY Cl.name_client

HAVING COUNT(Sc.id_saving_contract) > 1

Query Query History

```
SELECT Cl.name_client, COUNT(Sc.id_saving_contract) AS number_of_savings
FROM ("Bank_schema".client Cl JOIN "Bank_schema".saving_contract Sc ON Cl.id_client=Sc.id_client)
WHERE Sc.startdate_saving_contract <= CURRENT_DATE AND Sc.enddate_saving_contract >= CURRENT_DATE
GROUP BY Cl.name_client
HAVING COUNT(Sc.id_saving_contract) > 1
```

Data Output Messages Notifications

	name_client character varying (100)	number_of_savings bigint
1	Fatima Ahmed	3

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

SELECT Cl.name_client

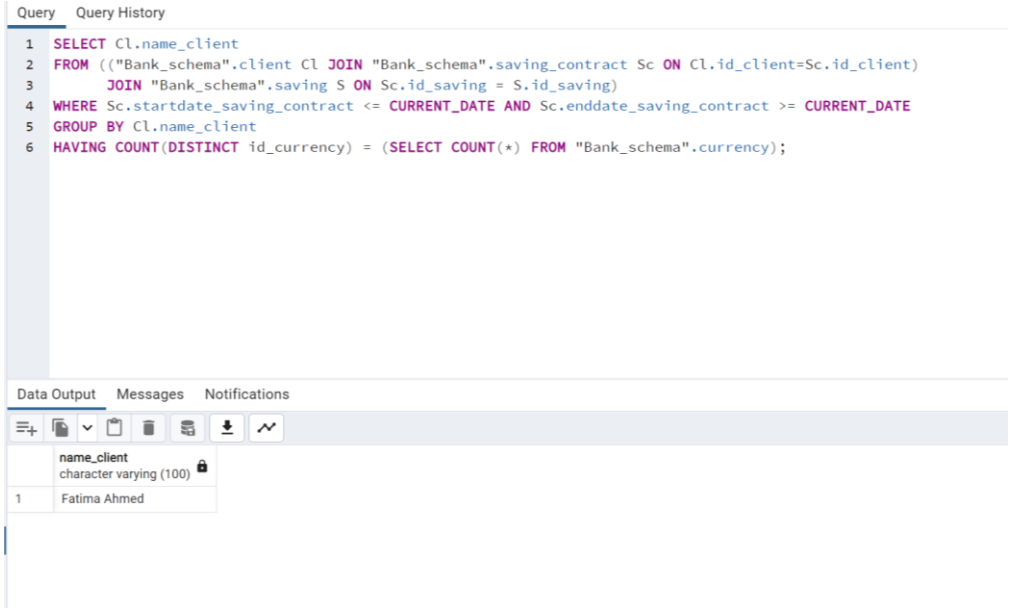
FROM (("Bank_schema".client Cl JOIN "Bank_schema".saving_contract Sc ON Cl.id_client=Sc.id_client)

JOIN "Bank_schema".saving S ON Sc.id_saving = S.id_saving)

WHERE Sc.startdate_saving_contract <= CURRENT_DATE AND Sc.enddate_saving_contract >= CURRENT_DATE

GROUP BY Cl.name_client

HAVING COUNT(DISTINCT id_currency) = (SELECT COUNT(*) FROM "Bank_schema".currency);



- Вывести данные вкладчика, имеющего максимальный вклад в английский фунтах (изменила на евро)

SELECT Cl.name_client, Sc.start_amount_saving_contract

FROM (("Bank_schema".client Cl JOIN "Bank_schema".saving_contract Sc ON Cl.id_client=Sc.id_client)

JOIN "Bank_schema".saving S ON Sc.id_saving = S.id_saving)

WHERE S.id_currency = ANY (SELECT id_currency FROM "Bank_schema".currency WHERE name_currency LIKE 'EUR') AND

Sc.start_amount_saving_contract = (SELECT MAX(Sc.start_amount_saving_contract) FROM (("Bank_schema".client Cl JOIN "Bank_schema".saving_contract Sc ON Cl.id_client=Sc.id_client)

JOIN "Bank_schema".saving S ON Sc.id_saving = S.id_saving)

WHERE S.id_currency = ANY (SELECT id_currency FROM "Bank_schema".currency WHERE name_currency LIKE 'EUR'))

Query Query History

```
SELECT Cl.name_client, Sc.start_amount_saving_contract
FROM (("Bank_schema".client Cl JOIN "Bank_schema".saving_contract Sc ON Cl.id_client=Sc.id_client)
      JOIN "Bank_schema".saving S ON Sc.id_saving = S.id_saving)
WHERE S.id_currency = ANY (SELECT id_currency FROM "Bank_schema".currency WHERE name_currency LIKE 'EUR') AND
Sc.start_amount_saving_contract = (SELECT MAX(Sc.start_amount_saving_contract) FROM (("Bank_schema".client Cl JOIN "
      JOIN "Bank_schema".saving S ON Sc.id_saving = S.id_saving)
                                 WHERE S.id_currency = ANY (SELECT id_currency FROM "Bank_schema".currency WHERE n
```

Data Output Messages Notifications

	name_client character varying (100)	start_amount_saving_contract integer
1	Marco Marcucci	10000

- Какой из вкладов пользовался наибольшей популярностью за истекший год.

SELECT S.name_saving, COUNT (Sc.id_saving) AS number_of_savings

FROM ("Bank_schema".saving_contract Sc JOIN "Bank_schema".saving S ON S.id_saving=Sc.id_saving)

GROUP BY S.name_saving

HAVING COUNT (Sc.id_saving) = (SELECT MAX(subquery.savings_count) FROM (SELECT COUNT(id_saving) AS savings_count FROM "Bank_schema".saving_contract GROUP BY id_saving) AS subquery)

Query Query History

```
SELECT S.name_saving, COUNT (Sc.id_saving) AS number_of_savings
FROM ("Bank_schema".saving_contract Sc JOIN "Bank_schema".saving S ON S.id_saving=Sc.id_saving)
GROUP BY S.name_saving
HAVING COUNT (Sc.id_saving) = (SELECT MAX(subquery.savings_count) FROM
                               (SELECT COUNT(id_saving) AS savings_count
                                FROM "Bank_schema".saving_contract GROUP BY id_saving) AS subquery)
```

Data Output Messages Notifications

	name_saving character varying (50)	number_of_savings bigint
1	Regular saving	4
2	Zero balance saving	4

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

SELECT S.name_worker, COUNT (Sc.id_loan) AS number_of_contracts

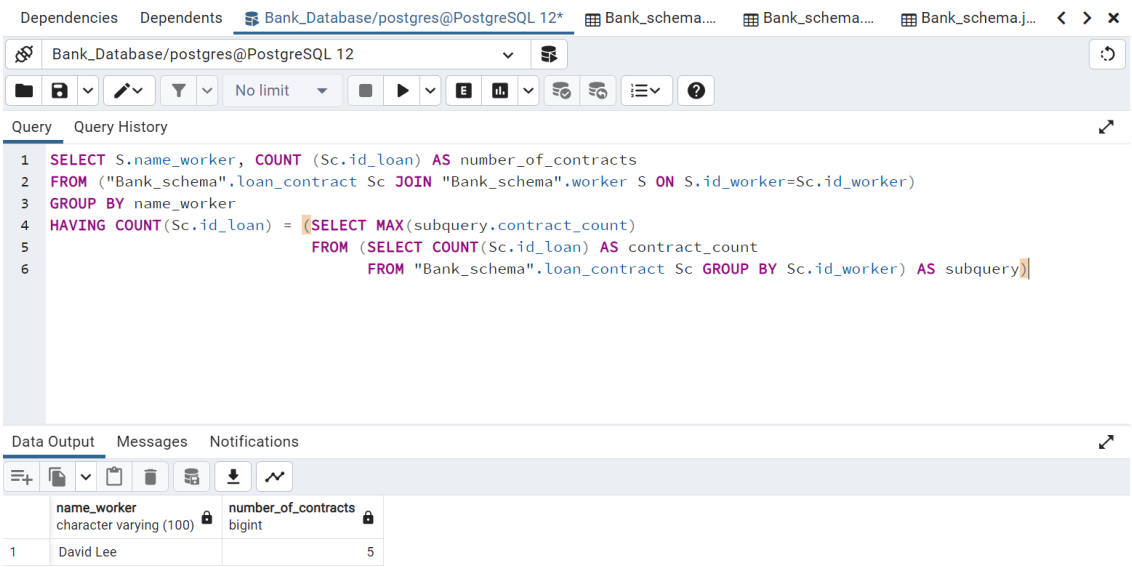
FROM ("Bank_schema".loan_contract Sc JOIN "Bank_schema".worker S ON S.id_worker=Sc.id_worker)

GROUP BY name_worker

HAVING COUNT(Sc.id_loan) = (SELECT MAX(subquery.contract_count)

FROM (SELECT COUNT(Sc.id_loan) AS contract_count

FROM "Bank_schema".loan_contract Sc GROUP BY Sc.id_worker) AS subquery)



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

SELECT Cl.name_client, Sc.interest_payment_saving

FROM (("Bank_schema".schedule_saving_contract Sc INNER JOIN "Bank_schema".saving_contract S ON S.id_saving_contract=Sc.id_saving_contract)

JOIN "Bank_schema".client Cl ON Cl.id_client=S.id_client)

WHERE enddate_saving_contract = '2023-03-25'

Query Query History

```
SELECT Cl.name_client, Sc.interest_payment_saving
FROM (("Bank_schema".schedule_saving_contract Sc INNER JOIN "Bank_schema".saving_contract S ON S.id_saving_contract=
      JOIN "Bank_schema".client Cl ON Cl.id_client=S.id_client)
WHERE enddate_saving_contract = '2023-03-25'
```

Data Output Messages Notifications

	name_client character varying (100)	interest_payment_saving integer
1	Fatima Ahmed	1680

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

SELECT S.name_worker, SUM(Sc.start_amount_saving_contract) AS amount_of_contracts
FROM ("Bank_schema".saving_contract Sc JOIN "Bank_schema".worker S ON S.id_worker=Sc.id_worker)
GROUP BY name_worker
HAVING SUM(Sc.start_amount_saving_contract) = (SELECT MAX(subquery.amount_contracts)
FROM (SELECT SUM(Sc.start_amount_saving_contract) AS amount_contracts
FROM "Bank_schema".saving_contract Sc GROUP BY Sc.id_worker) AS subquery)

Query Query History

```
SELECT S.name_worker, SUM(Sc.start_amount_saving_contract) AS amount_of_contracts
FROM ("Bank_schema".saving_contract Sc JOIN "Bank_schema".worker S ON S.id_worker=Sc.id_worker)
GROUP BY name_worker
HAVING SUM(Sc.start_amount_saving_contract) = (SELECT MAX(subquery.amount_contracts)
                                               FROM (SELECT SUM(Sc.start_amount_saving_contract) AS amount_contracts
                                                     FROM "Bank_schema".saving_contract Sc GROUP BY Sc.id_worker) AS subquery)
```

Data Output Messages Notifications

	name_worker character varying (100)	amount_of_contracts bigint
1	David Johns	38000

Представления

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

CREATE VIEW "Bank_schema".workers_contracts AS

SELECT L.id_worker, name_worker, id_loan_contract, amount_loan_contract, total_amount_loan_contract

FROM "Bank_schema".worker W, "Bank_schema".loan_contract L

WHERE W.id_worker=L.id_worker AND startdate_loan_contract <= CURRENT_DATE AND startdate_loan_contract >= '2023-02-23'

UNION

SELECT id_worker, name_worker, 0, 0, 0

FROM "Bank_schema".worker

WHERE id_worker NOT IN (SELECT id_worker FROM "Bank_schema".loan_contract)

ORDER BY id_worker

Query Query History

```
1  CREATE VIEW "Bank_schema".workers_contracts AS
2  SELECT L.id_worker, name_worker, id_loan_contract, amount_loan_contract, total_amount_loan_contract
3  FROM "Bank_schema".worker W, "Bank_schema".loan_contract L
4  WHERE W.id_worker=L.id_worker AND startdate_loan_contract <= CURRENT_DATE AND startdate_loan_contract >= '2023-02-23
5  UNION
6  SELECT id_worker, name_worker, 0, 0, 0
7  FROM "Bank_schema".worker
8  WHERE id_worker NOT IN (SELECT id_worker FROM "Bank_schema".loan_contract)
9  ORDER BY id_worker
10
11 SELECT * FROM "Bank_schema".workers_contracts
```

Data Output Messages Notifications

	id_worker integer	name_worker character varying (100)	id_loan_contract integer	amount_loan_contract integer	total_amount_loan_contract integer
1	1	John Black	0	0	0
2	2	Jane Willow	3	3000	3144
3	3	Bob Johnson	2	350000	487680
4	4	Mary Brown	0	0	0
5	5	David Lee	1	70000	83700
6	5	David Lee	4	200000	215640
7	5	David Lee	8	160000	172516
8	5	David Lee	7	160000	172516
9	5	David Lee	6	160000	172516
10	6	Sarah Kim	0	0	0
11	7	David Johns	0	0	0
12	8	Amanda Smith	5	1000	1026
13	9	Peter Brown	0	0	0

- Найти клиентов банка, имеющих задолженности по кредитам.

CREATE VIEW "Bank_schema".clients_loans AS

SELECT name_client, payment_loan, interest_payment_loan

FROM (("Bank_schema".client Cl JOIN "Bank_schema".loan_contract Lc ON Cl.id_client=Lc.id_client)

JOIN "Bank_schema".schedule_loan_contract Slc ON Lc.id_loan_contract=Slc.id_loan_contract)

WHERE state_payment_loan = 'false'

Query Query History

```
1 CREATE VIEW "Bank_schema".clients_loans AS
2 SELECT name_client, payment_loan, interest_payment_loan
3 FROM (("Bank_schema".client Cl JOIN "Bank_schema".loan_contract Lc ON Cl.id_client=Lc.id_client)
4       JOIN "Bank_schema".schedule_loan_contract Slc ON Lc.id_loan_contract=Slc.id_loan_contract)
5 WHERE state_payment_loan = 'false'
6
7
8 SELECT * FROM "Bank_schema".clients_loans
```

Data Output Messages Notifications

	name_client character varying (100)	payment_loan integer	interest_payment_loan integer
1	Jane Doe	70000	13700
2	Sarah Johnson	350000	137680
3	John Smith	3000	144
4	Hannah Marie Meloche	200000	15640
5	Michael Lee	1000	26
6	John Smith	160000	12516
7	Hannah Marie Meloche	160000	12516
8	Ellie Thumann	160000	12516

Запросы на модификацию данных.

Выполнить запросы на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов (составить самостоятельно). В отчете привести формулировку запроса, команду, скриншот до и после выполнения запроса. Учитывается сложность запроса;

- Запрос, который бы «открывал» сегодня клиентам из США, у которых нет вкладов, новый вклад/ы Regular Saving с суммой в 3000 долларов, дата открытия – текущий день. Отдел банка – 3. У работника, который открывает вклад, входит в обязанности открытие обычных вкладов. Дата окончания: спустя 5 лет текущей даты.

Скриншот «До»

Query Query History Scratch Pad

```
SELECT * FROM "Bank_schema".saving_contract
ORDER BY id_saving_contract ASC
```

Data Output Messages Notifications

	id_saving_contract [PK] integer	startdate_saving_contract date	id_client character varying (11)	id_saving integer	id_worker integer	start_amount_saving_contract integer	enddate_saving_contract date
1	1	2023-03-17	12345678908	1	9	2000	2028-03-17
2	2	2023-03-17	12345678909	2	7	150000	2026-03-17
3	3	2023-03-17	YA123456789	3	6	10000	2026-03-17
4	4	2023-03-17	4016565668	4	8	0	2028-03-17
5	5	2020-03-25	12345678909	3	7	8000	2023-03-25
6	6	2022-03-20	12345678909	5	7	30000	2025-03-20

Запрос:

CREATE TEMPORARY SEQUENCE tmp START 7;
INSERT
INTO "Bank_schema".saving_contract(id_saving_contract, startdate_saving_contract, id_client, id_saving, id_worker, start_amount_saving_contract, enddate_saving_contract, real_enddate_saving_contract)
SELECT nextval('tmp'), CURRENT_DATE, id_client, id_saving, id_worker, 3000, (CURRENT_DATE + INTERVAL '5 years'), null
FROM "Bank_schema".client Cl, "Bank_schema".saving, "Bank_schema".worker
WHERE id_client NOT IN (SELECT id_client FROM "Bank_schema".saving_contract) AND
id_client IN (SELECT id_client FROM "Bank_schema".client WHERE id_information_client LIKE '%U.S.A.%') AND
id_saving IN (SELECT id_saving FROM "Bank_schema".saving WHERE name_saving LIKE 'Regular saving') AND
id_worker IN (SELECT id_worker FROM ("Bank_schema".worker W JOIN "Bank_schema".job J ON W.id_job=J.id_job) WHERE responsibilities LIKE '%regular saving%' AND id_departament = 3)

Скриншот «После»

Query Query History

```
1 SELECT * FROM "Bank_schema".saving_contract
2 ORDER BY id_saving_contract ASC
```

Data Output Messages Notifications

	id_saving_contract [PK] integer	startdate_saving_contract date	id_client character varying (11)	id_saving integer	id_worker integer	start_amount_saving_contract integer	enddate_saving_contract date
1	1	2023-03-17	12345678908	1	9	2000	2028-03-17
2	2	2023-03-17	12345678909	2	7	150000	2026-03-17
3	3	2023-03-17	YA123456789	3	6	10000	2026-03-17
4	4	2023-03-17	4016565668	4	8	0	2028-03-17
5	5	2020-03-25	12345678909	3	7	8000	2023-03-25
6	6	2022-03-20	12345678909	5	7	30000	2025-03-20
7	7	2023-03-24	12345678901	1	9	3000	2028-03-24
8	8	2023-03-24	12345678903	1	9	3000	2028-03-24
9	9	2023-03-24	12345678902	1	9	3000	2028-03-24
10	10	2023-03-24	12345678904	1	9	3000	2028-03-24
11	11	2023-03-24	12345678900	1	9	3000	2028-03-24
12	12	2023-03-24	12345678911	1	9	3000	2028-03-24
13	13	2023-03-24	12345678912	1	9	3000	2028-03-24

Total rows: 13 of 13 Query complete 00:00:00.681 Ln 1, Col 1

- Запрос, который бы открывал сегодня кредиты на четырехлетнее обучение для клиентов 2000 года рождения в банке с адресом «123 Main Street, Los Angeles» на сумму 160000 долларов

Скриншот «До»

Bank_schema.loan_contract/Bank_Database/postgres@Postgr... No limit

Query Query History Scratch Pad

```
1 SELECT * FROM "Bank_schema".loan_contract
2 ORDER BY id_loan_contract ASC
```

Data Output Messages Notifications

	id_loan_contract [PK] integer	startdate_loan_contract date	id_client character varying (11)	id_loan integer	id_worker integer	amount_loan_contract integer	total_amount_loan_contract integer	end date
1	1	2023-03-17	12345678902	6	5	70000	83700	202
2	2	2023-03-17	12345678900	5	3	350000	487680	203
3	3	2023-03-17	12345678901	2	2	3000	3144	202
4	4	2023-03-17	12345678907	1	5	200000	215640	203
5	5	2023-03-17	12345678903	7	8	1000	1026	202

Запрос:

```
CREATE TEMPORARY SEQUENCE ls START 6;
INSERT
INTO "Bank_schema".loan_contract(id_loan_contract, startdate_loan_contract, id_client, id_loan, id_worker, amount_loan_contract, total_amount_loan_contract, enddate_loan_contract, real_enddate_loan_contract, interest_loan_contract)
SELECT nextval('ls'), CURRENT_DATE, id_client, id_loan, id_worker, 160000, 172516, (CURRENT_DATE + INTERVAL '9 years'), NULL, interest_loan
FROM "Bank_schema".client, "Bank_schema".loan, "Bank_schema".worker
WHERE id_client IN (SELECT id_client FROM "Bank_schema".client WHERE id_information_client LIKE '%2000%') AND
id_loan IN (SELECT DISTINCT id_loan FROM "Bank_schema".loan WHERE name_loan LIKE 'Student loan') AND
id_worker IN (SELECT DISTINCT id_worker FROM ("Bank_schema".worker W JOIN "Bank_schema".job J ON W.id_job=J.id_job)
                    WHERE responsibilities LIKE '%student loan%' AND
                    id_departament IN (SELECT DISTINCT id_department FROM "Bank_schema".department WHERE address_department LIKE '123 Main Street, Los Angeles')) AND
interest_loan IN (SELECT DISTINCT interest_loan FROM "Bank_schema".loan WHERE name_loan LIKE 'Student loan')
```

Скриншот «После»

Query Query History

```
SELECT * FROM "Bank_schema".loan_contract
ORDER BY id_loan_contract ASC
```

Scratch Pad

Data Output Messages Notifications

	id_loan_contract [PK] integer	startdate_loan_contract date	id_client character varying (11)	id_loan integer	id_worker integer	amount_loan_contract integer	total_amount_loan_contract integer	endd date
1	1	2023-03-17	12345678902	6	5	70000	83700	2026
2	2	2023-03-17	12345678900	5	3	350000	487680	2033
3	3	2023-03-17	12345678901	2	2	3000	3144	2024
4	4	2023-03-17	12345678907	1	5	200000	215640	2032
5	5	2023-03-17	12345678903	7	8	1000	1026	2023
6	6	2023-03-24	12345678901	1	5	160000	172516	2032
7	7	2023-03-24	12345678907	1	5	160000	172516	2032
8	8	2023-03-24	12345678911	1	5	160000	172516	2032

- Запрос, который бы создавал в таблице расписания выплат строки с новыми вкладами с датой выплаты – 24 число

Скриншот «До»

No limit

Query Query History

```
SELECT * FROM "Bank_schema".schedule_saving_contract
ORDER BY id_saving_contract ASC
```

Data Output Messages Notifications

	id_saving_contract [PK] integer	date_interest_payment_saving character varying (2)	interest_payment_saving integer	realdate_interest_payment_saving date	state_payment_saving boolean
1	1	17	5	[null]	false
2	2	17	31500	[null]	false
3	3	17	2100	[null]	false
4	4	17	0	[null]	false
5	5	25	1680	[null]	false
6	6	20	6300	[null]	false

Запрос:

INSERT INTO "Bank_schema".schedule_saving_contract(id_saving_contract, date_interest_payment_saving, state_payment_saving)

SELECT id_saving_contract, 24, false

FROM "Bank_schema".saving_contract

WHERE id_saving_contract NOT IN (SELECT id_saving_contract FROM "Bank_schema".schedule_saving_contract)

Скришот «После»

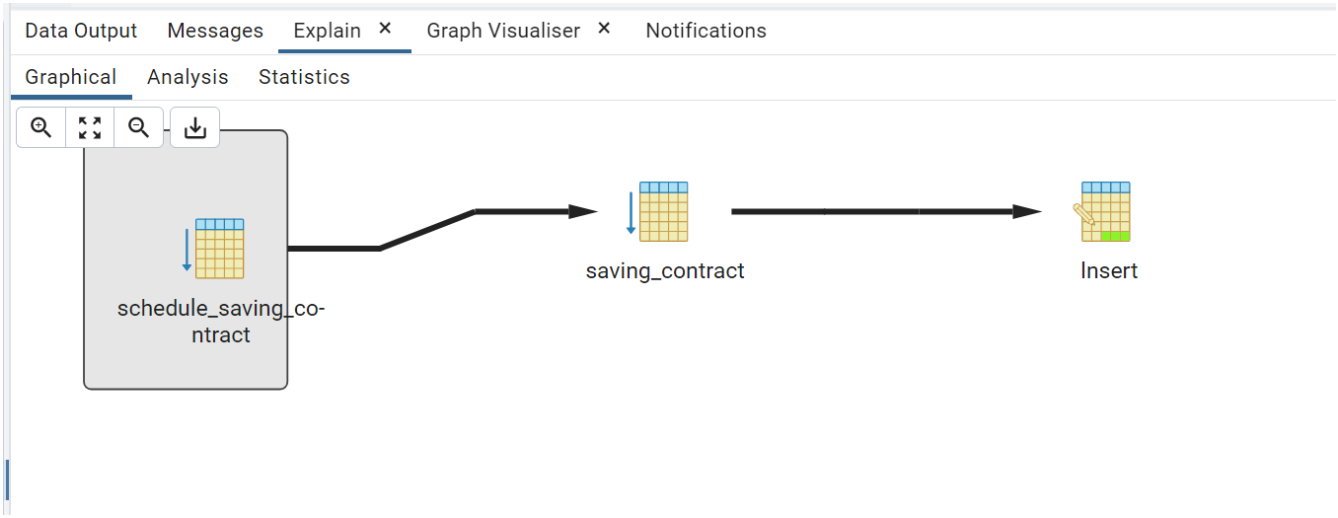
Query Query History

```
1 SELECT * FROM "Bank_schema".schedule_saving_contract
2 ORDER BY id_saving_contract ASC
```

Data Output Messages Notifications

	id_saving_contract [PK] integer	date_interest_payment_saving character varying (2)	interest_payment_saving integer	realdate_interest_payment_saving date	state_payment_saving boolean
1	1	17	5	[null]	false
2	2	17	31500	[null]	false
3	3	17	2100	[null]	false
4	4	17	0	[null]	false
5	5	25	1680	[null]	false
6	6	20	6300	[null]	false
7	7	24	[null]	[null]	false
8	8	24	[null]	[null]	false
9	9	24	[null]	[null]	false
10	10	24	[null]	[null]	false
11	11	24	[null]	[null]	false
12	12	24	[null]	[null]	false
13	13	24	[null]	[null]	false

Графическая диаграмма объяснения:



- Запрос

Скриншот «До»

Query Query History

```
1 SELECT * FROM "Bank_schema".schedule_loan_contract
2 ORDER BY id_loan_contract ASC
```

Data Output Messages Notifications

	id_loan_contract [PK] integer	date_payment_loan character varying (2)	payment_loan integer	interest_payment_loan integer	realdate_payment_loan date	state_payment_loan boolean
1	1	17	70000	13700	[null]	false
2	2	17	350000	137680	[null]	false
3	3	17	3000	144	[null]	false
4	4	17	200000	15640	[null]	false
5	5	17	1000	26	[null]	false

Запрос:

INSERT INTO "Bank_schema".schedule_loan_contract(id_loan_contract, date_payment_loan, payment_loan, interest_payment_loan, state_payment_loan)
SELECT id_loan_contract, 24, amount_loan_contract, (total_amount_loan_contract-amount_loan_contract), false
FROM "Bank_schema".loan_contract
WHERE id_loan_contract NOT IN (SELECT id_loan_contract FROM "Bank_schema".schedule_loan_contract)

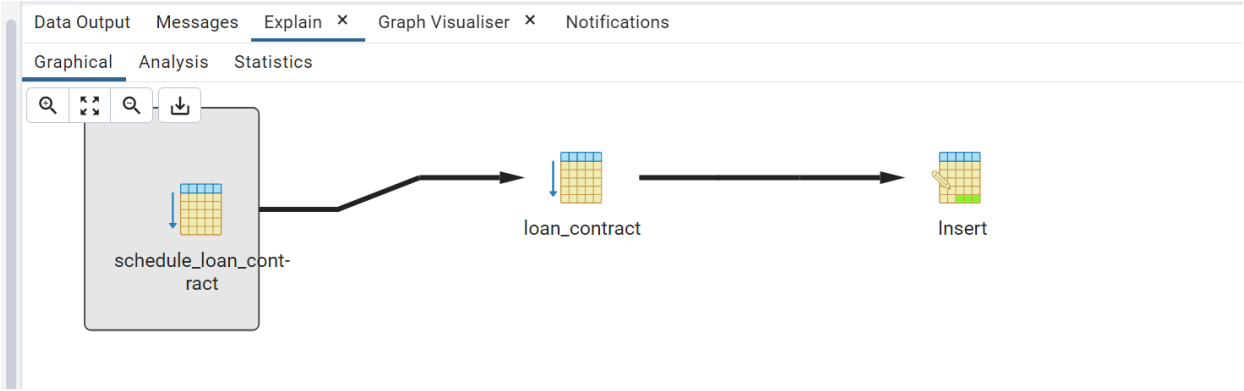
Скриншот «После»

```
1  SELECT * FROM "Bank_schema".schedule_loan_contract
2  ORDER BY id_loan_contract ASC
```

Data Output Messages Notifications

	id_loan_contract [PK] integer	date_payment_loan character varying (2)	payment_loan integer	interest_payment_loan integer	realdate_payment_loan date	state_payment_loan boolean
1	1	17	70000	13700	[null]	false
2	2	17	350000	137680	[null]	false
3	3	17	3000	144	[null]	false
4	4	17	200000	15640	[null]	false
5	5	17	1000	26	[null]	false
6	6	24	160000	12516	[null]	false
7	7	24	160000	12516	[null]	false
8	8	24	160000	12516	[null]	false

Графическая диаграмма объяснения:



- Запрос, который удаляет вклады иностранцев, открытые в долларах в офисе по адресу 789 Hollywood Boulevard, Los Angeles

Скриншот «До»

Query Query History

```
SELECT * FROM "Bank_schema".saving_contract
ORDER BY id_saving_contract ASC
```

Data Output Messages Notifications

	id_saving_contract [PK] integer	startdate_saving_contract date	id_client character varying (11)	id_saving integer	id_worker integer	start_amount_saving_contract integer	enddate_saving_contract date
1	1	2023-03-17	12345678908	1	9	2000	2028-03-17
2	2	2023-03-17	12345678909	2	7	150000	2026-03-17
3	3	2023-03-17	YA123456789	3	6	10000	2026-03-17
4	4	2023-03-17	4016565668	4	8	0	2028-03-17
5	5	2020-03-25	12345678909	3	7	8000	2023-03-25
6	6	2022-03-20	12345678909	5	7	30000	2025-03-20
7	7	2023-03-24	12345678901	1	9	6000	2028-03-24
8	8	2023-03-24	12345678903	1	9	3000	2028-03-24
9	9	2023-03-24	12345678902	1	9	6000	2028-03-24
10	10	2023-03-24	12345678904	1	9	3000	2028-03-24
11	11	2023-03-24	12345678900	1	9	3000	2028-03-24
12	12	2023-03-24	12345678911	1	9	3000	2028-03-24
13	13	2023-03-24	12345678912	1	9	3000	2028-03-24
14	14	2023-03-17	YA123456789	4	8	1000	2028-03-17

Запрос:

DELETE FROM "Bank_schema".saving_contract

WHERE id_client IN (SELECT id_client FROM "Bank_schema".client WHERE id_information_client NOT LIKE '%U.S.A.%') AND

id_saving IN (SELECT id_saving FROM "Bank_schema".saving WHERE id_currency IN (SELECT id_currency FROM "Bank_schema".currency WHERE name_currency LIKE 'USD')) AND

id_worker IN (SELECT id_worker FROM "Bank_schema".worker WHERE id_departament IN (SELECT id_department FROM "Bank_schema".department WHERE address_department LIKE '789 Hollywood Boulevard, Los Angeles'))

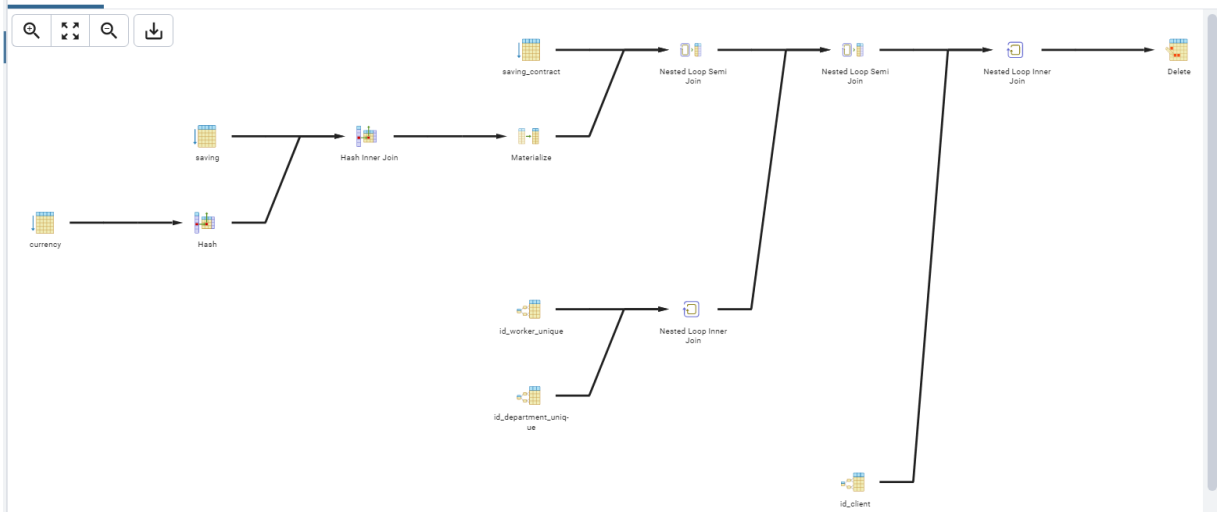
Скриншот «После»

```
SELECT * FROM "Bank_schema".saving_contract
ORDER BY id_saving_contract ASC
```

	id_saving_contract [PK] integer	startdate_saving_contract date	id_client character varying (11)	id_saving integer	id_worker integer	start_amount_saving_contract integer	enddate_saving_contract date
1	3	2023-03-17	YA123456789	3	6	10000	2026-03-17
2	5	2020-03-25	12345678909	3	7	8000	2023-03-25
3	6	2022-03-20	12345678909	5	7	30000	2025-03-20
4	7	2023-03-24	12345678901	1	9	6000	2028-03-24
5	8	2023-03-24	12345678903	1	9	3000	2028-03-24
6	9	2023-03-24	12345678902	1	9	6000	2028-03-24
7	10	2023-03-24	12345678904	1	9	3000	2028-03-24
8	11	2023-03-24	12345678900	1	9	3000	2028-03-24
9	12	2023-03-24	12345678911	1	9	3000	2028-03-24
10	13	2023-03-24	12345678912	1	9	3000	2028-03-24

Графическая диаграмма объяснения

```
DELETE FROM "Bank_schema".saving_contract
WHERE id_client IN (SELECT id_client FROM "Bank_schema".client WHERE id_information_client NOT LIKE '%U.S.A.%') AND
id_saving IN (SELECT id_saving FROM "Bank_schema".saving WHERE id_currency IN
              (SELECT id_currency FROM "Bank_schema".currency WHERE name_currency LIKE 'USD')) AND
id_worker IN (SELECT id_worker FROM "Bank_schema".worker WHERE id_departament IN
              (SELECT id_department FROM "Bank_schema".department WHERE address_department LIKE '789 Hollywood Boule
```



- Запрос, который бы изменял сумму вклада жителей Нью-Йорка на 6000 долларов

Скриншот «До»

Query Query History

```
1 SELECT * FROM "Bank_schema".saving_contract
2 ORDER BY id_saving_contract ASC
```

Data Output Messages Notifications

	id_saving_contract [PK] integer	startdate_saving_contract date	id_client character varying (11)	id_saving integer	id_worker integer	start_amount_saving_contract integer	enddate_saving_contract date
1	1	2023-03-17	12345678908	1	9	2000	2028-03-17
2	2	2023-03-17	12345678909	2	7	150000	2026-03-17
3	3	2023-03-17	YA123456789	3	6	10000	2026-03-17
4	4	2023-03-17	4016565668	4	8	0	2028-03-17
5	5	2020-03-25	12345678909	3	7	8000	2023-03-25
6	6	2022-03-20	12345678909	5	7	30000	2025-03-20
7	7	2023-03-24	12345678901	1	9	3000	2028-03-24
8	8	2023-03-24	12345678903	1	9	3000	2028-03-24
9	9	2023-03-24	12345678902	1	9	3000	2028-03-24
10	10	2023-03-24	12345678904	1	9	3000	2028-03-24
11	11	2023-03-24	12345678900	1	9	3000	2028-03-24
12	12	2023-03-24	12345678911	1	9	3000	2028-03-24
13	13	2023-03-24	12345678912	1	9	3000	2028-03-24

Запрос:

UPDATE "Bank_schema".saving_contract
SET start_amount_saving_contract = 6000
WHERE id_client IN (SELECT id_client FROM "Bank_schema".client WHERE id_information_client LIKE '%New York%')

Скриншот «После»

```
SELECT * FROM "Bank_schema".saving_contract
ORDER BY id_saving_contract ASC
```

Data Output Messages Notifications

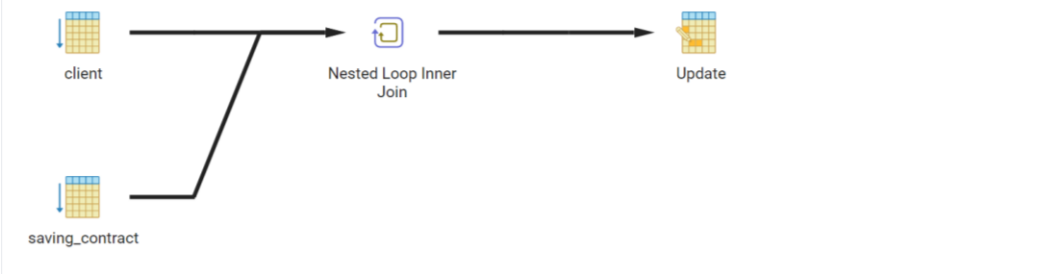
	id_saving_contract [PK] integer	startdate_saving_contract date	id_client character varying (11)	id_saving integer	id_worker integer	start_amount_saving_contract integer	enddate_saving_contract date
1	1	2023-03-17	12345678908	1	9	2000	2028-03-17
2	2	2023-03-17	12345678909	2	7	150000	2026-03-17
3	3	2023-03-17	YA123456789	3	6	10000	2026-03-17
4	4	2023-03-17	4016565668	4	8	0	2028-03-17
5	5	2020-03-25	12345678909	3	7	8000	2023-03-25
6	6	2022-03-20	12345678909	5	7	30000	2025-03-20
7	7	2023-03-24	12345678901	1	9	6000	2028-03-24
8	8	2023-03-24	12345678903	1	9	3000	2028-03-24
9	9	2023-03-24	12345678902	1	9	6000	2028-03-24
10	10	2023-03-24	12345678904	1	9	3000	2028-03-24
11	11	2023-03-24	12345678900	1	9	3000	2028-03-24
12	12	2023-03-24	12345678911	1	9	3000	2028-03-24
13	13	2023-03-24	12345678912	1	9	3000	2028-03-24

Графическая диаграмма объяснения

```
UPDATE "Bank_schema".saving_contract
SET start_amount_saving_contract = 6000
WHERE id_client IN (SELECT id_client FROM "Bank_schema".client WHERE id_information_client LIKE '%New York%')
```

Data Output Messages Explain × Graph Visualiser × Notifications

Graphical Analysis Statistics



Создание индексов.

- Запрос на поиск клиентов с номерами, начинающимися на +39

SELECT * FROM "Bank_schema".client

WHERE phonenumber_client LIKE '+39%'

Query Query History

```
WHERE Cl.id_information_client LIKE 'U.S.A%'
GROUP BY Cl.name_client

SELECT * FROM pg_indexes WHERE tablename = 'client'

DROP INDEX "Bank_schema".my_index2

SELECT * FROM "Bank_schema".client
WHERE phonenumber_client LIKE '+39%'
```

Data Output Messages Explain × Notifications

```
Successfully run. Total query runtime: 201 msec.
3 rows affected.
```

```
CREATE INDEX my_index ON "Bank_schema".client (phonenumber_client)

SELECT * FROM pg_indexes WHERE tablename = 'client'
```

Data Output Messages Notifications

	schemaname name	tablename name	indexname name	tablespace name	indexdef text
1	Bank_schema	client	client_pkey	[null]	CREATE UNIQUE INDEX client_pkey ON "Bank_schema".client USING btree (id_clie...
2	Bank_schema	client	id_client	[null]	CREATE UNIQUE INDEX id_client ON "Bank_schema".client USING btree (id_client)
3	Bank_schema	client	my_index	[null]	CREATE INDEX my_index ON "Bank_schema".client USING btree (phonenumber_cl...

- Запрос

SELECT Cl.name_client, COUNT(Sc.id_saving_contract) AS number_of_savings

FROM ("Bank_schema".client Cl JOIN "Bank_schema".saving_contract Sc ON Cl.id_client=Sc.id_client)

WHERE Cl.id_information_client LIKE 'U.S.A%'

GROUP BY Cl.name_client

Query Query History

```
SELECT Cl.name_client, COUNT(Sc.id_saving_contract) AS number_of_savings
FROM ("Bank_schema".client Cl JOIN "Bank_schema".saving_contract Sc ON Cl.id_client=Sc.id_client)
WHERE Cl.id_information_client LIKE 'U.S.A%'
GROUP BY Cl.name_client
```

Data Output Messages Explain × Notifications

```
Successfully run. Total query runtime: 221 msec.
6 rows affected.
```

Query Query History

```
CREATE INDEX my_index ON "Bank_schema".client (name_client, id_information_client)
CREATE INDEX my_index2 ON "Bank_schema".saving_contract (id_client, id_saving_contract)

EXPLAIN
SELECT Cl.name_client, COUNT(Sc.id_saving_contract) AS number_of_savings
FROM ("Bank_schema".client Cl JOIN "Bank_schema".saving_contract Sc ON Cl.id_client=Sc.id_client)
WHERE Cl.id_information_client LIKE 'U.S.A%'
GROUP BY Cl.name_client
```

Data Output Messages Explain Notifications

	QUERY PLAN text
1	GroupAggregate (cost=3.52..3.54 rows=1 width=226)
2	Group Key: cl.name_client
3	-> Sort (cost=3.52..3.53 rows=1 width=222)
4	Sort Key: cl.name_client
5	-> Hash Join (cost=2.39..3.51 rows=1 width=222)
6	Hash Cond: ((sc.id_client)::text = (cl.id_client)::text)
7	-> Seq Scan on saving_contract sc (cost=0.00..1.10 rows=10 width=15)
8	-> Hash (cost=2.38..2.38 rows=1 width=258)
9	-> Seq Scan on client cl (cost=0.00..2.38 rows=1 width=258)
10	Filter: (id_information_client ~~ 'U.S.A%'::text)

Индексы создаются для ускорения поиска данных в таблице. При выполнении запроса СУБД выбирает план выполнения, который оптимизирует выполнение запроса на основе статистики таблицы, доступных индексов и других факторов. Если индекс не используется в выбранном плане запроса, это может быть связано с тем, что СУБД решила, что использование индекса неоптимально для данного запроса. Например, если в таблице мало строк и индекс не является уникальным, СУБД может решить выполнить полный скан таблицы, так как использование индекса займет больше времени, чем простое сканирование таблицы, что и произошло в данной ситуации.

Вывод

В данной лабораторной работе выполнены различные запросы к базе данных «Банк», используя соединения таблиц, подзапросы и др. Были выполнены запросы на создание представлений, а также на модификацию данных: вставка, изменение и удаление. Были выполнены запросы без индекса и созданы планы запросов через EXPLAIN, далее были созданы различные индексы для различных запросов, но ни в одном из них не получилось их использовать, так как выполнение запроса без индекса вероятнее происходило быстрее из-за маленького количества данных в таблицах.