

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

Отчёт

по лабораторной работе №4 «Запросы на выборку и модификацию данных.
Представления. Работа с индексами»

По дисциплине «Проектирование и реализация баз данных»

Автор: Сергеев В. Ю.

Факультет: ИКТ

Группа: K3241

Преподаватель: Говорова М. М.



Санкт-Петербург, 2023

Оглавление

Содержание отчёта

Оглавление	2
Содержание работы	3
Цель работы	3
Практическое задание	3
Вариант 19. БД «Банк»	3
Выполнение	4
Select	4
Modifications	11
View	15
Index	18
Вывод	23

Содержание работы

Цель работы

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

Практическое задание

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3). Создать схему в составе базы данных
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»

Задание 2: Создайте запросы:

- Найти вкладчика, имеющего на текущий день несколько вкладов.
- Найти вкладчика, имеющего вклады во всех видах валюты на текущий день.
- Вывести данные вкладчика, имеющего максимальный вклад в английский фунтах.
- Какой из вкладов пользовался наибольшей популярностью за истекший 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.

Выполнение

Select

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

```
select cl."PassportNumber", cl."FullName"
from "bankDB"."Client" cl, "bankDB"."DepositAgreement" da
where cl."PassportNumber" = da."PassportNumber"
and da."Status" = 'Open'
group by cl."PassportNumber"
having count(*) > 1;
```

The screenshot shows the pgAdmin 4 interface. The left pane displays the 'Object Explorer' with a tree view of the database structure. The 'Tables (10)' folder is expanded, showing tables like 'Client', 'DepositAgreement', and 'DepositType'. The 'Query' tab is active, displaying the SQL query. The 'Data Output' pane shows the results of the query, which is a single row with the passport number '952442573' and the full name 'Ираклий Елисеевич Макаров'. The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.165'.

PassportNumber [PK] bigint	FullName character varying (50)
952442573	Ираклий Елисеевич Макаров

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Total rows: 1 of 1 Query complete 00:00:00.165 Ln 6, Col 20

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

```
select da."PassportNumber"
from "bankDB"."DepositAgreement" da
group by da."PassportNumber"
having count(distinct da."IdCurrency") =
(select count(*) from "bankDB"."Currency");
```

Properties Dashboard SQL Statistics Dependents select_queries... Bank/postgres@PostgreSQL 16* Bank/postgres...

Bank/postgres@PostgreSQL 16

Query Query History

1 select da."PassportNumber"

2 from "bankDB"."DepositAgreement" da

3 group by da."PassportNumber"

4 having count(distinct da."IdCurrency") =

5 (select count(*) from "bankDB"."Currency");

Scratch Pad

Data Output Notifications Messages

PassportNumber
integer

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

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

```
select cl.*
from "bankDB"."Client" cl, "bankDB"."DepositAgreement" da
where cl."PassportNumber" = da."PassportNumber"
and da."IdCurrency" =
(select "IdCurrency" from "bankDB"."Currency" where "Name"='фунт стерлингов')
and da."StartSum" =
(select max(da."StartSum")
 from "bankDB"."DepositAgreement" da
 where da."IdCurrency" =
 (select "IdCurrency" from "bankDB"."Currency" where "Name"='фунт
стерлингов')));
```

The screenshot shows a PostgreSQL client interface with a query editor and a results pane. The query editor contains the following SQL query:

```
1 select cl.*
2 from "bankDB"."Client" cl, "bankDB"."DepositAgreement" da
3 where cl."PassportNumber" = da."PassportNumber"
4 and da."IdCurrency" =
5 (select "IdCurrency" from "bankDB"."Currency" where "Name"='фунт стерлингов')
6 and da."StartSum" =
7 (select max(da."StartSum")
8  from "bankDB"."DepositAgreement" da
9  where da."IdCurrency" =
10 (select "IdCurrency" from "bankDB"."Currency" where "Name"='фунт стерлингов'));
11
```

The results pane displays a table with the following columns and data:

	PassportNumber [PK] bigint	FullName character varying (50)	Address character varying (150)	Phone bigint	E-mail character varying (30)
1	189266391	Крюкова Елизавета Борисовна	д. Оленегорск (Якут.), пр. Королева, д. 95, 771803	86970156808	oktjabrina_16@gmail.com

At the bottom of the interface, a status bar indicates: "Total rows: 1 of 1", "Query complete 00:00:00.150", and a green message box stating "Successfully run. Total query runtime: 150 msec. 1 rows affected." The interface also shows a Windows activation watermark and a "Scratch Pad" tab.

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

```
select dt.*
from "bankDB"."DepositType" dt
join "bankDB"."DepositAgreement" da
on dt."Id" = da."IdType"
group by dt."Id"
having count(da.*) =
(select max(tbl.counter) from
(select count(da.*) as counter
from "bankDB"."DepositAgreement" da
where da."DepositDate" > (current_date - interval '1 YEAR')
group by da."IdType") tbl)
```

Properties Dashboard SQL Statistics Dependents select_queries... Bank/postgres@PostgreSQL 16* Bank/postgres...

Bank/postgres@PostgreSQL 16

No limit

Query Query History

Scratch Pad

1 select dt.*
2 from "bankDB"."DepositType" dt
3 join "bankDB"."DepositAgreement" da
4 on dt."Id" = da."IdType"
5 group by dt."Id"
6 having count(da.*) =
7 (select max(tbl.counter) from
8 (select count(da.*) as counter
9 from "bankDB"."DepositAgreement" da
10 where da."DepositDate" > (current_date - interval '1 YEAR')
11 group by da."IdType") tbl)

Data Output Notifications Messages

	Id [PK] integer	Name character varying (30)	Description character varying (200)
1	1	Красивый	Спорт кидать теория основание промолчать инструкция скользить. Угроза район призыв дрогнуть привлекать г
2	2	Умный	Видимо хлеб за песня художественный теория. Войти столетие дошлый совет пространство затянуться лиловы

Активация Windows
Чтобы активировать Windows, перейдите в раздел
"Параметры".

Total rows: 2 of 2 Query complete 00:00:00.124 Ln 11, Col 27

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

```
select em."TabelNumber", em."FullName"
from "bankDB"."Employee" em
join "bankDB"."LoanAgreement" la
on em."TabelNumber" = la."TabelNumber"
group by em."TabelNumber"
having count(la.*) =
(select max(tbl.counter) from
(select count(la.*) as counter
from "bankDB"."LoanAgreement" la
where la."LoanDate" > (current_date - interval '1 MONTH') -- not enough data
group by la."TabelNumber"
union -- cus null
select(0)) tbl)
```

Bank/postgres@PostgreSQL 16

Query Query History

```
1 select em."TabelNumber", em."FullName"
2 from "bankDB"."Employee" em
3 join "bankDB"."LoanAgreement" la
4 on em."TabelNumber" = la."TabelNumber"
5 group by em."TabelNumber"
6 having count(la.*) =
7 (select max(tbl.counter) from
8 (select count(la.*) as counter
9 from "bankDB"."LoanAgreement" la
10 -- where la."LoanDate" > (current_date - interval '1 MONTH') -- not enough data
11 group by la."TabelNumber"
12 union -- cus null
13 select(0)) tbl)
```

Data Output Notifications Messages

	TabelNumber [PK] integer	FullName character varying (50)
1	100017	Агап Адамович Матвеев

Активация Windows
Чтобы активировать Windows, перейдите в раздел
Параметры

✓ Successfully run. Total query runtime: 189 msec. 1 rows affected. ✕

Total rows: 1 of 1 Query complete 00:00:00.189 Ln 10, Col 14

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

```
select cl."PassportNumber", cl."FullName", da."StartSum" + sum(dps."Payment")
as "TotalSum"
from "bankDB"."Client" cl, "bankDB"."DepositAgreement" da,
"bankDB"."DepositPaySchedule" dps
where cl."PassportNumber" = da."PassportNumber"
and da."AgreementNumber" = dps."AgreementNumber"
and da."PlanCloseDate" = (current_date + interval '1 DAY') -- not enough data
group by cl."PassportNumber", da."StartSum";
```

Bank/postgres@PostgreSQL 16

Query

Query History

1

```
select cl."PassportNumber", cl."FullName", da."StartSum" + sum(dps."Payment") as "TotalSum"
from "bankDB"."Client" cl, "bankDB"."DepositAgreement" da, "bankDB"."DepositPaySchedule" dps
where cl."PassportNumber" = da."PassportNumber"
and da."AgreementNumber" = dps."AgreementNumber"
-- and da."PlanCloseDate" = (current_date + interval '1 DAY') -- not enough data
group by cl."PassportNumber", da."StartSum";
```

Data Output

Notifications

Messages

PassportNumber
[PK] bigint

FullName
character varying (50)

TotalSum
real

1	123609496	Лука Вячеславович Наумов	256486.4
2	189266391	Крюкова Елизавета Борисовна	297024.34
3	129798791	Носов Анисим Владиславович	178177.1
4	952442573	Ираклий Елисеевич Макаров	214458.03
5	157712532	Лихачев Харитон Богданович	333166.72
6	952442573	Ираклий Елисеевич Макаров	107229
7	324629513	Владимирова Ольга Романовна	250137.72

Total rows: 7 of 7

Query complete 00:00:00.186

Ln 6, Col 45

Активация Windows
Чтобы активировать Windows, перейдите в раздел
"Параметры".

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

```
select em."TabelNumber", em."FullName", da."StartSum" as "AgreementStartSum"
from "bankDB"."Employee" em
join "bankDB"."DepositAgreement" da
on em."TabelNumber" = da."TabelNumber"
where da."StartSum" =
(select max(da."StartSum")
 from "bankDB"."DepositAgreement" da
 where da."DepositDate" > (current_date - interval '1 MONTH') -- not enough
data
)
```

Properties Dashboard SQL Statistics Dependents select_queries... Bank/postgres@PostgreSQL 16* Bank/postgres...

Bank/postgres@PostgreSQL 16

Query Query History

```
1 select em."TabelNumber", em."FullName", da."StartSum" as "AgreementStartSum"
2 from "bankDB"."Employee" em
3 join "bankDB"."DepositAgreement" da
4 on em."TabelNumber" = da."TabelNumber"
5 where da."StartSum" =
6 (select max(da."StartSum")
7 from "bankDB"."DepositAgreement" da
8 where da."DepositDate" > (current_date - interval '1 MONTH') -- not enough data
9 )
```

Data Output Notifications Messages

	TabelNumber integer	FullName character varying (50)	AgreementStartSum real
1	100020	Доронин Касьян Арсеньевич	300000

Активация Windows
Чтобы активировать Windows, перейдите в раздел
"Параметры".

Total rows: 1 of 1 Query complete 00:00:00.094 Ln 2, Col 28

Modifications

(Insert) Добавить договор на вклад вида «Умный» в Евро на сумму 300000, подписанный сегодня между сотрудником 100020 и клиентом с номером 324629513.

```
insert into "bankDB"."DepositAgreement"
("DepositDate", "PaymentDay", "PlanCloseDate", "StartSum", "TabelNumber",
"PassportNumber", "IdCurrency", "IdType")
values (current_date, extract(day from current_date),
       case
         when (select "Term" from "bankDB"."DepositType" where
"Name"='Умный') =
           12 then current_date + INTERVAL '1 YEAR'
         when (select "Term" from "bankDB"."DepositType" where
"Name"='Умный') =
           24 then current_date + INTERVAL '2 YEAR'
         end,
       300000, 100020, 324629513,
       (select "IdCurrency" from "bankDB"."Currency" where
"Name"='Евро'),
       (select "Id" from "bankDB"."DepositType" where "Name"='Умный'))
```

The screenshot shows a PostgreSQL client window titled "Bank/postgres@PostgreSQL 16". The query editor contains the SQL code for inserting a deposit agreement. The interface includes a toolbar with icons for file operations, query execution, and settings. Below the editor, the "Messages" tab is active, displaying the execution results: "INSERT 0 1" and "Query returned successfully in 191 msec." The status bar at the bottom indicates "Total rows: 1 of 1" and "Query complete 00:00:00.191". A Windows activation watermark is visible in the bottom right corner, and a green notification bubble states "Query returned successfully in 191 msec."

```
1 insert into "bankDB"."DepositAgreement"
2 ("DepositDate", "PaymentDay", "PlanCloseDate", "StartSum", "TabelNumber", "PassportNumber", "IdCurrency", "IdType")
3 values (current_date, extract(day from current_date),
4        case
5          when (select "Term" from "bankDB"."DepositType" where "Name"='Умный') =
6            12 then current_date + INTERVAL '1 YEAR'
7          when (select "Term" from "bankDB"."DepositType" where "Name"='Умный') =
8            24 then current_date + INTERVAL '2 YEAR'
9          end,
10         300000, 100020, 324629513,
11         (select "IdCurrency" from "bankDB"."Currency" where "Name"='Евро'),
12         (select "Id" from "bankDB"."DepositType" where "Name"='Умный'))
13
```

Bank/postgres@PostgreSQL 16

Query Query History Scratch Pad

1 insert into "bankDB"."DepositAgreement"
2 ("DepositDate", "PaymentDay", "PlanCloseDate", "StartSum", "TabelNumber", "PassportNumber", "IdCurrency", "IdType")
3 values (current_date, extract(day from current_date),
4 case
5 when (select "Term" from "bankDB"."DepositType" where "Name"='Умный') =
6 12 then current_date + INTERVAL '1 YEAR'
7 when (select "Term" from "bankDB"."DepositType" where "Name"='Умный') =
8 24 then current_date + INTERVAL '2 YEAR'
9 end,
10 300000, 100020, 324629513,
11 (select "IdCurrency" from "bankDB"."Currency" where "Name"='Евро'),
12 (select "Id" from "bankDB"."DepositType" where "Name"='Умный'))
13

Data Output Notifications Messages

INSERT 0 1

Query returned successfully in 191 msec.

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры"

✓ Query returned successfully in 191 msec. ✕

Total rows: 1 of 1 Query complete 00:00:00.191 Ln 12, Col 45

Bank/postgres@PostgreSQL 16
🔍
📁
🖨️
✎
⚙️
No limit
🏠
▶️
E
il.
🔄
🗑️
☰
?
↺

Query Query History
↗️

```

1 select * from "bankDB"."DepositAgreement"
    
```

Data Output Messages Notifications
↗️

	✎ AgreementNumber [PK] integer	✎ DepositDate date	✎ PaymentDay integer	✎ FactCloseDate date	✎ PlanCloseDate date	✎ Status character varying (10)	✎ StartSum real	✎ SummaryPayment real	✎ TabelNumber integer	✎ PassportNumber integer
1	21	2023-06-20	6	[null]	2024-06-20	Open	100000	0	100014	952442573
2	22	2023-08-13	20	[null]	2025-08-13	Open	202000	0	100020	123609496
3	23	2023-01-03	23	[null]	2025-01-03	Open	197000	0	100013	324629513
4	24	2023-05-28	24	[null]	2025-05-28	Open	273000	0	100015	157712532
5	25	2023-09-07	25	[null]	2025-09-07	Open	146000	0	100016	129798791
6	26	2023-07-17	22	[null]	2024-07-17	Open	277000	0	100016	189266391
7	27	2023-08-11	7	[null]	2024-08-11	Open	200000	0	100020	952442573
8	38	2023-11-11	11	[null]	2025-11-11	Open	300000	0	100020	324629513

Активация Windows

Чтобы активировать Windows, перейдите в раздел "Параметры".

Total rows: 8 of 8 Query complete 00:00:00.187
Ln 1, Col 42

(Update) Прописать на каждом договоре о вкладе суммарную выплату, которая накопила на сегодняшний день.

```
update "bankDB"."DepositAgreement" da
set "SummaryPayment" =
(select coalesce(sum("Payment"), 0)
 from "bankDB"."DepositPaySchedule" dps
 where not ("FactPaymentDate" is null)
 and da."AgreementNumber" = dps."AgreementNumber"
 order by 1 desc
 limit 1)
```

Bank/postgres@PostgreSQL 16

Query

Query History

```
1 -- update "bankDB"."DepositPaySchedule"
2 -- set "FactPaymentDate" = current_date
3 -- where "Id" < 367
4 -- or "Id" = 380;
5
6 update "bankDB"."DepositAgreement" da
7 set "SummaryPayment" =
8 (select coalesce(sum("Payment"), 0)
9  from "bankDB"."DepositPaySchedule" dps
10 where not ("FactPaymentDate" is null)
11 and da."AgreementNumber" = dps."AgreementNumber"
12 order by 1 desc
13 limit 1)
```

Data Output

Notifications

Messages

UPDATE 8

Query returned successfully in 118 msec.

Ln 6, Col 38

Successfully run. Total query runtime: 93 msec. 8 rows affected.

Bank/postgres@PostgreSQL 16

Query

Query History

```
1 select * from "bankDB"."DepositAgreement"
```

Data Output

Messages

Notifications

	AgreementNumber [PK] Integer	DepositDate date	PaymentDay Integer	FactCloseDate date	PlanCloseDate date	Status character varying (10)	StartSum real	SummaryPayment real	TabelNumber Integer	PassportNumber Integer
1	21	2023-06-20	6	[null]	2024-06-20	Open	100000	2353.83	100014	952442573
2	22	2023-08-13	20	[null]	2025-08-13	Open	202000	2123.04	100020	123609496
3	23	2023-01-03	23	[null]	2025-01-03	Open	197000	0	100013	324629513
4	24	2023-05-28	24	[null]	2025-05-28	Open	273000	0	100015	157712532
5	25	2023-09-07	25	[null]	2025-09-07	Open	146000	0	100016	129798791
6	26	2023-07-17	22	[null]	2024-07-17	Open	277000	0	100016	189266391
7	27	2023-08-11	7	[null]	2024-08-11	Open	200000	0	100020	952442573
8	38	2023-11-11	11	[null]	2025-11-11	Open	300000	0	100020	324629513

Ln 1, Col 42

Successfully run. Total query runtime: 93 msec. 8 rows affected.

(Delete) Удалить из расписания выплаты вклада все записи если статус 'Close'

```
delete from "bankDB"."DepositPaySchedule" dps
where 'Close' =
(select "Status"
from "bankDB"."DepositAgreement" da
where dps."AgreementNumber" = da."AgreementNumber");
```

The screenshot shows the PostgreSQL 16 query editor interface. The query editor contains the following SQL code:

```
1 -- insert into "bankDB"."DepositPaySchedule"
2 -- ("AgreementNumber", "PlanPaymentDate", "Payment")
3 -- values (39, current_date, 500)
4 -- update "bankDB"."DepositAgreement" da
5 -- set "Status"='Close'
6 -- where da."AgreementNumber" = 39
7
8 delete from "bankDB"."DepositPaySchedule" dps
9 where 'Close' = (
10 select "Status"
11 from "bankDB"."DepositAgreement" da
12 where dps."AgreementNumber" = da."AgreementNumber");
```

The query history shows the execution of the DELETE statement, indicating it was successful in 67 msec.

At the bottom, a status bar shows "Total rows: 1 of 1" and "Query complete 00:00:00.067".

Bank/postgres@PostgreSQL 16

No limit

Query Query History

```

1 -- select * from "bankDB"."DepositAgreement"
2 select * from "bankDB"."DepositPaySchedule"
```

Data Output Messages Notifications

	Id <small>[PK] bigint</small>	AgreementNumber <small>integer</small>	PlanPaymentDate <small>date</small>	Payment <small>real</small>	FactPaymentDate <small>date</small>
1	497	21	2023-07-06	583.33	[null]
2	498	21	2023-08-06	586.74	[null]
3	499	21	2023-09-06	590.16	[null]
4	500	21	2023-10-06	593.6	[null]
5	501	21	2023-11-06	597.06	[null]
6	502	21	2023-12-06	600.55	[null]
7	503	21	2024-01-06	604.05	[null]
8	504	21	2024-02-06	607.57	[null]
9	505	21	2024-03-06	611.12	[null]
10	506	21	2024-04-06	614.68	[null]
11	507	21	2024-05-06	618.27	[null]
12	508	21	2024-06-06	621.87	[null]

Активация Windows
Чтобы активировать Windows, перейдите в раздел «Параметры»

Total rows: 120 of 120 Query complete 00:00:00.176

View

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

```
drop view if exists "bankDB"."SignLoansEmployees";
create view "bankDB"."SignLoansEmployees" as
select em."TabelNumber", em."FullName", em."PassportNumber" as
"EmployeePassport",
em."Age", em."Phone", em."Address", em."IdCategory", la."AgreementNumber",
la."LoanDate", la."PaymentDay", la."FactCloseDate", la."PlanCloseDate",
la."Status", la."StartSum", la."Debt", la."IdCurrency", la."TypeId"
from "bankDB"."Employee" em
join "bankDB"."LoanAgreement" la
on em."TabelNumber" = la."TabelNumber"
where la."LoanDate" > (current_date - interval '1 MONTH') -- not enough data
order by la."LoanDate" desc;
```

Bank/postgres@PostgreSQL 16

Query Query History

```
1 drop view if exists "bankDB"."SignLoansEmployees";
2 create view "bankDB"."SignLoansEmployees" as
3 select em."TabelNumber", em."FullName", em."PassportNumber" as "EmployeePassport",
4 em."Age", em."Phone", em."Address", em."IdCategory", la."AgreementNumber",
5 la."LoanDate", la."PaymentDay", la."FactCloseDate", la."PlanCloseDate",
6 la."Status", la."StartSum", la."Debt", la."IdCurrency", la."TypeId"
7 from "bankDB"."Employee" em
8 join "bankDB"."LoanAgreement" la
9 on em."TabelNumber" = la."TabelNumber"
10 -- where la."LoanDate" > (current_date - interval '1 MONTH') -- not enough data
11 order by la."LoanDate" desc;
```

Scratch Pad

Data Output Notifications Messages

CREATE VIEW

Query returned successfully in 59 msec.

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

✓ Query returned successfully in 59 msec. ✕

✓ File saved successfully. ✕

Total rows: 1 of 1 Query complete 00:00:00.059 Ln 4, Col 41

2. Найти клиентов банка, имеющих задолженности по кредитам.

```
drop view if exists "bankDB"."NotPaidClients";
create view "bankDB"."NotPaidClients" as
select distinct cl.*
from "bankDB"."Client" cl
join "bankDB"."LoanAgreement" la
on cl."PassportNumber"=la."PassportNumber"
join "bankDB"."LoanPaySchedule" lps
on la."AgreementNumber"=lps."AgreementNumber"
where (lps."FactPaymentDay" is null and lps."PlanPaymentDate" <
current date);
```

Bank/postgres@PostgreSQL 16

Query

Query History

1 drop view if exists "bankDB"."NotPaidClients";
2 create view "bankDB"."NotPaidClients" as
3 select distinct cl.*
4 from "bankDB"."Client" cl
5 join "bankDB"."LoanAgreement" la
6 on cl."PassportNumber"=la."PassportNumber"
7 join "bankDB"."LoanPaySchedule" lps
8 on la."AgreementNumber"=lps."AgreementNumber"
9 where (lps."FactPaymentDay" is null and lps."PlanPaymentDate" < current_date);
10

Data Output

Notifications

Messages

CREATE VIEW

Query returned successfully in 62 msec.

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

✓ Query returned successfully in 62 msec. ✕

Total rows: 1 of 1 Query complete 00:00:00.062 Ln 10, Col 1

Bank/postgres@PostgreSQL 16

Query

Query History

1 select * from "bankDB"."NotPaidClients"
2 -- all fact dates are null so system thinks everyone didn't pay

Data Output

Messages

Notifications

	PassportNumber bigint	FullName character varying (50)	Address character varying (150)	Phone bigint	E-mail character varying (30)
1	324629513	Владимирова Ольга Романовна	к. Старая Русса, алл. Приморская, д. 39 стр. 2/5, 5001...	80745955216	kuprijan_23@mail.ru
2	123609496	Лука Вячеславович Наумов	д. Карачаевск, ш. Астраханское, д. 9/8 к. 522, 901051	72381924356	seliverst_1983@yandex.ru
3	571140644	Оксана Аркадьевна Соловьева	д. Хасавюрт, ш. Станционное, д. 4, 640885	80209282910	makar24@hotmail.com
4	129798791	Носов Анисим Владиславович	г. Валаам, бул. Баумана, д. 9/8 стр. 60, 046782	71400117644	tkonovalev@hotmail.com
5	121748022	Галина Святославовна Некрасова	п. Усть-Катав, бул. Кочубея, д. 606 к. 3/5, 846641	77476804830	isidor69@gmail.com
6	51690385	Фадеев Фока Яковлевич	ст. Ногинск (Моск.), пр. Щорса, д. 813 стр. 5, 756425	89348592289	vorobevelzar@hotmail.com

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Total rows: 6 of 6 Query complete 00:00:00.094 Ln 2, Col 64

Index

1. Простые индексы

Сделаем запрос, который собирает инфу по всем договорам, клиентам и сотрудникам, их подписавших без индексов:

Bank/postgres@PostgreSQL 16

```

1 -- w/o index
2
3 select cl."PassportNumber", cl."FullName", em."TabelNumber", da."AgreementNumber", 'Deposit' as type
4 from "bankDB"."Client" cl, "bankDB"."Employee" em, "bankDB"."DepositAgreement" da
5 where cl."PassportNumber" = da."PassportNumber"
6 and em."TabelNumber" = da."TabelNumber"
7 union
8 select cl."PassportNumber", cl."FullName", em."TabelNumber", la."AgreementNumber", 'Loan' as type
9 from "bankDB"."Client" cl, "bankDB"."Employee" em, "bankDB"."LoanAgreement" la
10 where cl."PassportNumber" = la."PassportNumber"
11 and em."TabelNumber" = la."TabelNumber"
12
13 order by type

```

	PassportNumber bigint	FullName character varying (50)	TabelNumber integer	AgreementNumber integer	type text
1	324629513	Владими́рова О́льга Рома́новна	100020	39	Deposit
2	129798791	Носов Ани́сим Влади́славович	100016	25	Deposit
3	324629513	Владими́рова О́льга Рома́новна	100013	23	Deposit
4	123609496	Лука Вячесла́вович Нау́мов	100020	22	Deposit
5	952442573	Ираклі́й Елисе́евич Мака́ров	100014	21	Deposit
6	189266391	Крюкова́ Елизавета́ Бори́совна	100016	26	Deposit
7	157712532	Лихачев Ха́ритон Богода́нович	100015	24	Deposit
8	952442573	Ираклі́й Елисе́евич Мака́ров	100020	27	Deposit
9	571140644	Оксана́ Аркадьёвна Соловьёва	100017	8	Loan
10	123609496	Лука Вячесла́вович Нау́мов	100015	7	Loan

Total rows: 14 of 14 Query complete 00:00:00.157

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Пропишем индексы на первичные ключи таблиц, которые использовались в запросе

Bank/postgres@PostgreSQL 16

Query | Query History

```

1 create index if not exists idx_client_passport on "bankDB"."Client" ("PassportNumber");
2
3 create index if not exists idx_employee_number on "bankDB"."Employee" ("TabelNumber");
4
5 create index if not exists idx_deposit_number on "bankDB"."DepositAgreement" ("AgreementNumber");
6
7 create index if not exists idx_loan_number on "bankDB"."LoanAgreement" ("AgreementNumber");
8
9

```

Scratch Pad

Data Output | Messages | Notifications

CREATE INDEX

Query returned successfully in 50 msec.

Активация Windows
Чтобы активировать Windows, перейдите в раздел
"Параметры"

✓ Query returned successfully in 50 msec. ✕

Total rows: 0 of 0 | Query complete 00:00:00.050 | Ln 8, Col 1

Активация Windows

✓ Query returned successfully in 50 msec. ✕

Bank/postgres@PostgreSQL 16

Query Query History Scratch Pad

```

1  -- index
2
3  select cl."PassportNumber", cl."FullName", em."TabelNumber", da."AgreementNumber", 'Deposit' as type
4  from "bankDB"."Client" cl, "bankDB"."Employee" em, "bankDB"."DepositAgreement" da
5  where cl."PassportNumber" = da."PassportNumber"
6  and em."TabelNumber" = da."TabelNumber"
7  union
8  select cl."PassportNumber", cl."FullName", em."TabelNumber", la."AgreementNumber", 'Loan' as type
9  from "bankDB"."Client" cl, "bankDB"."Employee" em, "bankDB"."LoanAgreement" la
10 where cl."PassportNumber" = la."PassportNumber"
11 and em."TabelNumber" = la."TabelNumber"
12
13 order by type

```

Data Output Notifications Messages

	PassportNumber bigint	FullName character varying (50)	TabelNumber integer	AgreementNumber integer	type text
1	324629513	Владими́рова О́льга Рома́новна	100020	39	Deposit
2	129798791	Носов А́нисим Влади́славович	100016	25	Deposit
3	324629513	Владими́рова О́льга Рома́новна	100013	23	Deposit
4	123609496	Лука Вя́чеславович Нау́мов	100020	22	Deposit
5	952442573	Ира́клий Ели́сеевич Мака́ров	100014	21	Deposit
6	189266391	Крю́кова Ели́завета Бори́совна	100016	26	Deposit
7	157712532	Лиха́чев Ха́ритон Бо́гданович	100015	24	Deposit
8	952442573	Ира́клий Ели́сеевич Мака́ров	100020	27	Deposit
9	571140644	Окса́на А́ркадьевна Соловье́ва	100017	8	Loan
10	123609496	Лука Вя́чеславович Нау́мов	100015	7	Loan

Total rows: 14 of 14 Query complete 00:00:00.079 Ln 1, Col 4

Data Output Messages Explain Notifications

Graphical Analysis Statistics

```

graph LR
    subgraph "bankDB"
        Client[Client]
        Employee[Employee]
        DepositAgreement[DepositAgreement]
        LoanAgreement[LoanAgreement]
    end
    Client --> H1[Hash Inner Join]
    Employee --> H1
    DepositAgreement --> H1
    LoanAgreement --> H2[Hash Inner Join]
    Employee --> H2
    H1 --> H3[Hash Inner Join]
    H2 --> H3
    H3 --> Append[Append]
    Append --> Aggregate[Aggregate]
    Aggregate --> Sort[Sort]

```

Total rows: 1 of 1 Query complete 00:00:00.096 Ln 1, Col 4

Напишем запрос, который выдаёт нам все невыплаченные на данный день платежи по кредиту

Активация Windows

Чтобы активировать Windows, перейдите в раздел "Параметры".

Bank/postgres@PostgreSQL 16

Query Query History Scratch Pad

```

1  -- index
2
3  select cl."PassportNumber", cl."FullName", em."TabelNumber", la."AgreementNumber",
4  lps."DebtPayment", lps."InterestPayment"
5  from "bankDB"."LoanAgreement" la
6  join "bankDB"."Employee" em
7  on em."TabelNumber" = la."TabelNumber"
8  join "bankDB"."Client" cl
9  on cl."PassportNumber" = la."PassportNumber"
10 join "bankDB"."LoanPaySchedule" lps
11 on lps."AgreementNumber" = la."AgreementNumber"
12 where lps."FactPaymentDay" is null
13
14 order by "AgreementNumber", "InterestPayment"

```

Data Output Notifications Messages

	PassportNumber bigint	FullName character varying (50)	TabelNumber integer	AgreementNumber integer	DebtPayment real	InterestPayment real
1	123609496	Лука Вячеславович Наумов	100015	7	6416.67	42.78
2	123609496	Лука Вячеславович Наумов	100015	7	6416.67	85.56
3	123609496	Лука Вячеславович Наумов	100015	7	6416.67	128.33
4	123609496	Лука Вячеславович Наумов	100015	7	6416.67	171.11
5	123609496	Лука Вячеславович Наумов	100015	7	6416.67	213.89
6	123609496	Лука Вячеславович Наумов	100015	7	6416.67	256.67
7	123609496	Лука Вячеславович Наумов	100015	7	6416.67	299.44
8	123609496	Лука Вячеславович Наумов	100015	7	6416.67	342.22
9	123609496	Лука Вячеславович Наумов	100015	7	6416.67	385

Total rows: 132 of 132 Query complete 00:00:00.088 Ln 6, Col 28

В конце дропаем все индексы

Bank/postgres@PostgreSQL 16

Query Query History Scratch Pad

```

1  drop index if exists "bankDB"."idx_client_passport";
2
3  drop index if exists "bankDB"."idx_employee_number";
4
5  drop index if exists "bankDB"."idx_deposit_number";
6
7  drop index if exists "bankDB"."idx_loan_number";
8
9  drop index if exists "bankDB"."idx_loan_number_passport_tabel";
10
11 drop index if exists "bankDB"."idx_loan_schedule_payment";

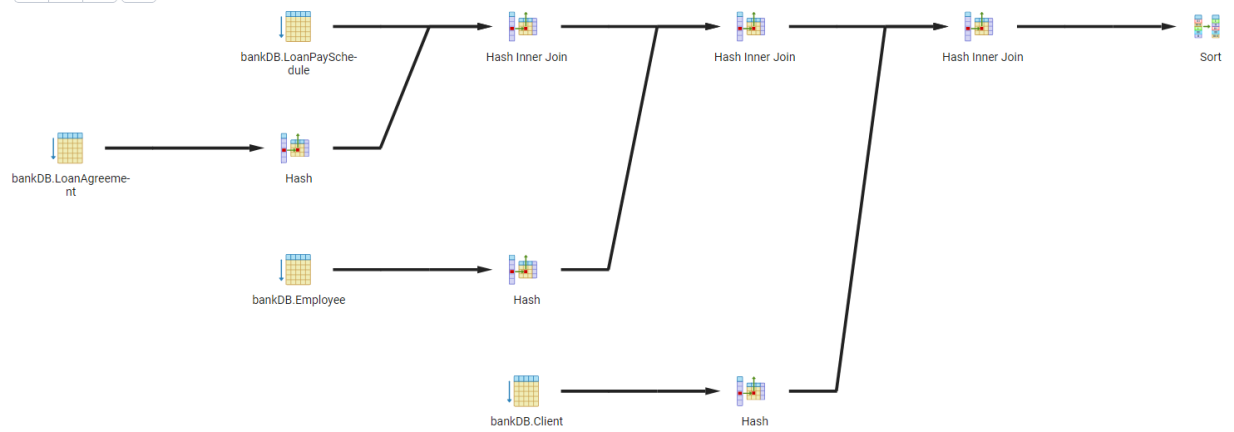
```

Data Output Messages Notifications

DROP INDEX

Query returned successfully in 50 msec.

Total rows: 0 of 0 Query complete 00:00:00.050 Ln 11, Col 59



Активация Windows

Чтобы активировать Windows, перейдите в раздел "Параметры".

Вывод

В ходе лабораторной работы были изучены возможности написания различных запросов в PostgreSQL. В процессе лабораторной работы были написаны, в соответствии с индивидуальным заданием запросы на выбор данных из таблицы и создание представлений. Были написаны запросы на модификацию данных с помощью подзапросов, что позволяли сделать эти запросы более автоматическими. Напоследок, были написаны простые и составные индексы для таблиц с данными и произведено сравнение времени работы select запроса с индексами и без них. Время с индексами оказалось меньше, чем без них.