

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

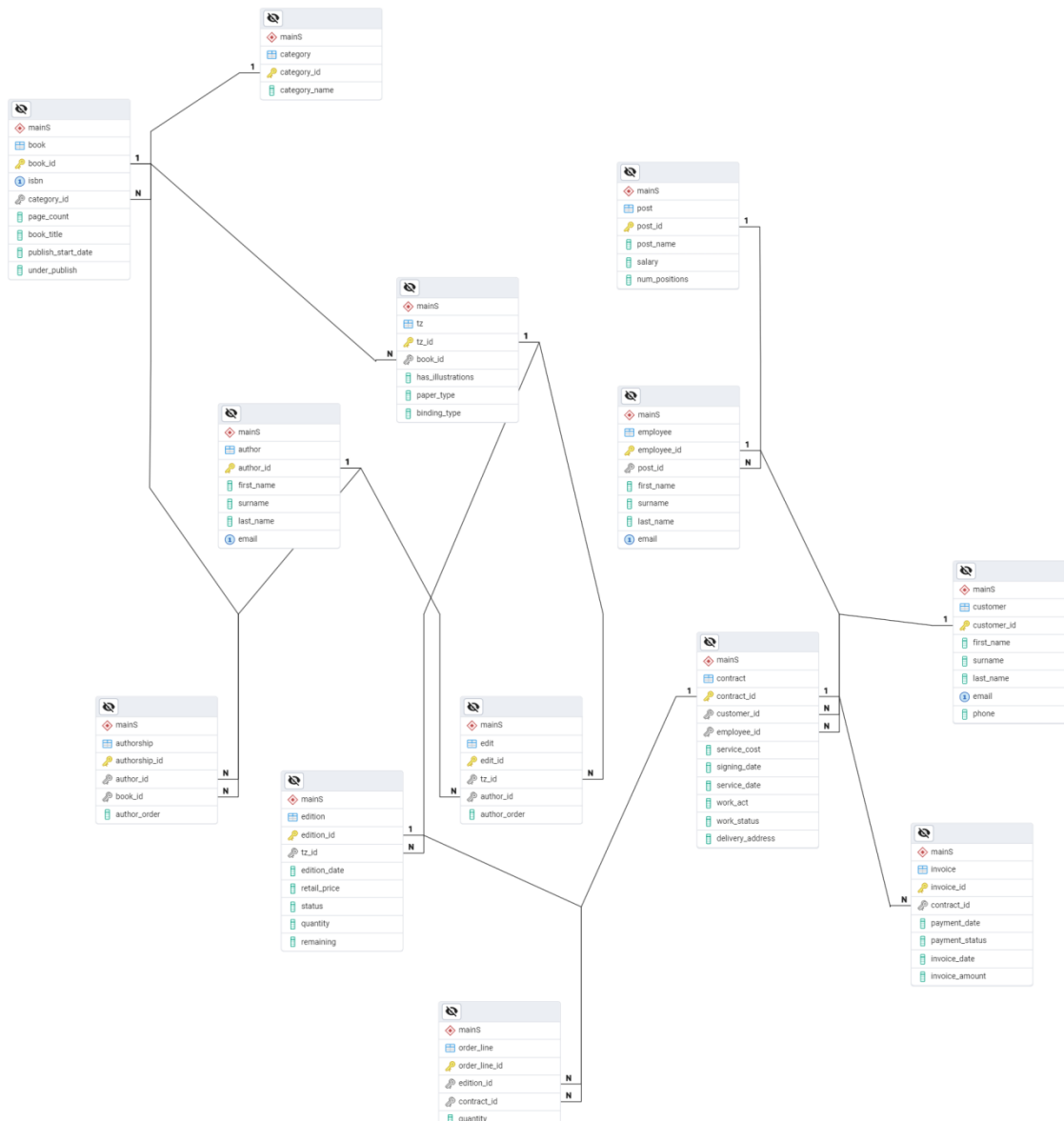
Санкт-Петербург
2024/2025

Цель работы: овладеть практическими навыками создания 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 лабораторной работы №2, часть 2 и 3).
2. Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов.
3. Изучить графическое представление запросов и 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.

Generated ERD diagram:



Задание по варианту:

Задание 2. Создать запросы:

- Список книг, изданных в текущем году и относящихся к категории «Базы данных».

SQL-команда:

```
SELECT * FROM "mainS".book
JOIN "mainS".category ON book.category_id=category.category_id
JOIN "mainS".tz ON tz.book_id=book.book_id
JOIN "mainS".edition ON edition.tz_id=tz.tz_id
WHERE category_name = 'Базы данных' AND edition_date > '2025-01-01' AND status IN
('в продаже', 'распродан');
```

	book_id integer	isbn character varying (100)	category_id integer	page_count integer	book_title character varying (50)	publish_start_date date	under_publish boolean	category_id integer	category_na character va
1	11	978-5-93700-287-7	6	520	Путеводитель по базам данных	2025-01-02	true	6	Базы данн

- Список покупателей, заказавших книг на сумму, превышающую среднюю сумму заказа за последний календарный год.

SQL-команда:

```
SELECT DISTINCT c.customer_id,
       c.first_name,
       c.surname,
       c.last_name,
       SUM(i.invoice_amount) AS total_spent
FROM "mainS".customer c
JOIN "mainS".contract ct ON c.customer_id = ct.customer_id
JOIN "mainS".invoice i ON ct.contract_id = i.contract_id
WHERE i.invoice_date BETWEEN '2023-01-01' AND '2024-01-01'
GROUP BY c.customer_id, c.first_name, c.surname, c.last_name
HAVING SUM(i.invoice_amount) >
(
  SELECT AVG(total_amount)
  FROM (
    SELECT SUM(i2.invoice_amount) AS total_amount
    FROM "mainS".invoice i2
    WHERE i2.invoice_date BETWEEN '2023-01-01' AND '2024-01-01'
    GROUP BY i2.contract_id
  ) AS subquery
)
ORDER BY total_spent DESC;
```

Data Output Messages Notifications					
	customer_id [PK] integer	first_name character varying (25)	surname character varying (25)	last_name character varying (25)	total_spent bigint
1	5	AO	Программист	[null]	110000
2	1	000	ИнфоТех	[null]	100000

- Список книг, которые не заказывались в течение последних двух кварталов.

SQL-команда:

```
SELECT b.book_id, b.book_title, b.ISBN
FROM "mainS".book b
WHERE b.book_id NOT IN (
    SELECT DISTINCT b2.book_id
    FROM "mainS".book b2
    JOIN "mainS".tz t ON b2.book_id = t.book_id
    JOIN "mainS".edition e ON t.tz_id = e.tz_id
    JOIN "mainS".order_line ol ON e.edition_id = ol.edition_id
    JOIN "mainS".contract c ON ol.contract_id = c.contract_id
    WHERE c.signing_date >= DATE_TRUNC('quarter', CURRENT_DATE) -
    INTERVAL '6 months')
ORDER BY b.book_title;
```

	book_id [PK] integer	book_title character varying (50)	isbn character varying (100)
1	6	Python для всех	978-0-596-15806-4
2	9	Архитектура UNIX	978-0-13-110362-8
3	1	Искусство программирования	978-5-907144-23-1
4	2	Операционные системы	978-5-4461-1155-8
5	8	Программирование на языке C	978-0-13-142901-7
6	11	Путеводитель по базам данных	978-5-93700-287-7
7	10	Свобода программирования	978-0-13-409341-3
8	5	Чистый код	978-5-4461-0960-9
9	4	Шаблоны корпоративных приложений	978-5-6041394-0-0
10	7	Язык программирования C	978-0-13-110362-7
11	3	Язык программирования C++	978-5-6045724-6-7

- Список авторов, не написавших ни одной книги, относящейся к категории “Языки программирования”.

SQL-команда:

```
SELECT a.author_id, a.first_name, a.surname
FROM "mainS".author a
WHERE a.author_id NOT IN (
    SELECT DISTINCT auth.author_id
    FROM "mainS".author auth
    JOIN "mainS".authorship au ON auth.author_id = au.author_id
    JOIN "mainS".book b ON au.book_id = b.book_id
    JOIN "mainS".category c ON b.category_id = c.category_id
    WHERE c.category_name = 'Языки программирования'
);
```

Data Output Messages Notifications			
	author_id [PK] integer	first_name character varying (25)	surname character varying (25)
1	1	Дональд	Кнут
2	2	Эндрю	Таненбаум
3	3	Бьерн	Страуструп
4	4	Мартин	Фаулер
5	5	Роберт	Мартин
6	6	Гвидо	ван Россум
7	7	Кен	Томпсон
8	8	Деннис	Ритчи
9	9	Линус	Торвальдс
10	10	Ричард	Столлман
11	11	Владимир	Комаров

- Список книг, в названиях которых содержится слово “проектирование” и которые присутствуют на базе в количестве, превышающем 50 экземпляров.

SQL-команда:

```
SELECT b.book_id, b.book_title, b.ISBN, SUM(ed.remaining) as tottal_remaning
FROM "mainS".book b
JOIN "mainS".tz ON tz.book_id = b.book_id
JOIN "mainS".edition ed ON ed.tz_id = tz.tz_id
WHERE b.book_title LIKE '%проектирование%' AND ed.status = 'в продаже' AND
ed.remaining > 50
GROUP BY b.book_id, b.book_title, b.ISBN;
```

Data Output Messages Notifications			
	book_id [PK] integer	book_title character varying (50)	isbn character varying (100)
			total_remaining bigint

- Покупателя, сделавшего заказ на максимальную сумму за последний месяц.

SQL-команда:

```
SELECT c.customer_id, c.first_name, c.surname, co.service_cost , co.signing_date
FROM "mainS".customer c
JOIN "mainS".contract co ON c.customer_id = co.customer_id
WHERE co.signing_date >= CURRENT_DATE - INTERVAL '1 month'
ORDER BY co.service_cost DESC;
```

Data Output Messages Notifications					
	customer_id integer	first_name character varying (25)	surname character varying (25)	service_cost integer	signing_date date
1	2	АО	АйТи Солюшнс	340000	2025-04-01

- Список книг, не попавших ни в один из заказов в течение последнего года.

SQL-команда:

```
SELECT b.book_id, b.book_title, b.ISBN
FROM "mainS".book b
WHERE b.book_id NOT IN (
    SELECT DISTINCT b2.book_id
    FROM "mainS".book b2
    JOIN "mainS".tz t ON b2.book_id = t.book_id
    JOIN "mainS".edition e ON t.tz_id = e.tz_id
    JOIN "mainS".order_line ol ON e.edition_id = ol.edition_id
    JOIN "mainS".contract co ON ol.contract_id = co.contract_id
    WHERE co.signing_date >= CURRENT_DATE - INTERVAL '1 year'
);
```

Data Output Messages Notifications			
	book_id [PK] integer	book_title character varying (50)	isbn character varying (100)
1	1	Искусство программирования	978-5-907144-23-1
2	2	Операционные системы	978-5-4461-1155-8
3	3	Язык программирования C++	978-5-6045724-6-7
4	4	Шаблоны корпоративных приложений	978-5-6041394-0-0
5	5	Чистый код	978-5-4461-0960-9
6	6	Python для всех	978-0-596-15806-4
7	7	Язык программирования C	978-0-13-110362-7
8	8	Программирование на языке C	978-0-13-142901-7
9	9	Архитектура UNIX	978-0-13-110362-8
10	10	Свобода программирования	978-0-13-409341-3
11	11	Путеводитель по базам данных	978-5-93700-287-7

Задание 3. Создать представления:

- содержащее сведения о количестве заказанных экземпляров каждой книги, изданной в текущем году;

SQL-команда:

```
CREATE OR REPLACE VIEW "mainS".current_year_orders AS
SELECT b.book_id, b.book_title, b.ISBN, SUM(ol.quantity) AS total_ordered,
COUNT(DISTINCT co.contract_id) AS order_count
FROM "mainS".book b
JOIN "mainS".tz t ON b.book_id = t.book_id
JOIN "mainS".edition e ON t.tz_id = e.tz_id
JOIN "mainS".order_line ol ON e.edition_id = ol.edition_id
JOIN "mainS".contract co ON ol.contract_id = co.contract_id
WHERE EXTRACT(YEAR FROM e.edition_date) = EXTRACT(YEAR FROM CURRENT_DATE)
GROUP BY b.book_id, b.book_title, b.ISBN
```

ORDER BY total_ordered DESC;

```
139
140 SELECT * FROM "mainS".current_year_orders;|
```

Data Output Messages Notifications

SQL

	book_id integer	book_title character varying (50)	isbn character varying (100)	total_ordered bigint	order_count bigint
1	5	Чистый код	978-5-4461-0960-9	50	1
2	2	Операционные системы	978-5-4461-1155-8	40	1
3	4	Шаблоны корпоративных приложений	978-5-6041394-0-0	35	1
4	1	Искусство программирования	978-5-907144-23-1	30	1
5	3	Язык программирования C++	978-5-6045724-6-7	20	1

- количество заказов по покупателям за последний год.

SQL-команда:

```
CREATE OR REPLACE VIEW "mainS".customer_orders_last_year AS
SELECT c.customer_id, c.first_name || ' ' || c.surname AS customer_name,
       COUNT(co.contract_id) AS order_count,
       SUM(co.service_cost) AS total_spent
FROM "mainS".customer c
JOIN "mainS".contract co ON c.customer_id = co.customer_id
WHERE co.signing_date >= CURRENT_DATE - INTERVAL '1 year'
GROUP BY c.customer_id, customer_name
ORDER BY order_count DESC, total_spent DESC;
```

```
154 SELECT * FROM "mainS".customer_orders_last_year;
```

Data Output Messages Notifications

SQL

	customer_id integer	customer_name text	order_count bigint	total_spent bigint
1	2	АО АйТи Солюшнс	1	340000

Составить 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов:

INSERT запрос: Вставить работника с должностью Редактор.
До

	employee_id [PK] integer	post_id integer	first_name character varying (25)	surname character varying (25)	last_name character varying (25)	email character varying (50)
1	21	1	Сергей	Морозов	Владимирович	sergey.morozov@mail.com
2	22	2	Ирина	Новикова	Анатолевна	irina.novikova@mail.com
3	23	3	Олег	Семенов	Григорьевич	oleg.semenov@mail.com
4	24	4	Екатерина	Орлова	Дмитриевна	ekaterina.orlova@mail.com
5	25	5	Роман	Макаров	Александрович	roman.makarov@mail.com
6	26	1	Анастасия	Зуева	Олеговна	anastasia.zueva@mail.com
7	27	2	Геннадий	Титов	Иванович	gennady.titov@mail.com
8	28	3	Людмила	Киселева	Сергеевна	lyudmila.kiseleva@mail.com
9	29	4	Владимир	Фролов	Васильевич	vladimir.frolov@mail.com
10	30	5	Светлана	Соловьева	Петровна	svetlana.solovieva@mail.com

SQL-команда:

```
INSERT INTO "mainS".employee (post_id, first_name, surname, last_name, email)
VALUES (
  (SELECT post_id FROM "mainS".post WHERE post_name = 'Редактор'),
  'Иван',
  'Петров',
  'Игоревич',
  'ivan.petrov@example.com'
);
```

После

	employee_id [PK] integer	post_id integer	first_name character varying (25)	surname character varying (25)	last_name character varying (25)	email character varying (50)
1	21	1	Сергей	Морозов	Владимирович	sergey.morozov@mail.com
2	22	2	Ирина	Новикова	Анатолевна	irina.novikova@mail.com
3	23	3	Олег	Семенов	Григорьевич	oleg.semenov@mail.com
4	24	4	Екатерина	Орлова	Дмитриевна	ekaterina.orlova@mail.com
5	25	5	Роман	Макаров	Александрович	roman.makarov@mail.com
6	26	1	Анастасия	Зуева	Олеговна	anastasia.zueva@mail.com
7	27	2	Геннадий	Титов	Иванович	gennady.titov@mail.com
8	28	3	Людмила	Киселева	Сергеевна	lyudmila.kiseleva@mail.com
9	29	4	Владимир	Фролов	Васильевич	vladimir.frolov@mail.com
10	30	5	Светлана	Соловьева	Петровна	svetlana.solovieva@mail.com
11	31	1	Иван	Петров	Игоревич	ivan.petrov@example.com

UPDATE запрос: Увеличить зарплату на 1.5 у Редакторов и программистов.
До

	post_id [PK] integer	post_name character varying (100)	salary integer	num_positions integer
1	1	Редактор	70000	5
2	2	Менеджер по продажам	80000	3
3	3	Технический писатель	75000	4
4	4	Программист	90000	2
5	5	Графический дизайнер	65000	3

SQL-команда:

```
UPDATE "mainS".post p
SET salary = salary * 1.15
```



```

WHERE p.post_name IN (
    SELECT post_name
    FROM "mainS".post pt
    WHERE pt.post_name IN ('Редактор', 'Программист')
);

```

После

	post_id [PK] integer	post_name character varying (100)	salary integer	num_positions integer
1	2	Менеджер по продажам	80000	3
2	3	Технический писатель	75000	4
3	5	Графический дизайнер	65000	3
4	1	Редактор	80500	5
5	4	Программист	103500	2

DELETE запрос: Удалить все контракты (и счета), которые были подписаны больше 10 лет назад.

До

	invoice_id [PK] integer	contract_id integer	payment_date date	payment_status boolean	invoice_date date	invoice_amount integer
1	8	10	2023-06-10	true	2023-06-01	50000
2	9	10	2023-09-20	true	2023-09-15	50000
3	10	11	2023-07-15	true	2023-07-10	40000
4	11	12	2023-10-30	false	2023-10-20	45000
5	12	13	2023-07-10	true	2023-07-01	70000
6	13	14	2023-06-15	false	2023-06-03	90000
7	14	14	2023-08-13	true	2023-08-02	20000

SQL-команда:

```

-- Удаляем все счета по старым контрактам
DELETE FROM "mainS".invoice
WHERE contract_id IN (
    SELECT contract_id
    FROM "mainS".contract
    WHERE signing_date < CURRENT_DATE - INTERVAL '10 years'
);

```

```

-- Удаляем сами контракты
DELETE FROM "mainS".contract
WHERE signing_date < CURRENT_DATE - INTERVAL '10 years';

```

После

	invoice_id [PK] integer	contract_id integer	payment_date date	payment_status boolean	invoice_date date	invoice_amount integer
1	8	10	2023-06-10	true	2023-06-01	50000
2	9	10	2023-09-20	true	2023-09-15	50000
3	10	11	2023-07-15	true	2023-07-10	40000
4	11	12	2023-10-30	false	2023-10-20	45000
5	12	13	2023-07-10	true	2023-07-01	70000
6	13	14	2023-06-15	false	2023-06-03	90000
7	14	14	2023-08-13	true	2023-08-02	20000

Ничего не изменилось, потому что нет таких старых заказов.

Изучить графическое представление запросов и просмотреть историю запросов

Графическое представление удаление счетов с подзапросом:

```

178 select * from "mainS".invoice;
179 -- Сначала удаляем все счета по старым контрактам
180 DELETE FROM "mainS".invoice
181 WHERE contract_id IN (
182     SELECT contract_id
183     FROM "mainS".contract
184     WHERE signing_date < CURRENT_DATE - INTERVAL '10 years'
185 );
186
187 -- Затем удаляем сами контракты

```

Data Output Messages Explain x Notifications

Graphical Analysis Statistics

invoice

contract

Hash

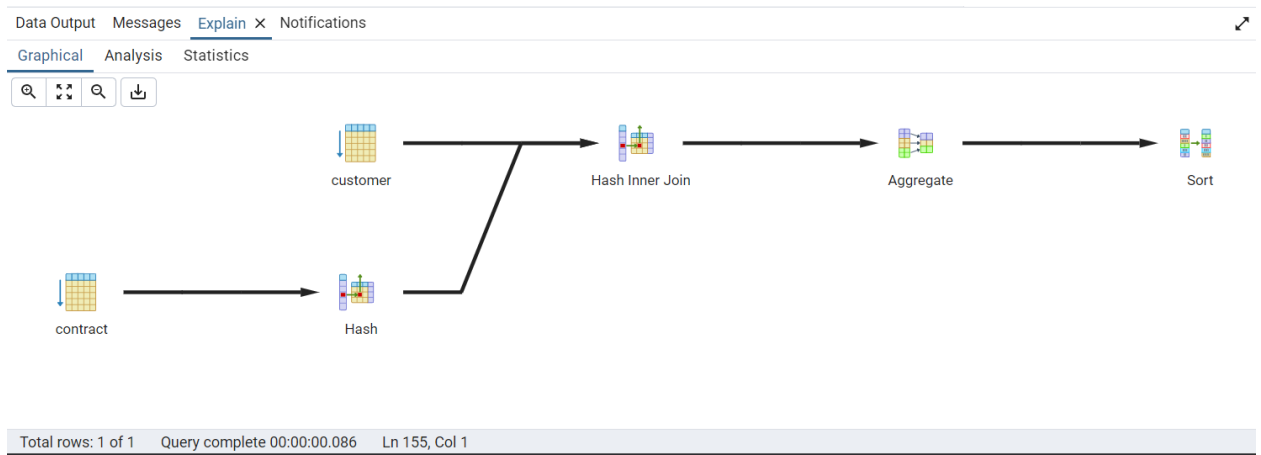
Hash Inner Join

Delete

Total rows: 1 of 1 Query complete 00:00:00.082 Ln 182, Col 23

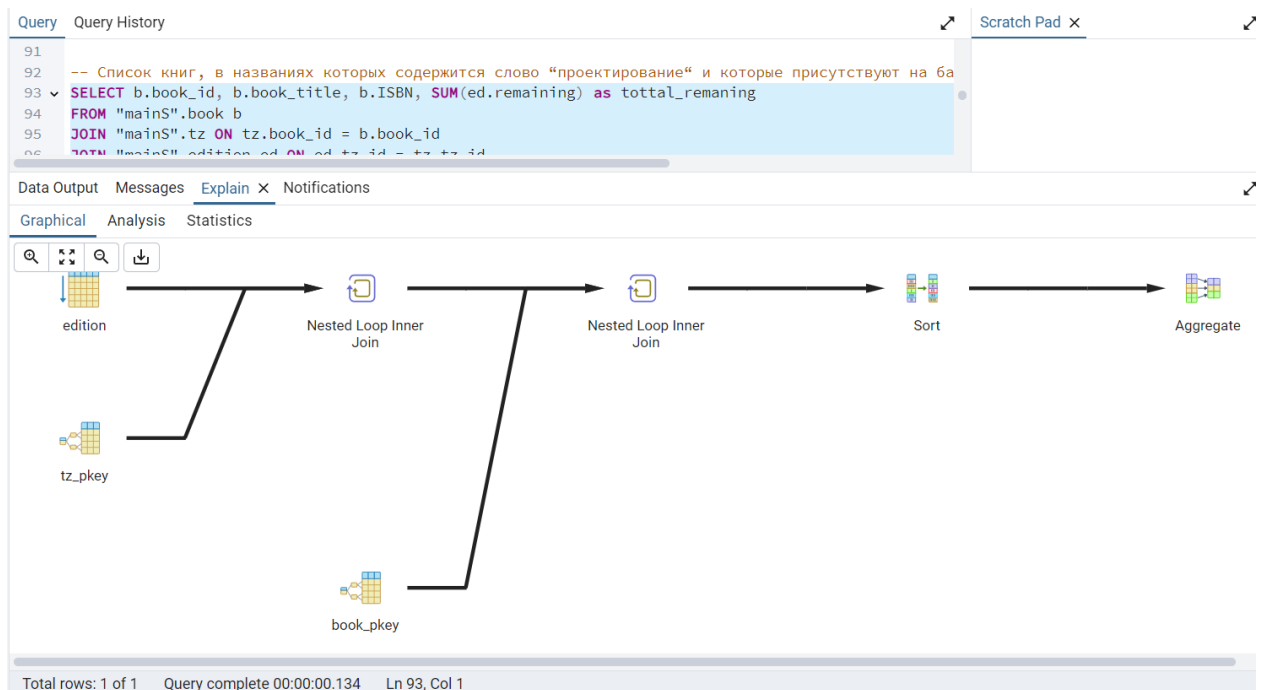
Запрос выполнен за 82 секунды с использованием хеш-соединения.

Графическое представление вызова представления количество заказов по покупателям за последний год.



Запрос выполнен за 86 секунд с использованием hash таблиц для соединения.

Графическое представление запроса на нахождение списка книг, в названиях которых содержится слово “проектирование” и которые присутствуют на базе в количестве, превышающем 50 экземпляров.

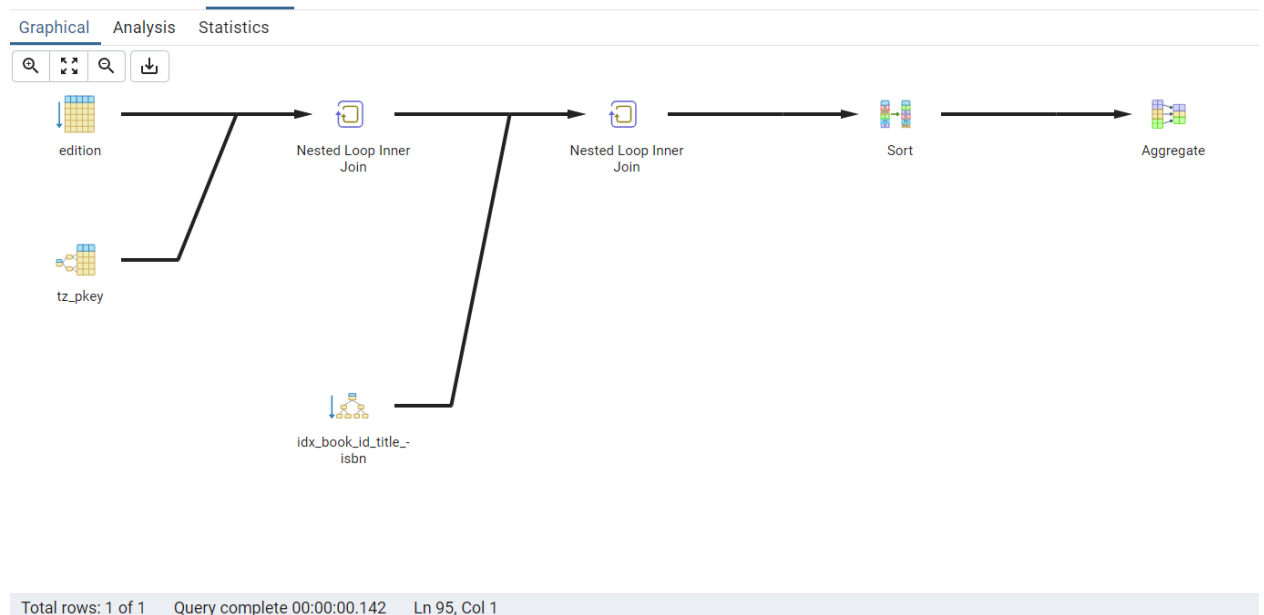


Запрос выполнен за 134 секунды с использованием Nested Loop Inner Join, так как таблицы маленькие.

Вот так выглядит история запросов:

Data Output	Messages	Explain	×	Notifications
<div> <div> <div>+</div> <div>SQL</div> </div> <div> <div>SQL</div> <div>SQL</div> </div> </div>				
<div> <div>QUERY PLAN</div> <div>text</div> <div> <div>1</div> <div>GroupAggregate (cost=29.55..29.57 rows=1 width=348)</div> </div> <div> <div>2</div> <div>Group Key: b.book_id</div> </div> <div> <div>3</div> <div>-> Sort (cost=29.55..29.55 rows=1 width=344)</div> </div> <div> <div>4</div> <div>Sort Key: b.book_id</div> </div> <div> <div>5</div> <div>-> Nested Loop (cost=0.28..29.54 rows=1 width=344)</div> </div> <div> <div>6</div> <div>-> Nested Loop (cost=0.15..29.30 rows=1 width=8)</div> </div> <div> <div>7</div> <div>-> Seq Scan on edition ed (cost=0.00..21.10 rows=1 width=8)</div> </div> <div> <div>8</div> <div>Filter: ((remaining > 50) AND ((status)::text = 'в продаже'::text))</div> </div> <div> <div>9</div> <div>-> Index Scan using tz_pkey on tz (cost=0.15..8.17 rows=1 width=8)</div> </div> <div> <div>10</div> <div>Index Cond: (tz_id = ed.tz_id)</div> </div> <div> <div>11</div> <div>-> Index <u>Only Scan using idx_book_id_title_isbn on book b</u> (cost=0.14..0.18 rows=1 width=3...</div> </div> <div> <div>12</div> <div>Index Cond: (book_id = tz.book_id)</div> </div> <div> <div>13</div> <div>Filter: ((book_title)::text ~~ '%проектирование%'::text)</div> </div> </div>				

В графическом представлении также видим использование индекса. Время увеличилось до 142 секунд, так как таблицы маленькие.



Создадим 2 простых индекса для запроса на удаление счетов с подзапросом на столбец *signing_date* из таблицы *contract* и столбец *contract_id* из таблицы *invoice*. SQL команда:

```
CREATE INDEX idx_contract_signing_date ON "mainS".contract(signing_date);
CREATE INDEX idx_invoice_contract_id ON "mainS".invoice(contract_id);
```

Выполнение запроса после создания индекса. Используя команду EXPLAIN можно увидеть, что запрос будет использовать только что созданные индексы:

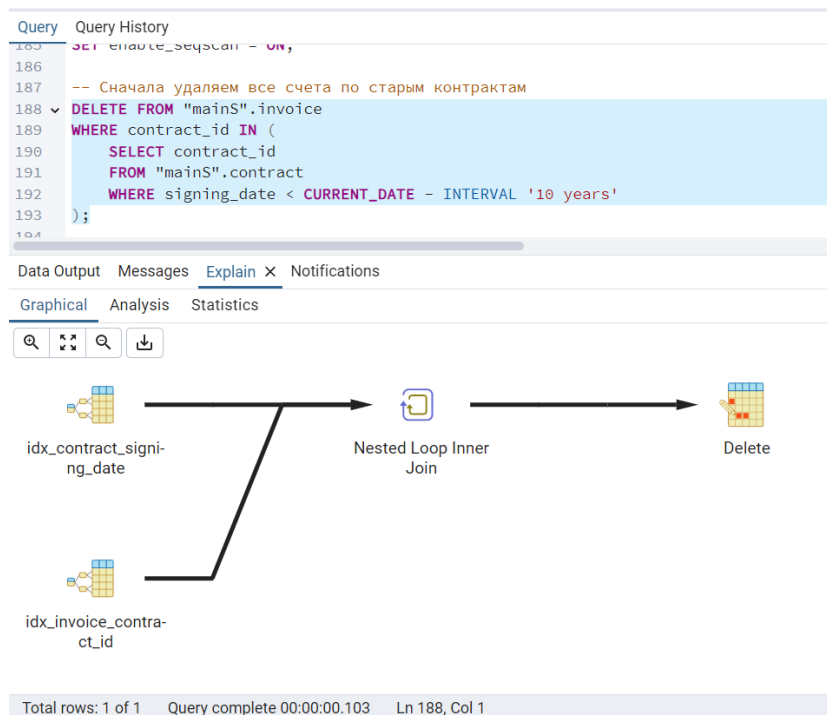
```

187 -- Сначала удаляем все счета по старым контрактам
188 ✓ DELETE FROM "mainS".invoice
189 WHERE contract_id IN (
190     SELECT contract_id
191     FROM "mainS".contract
192     WHERE signing_date < CURRENT_DATE - INTERVAL '10 years'
193 );
194
195 -- Затем удаляем сами контракты

```

Data Output		Messages	Explain X	Notifications
Graphical		Analysis	Statistics	
#	Node			
1.	→ Delete on invoice as invoice			
2.	→ Nested Loop Inner Join			
3.	→ Index Scan using idx_contract_signing_date on contract as contract Index Cond: (signing_date < (CURRENT_DATE - '10 years'::interval))			
4.	→ Index Scan using idx_invoice_contract_id on invoice as invoice Index Cond: (contract_id = contract.contract_id)			

В графическом представлении также видим использование индекса. Время увеличилось до 103 секунд, так как таблицы маленькие.



Удаляем индексы:

```

DROP INDEX "mainS".idx_contract_signing_date;
DROP INDEX "mainS".idx_invoice_contract_id;
DROP INDEX "mainS".idx_book_id_title_isbn;

```

Выводы:

Я овладела практическими навыками создания представлений и запросов на выборку данных к базе данных PostgreSQL. Научилась использовать подзапросы для модификации данных и создавать простые и составные индексы.