**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное автономное образовательное**

**учреждение высшего образования**

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Факультет инфокоммуникационных технологий**

**ОТЧЕТ**

О ЛАБОРАТОРНОЙ РАБОТЕ **№ 2**

Проектирование и реализация баз данных

Запросы на выборку данных к БД

**Специальность:**

09.03.03 Мобильные и сетевые технологии

**Проверил:**

Говорова М. М.

**Выполнил:**

студент группы K3240 Мамедов Тогрул

**Дата:** «03» 06 2022г.

**Оценка** \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург 2022

**ЦЕЛЬ РАБОТЫ:**

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

**ПРАКТИЧЕСКОЕ ЗАДАНИЕ**

1. Создать запросы и представления на выборку данных к базе данных PostgreSQL.

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

3. Изучить графическое представление запросов и посмотреть 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.

**Запросы**

1. **Вывести список читателей, имеющих на руках книги, переведенные с английского языка, изданные позднее 2000 года.**

SELECT reader.first\_name, reader.middle\_name, reader.last\_name, reader.reader\_ticket, book.original\_language, book.publish\_date, extradition.copy\_id From reader, book, extradition, copy\_book

Where extradition.reader\_ticket= reader.reader\_ticket And extradition.copy\_id = copy\_book.copy\_id And book.book\_id=

copy\_book.book\_id And book.publish\_date >='2001-01-01' AND book.original\_language='English'; Изображение выглядит как текст

Автоматически созданное описание

1. **Вывести список читателей, не вернувших в срок книги и имеющих на руках более десяти книг.**

SELECT t1.reader\_ticket, COUNT(\*)

From extradition AS t1

LEFT JOIN "reader" AS t2 ON t1.reader\_ticket = t2.reader\_ticket

Where ("return\_date" < "actual\_date" OR (current\_date > "return\_date" And "actual\_date" is null ))

Group BY t1.reader\_ticket

Having COUNT(\*) >10 ;

Изображение выглядит как текст

Автоматически созданное описание

1. **Найти количество читателей, не вернувших в срок книги и имеющих на руках более десяти книг.**

SELECT COUNT(reader\_ticket) FROM (SELECT t1.reader\_ticket, COUNT(t1.return\_date) FROM "extradition" AS t1 LEFT JOIN "reader" AS t2 ON t1.reader\_ticket = t2.reader\_ticket WHERE "return\_date"<"actual\_date" GROUP BY t1.reader\_ticket) AS t;

Изображение выглядит как текст

Автоматически созданное описание

1. **Вывести список книг, которые находятся в библиотеке в единственном экземпляре.**

SELECT \* FROM (SELECT book\_id, COUNT(book\_id) FROM "copy\_book" GROUP BY book\_id) AS

t1 WHERE count =1;

Изображение выглядит как текст

Автоматически созданное описание

1. **Подсчитать количество читателей, которые не обращались в библиотеку в течение года.**

SELECT COUNT(\*) FROM (

SELECT DISTINCT reader\_ticket FROM "extradition"

WHERE extradiction\_date < (NOW()-INTERVAL '1 year')

GROUP BY reader\_Ticket) AS T1;

Изображение выглядит как текст

Автоматически созданное описание

1. **Подсчитать количество читателей библиотеки по уровню образования.**

SELECT COUNT(level\_edu)

FROM reader

WHERE reader.level\_edu='Secondry'

UNION SELECT COUNT(level\_edu)

FROM reader

WHERE reader.level\_edu='h'

Изображение выглядит как текст

Автоматически созданное описание

1. **Вывести список книг по программированию на С#, экземпляры которых отсутствуют в библиотеке, и которые должны быть возвращены не позднее, чем через 3 дня.**

SELECT book.book\_id, book.book\_name, book.original\_language,

book.publish\_date,extradition.extradiction\_date, extradition.return\_date

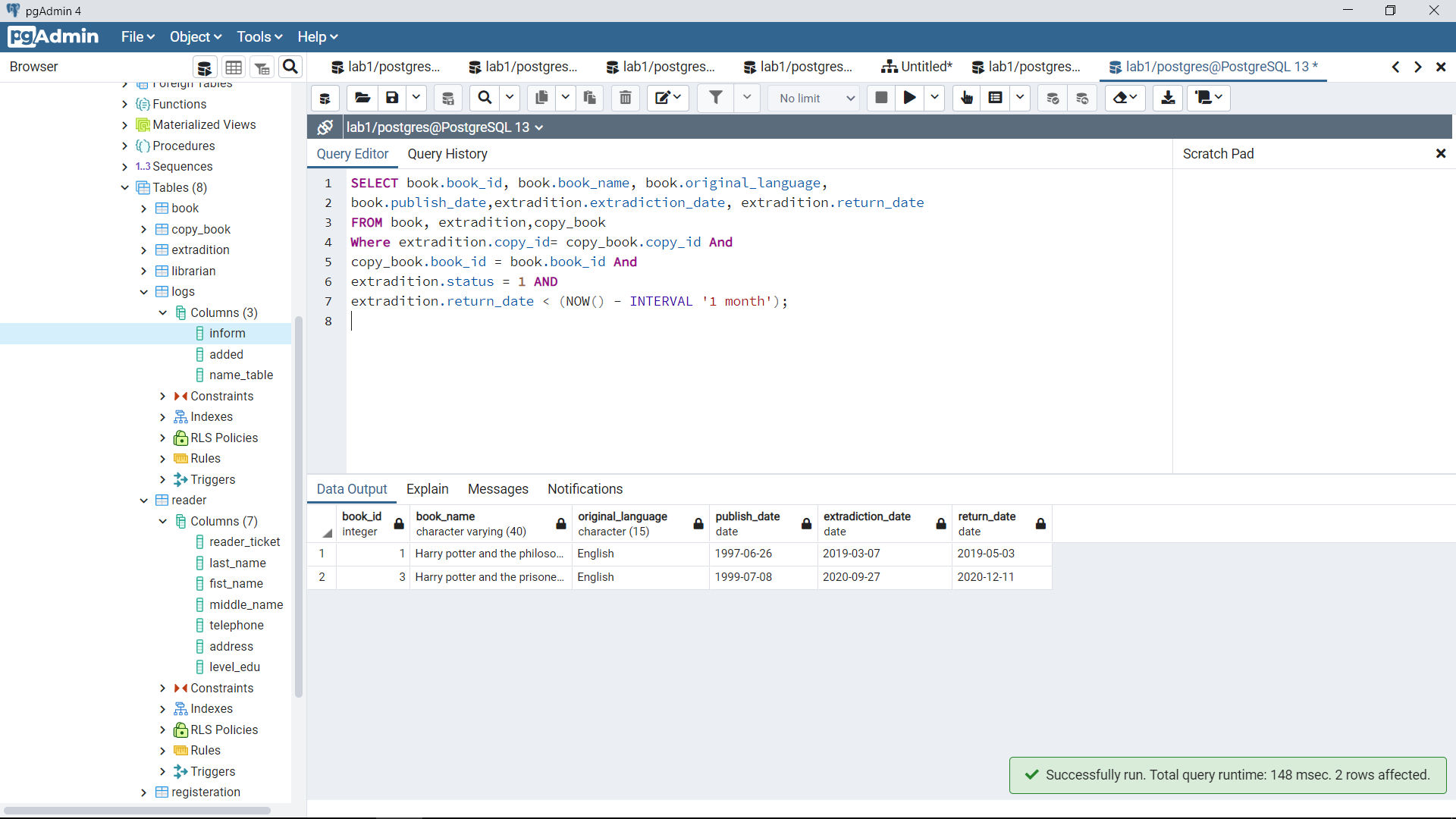
FROM book, extradition,copy\_book

Where extradition.copy\_id= copy\_book.copy\_id And

copy\_book.book\_id = book.book\_id And

extradition.status = 1 AND

extradition.return\_date < (NOW() - INTERVAL '1 month');



**Представления**

CREATE VIEW readers\_view1 AS

SELECT Distinct reader.reader\_ticket , reader.fist\_name, reader.last\_name

FROM reader,extradition

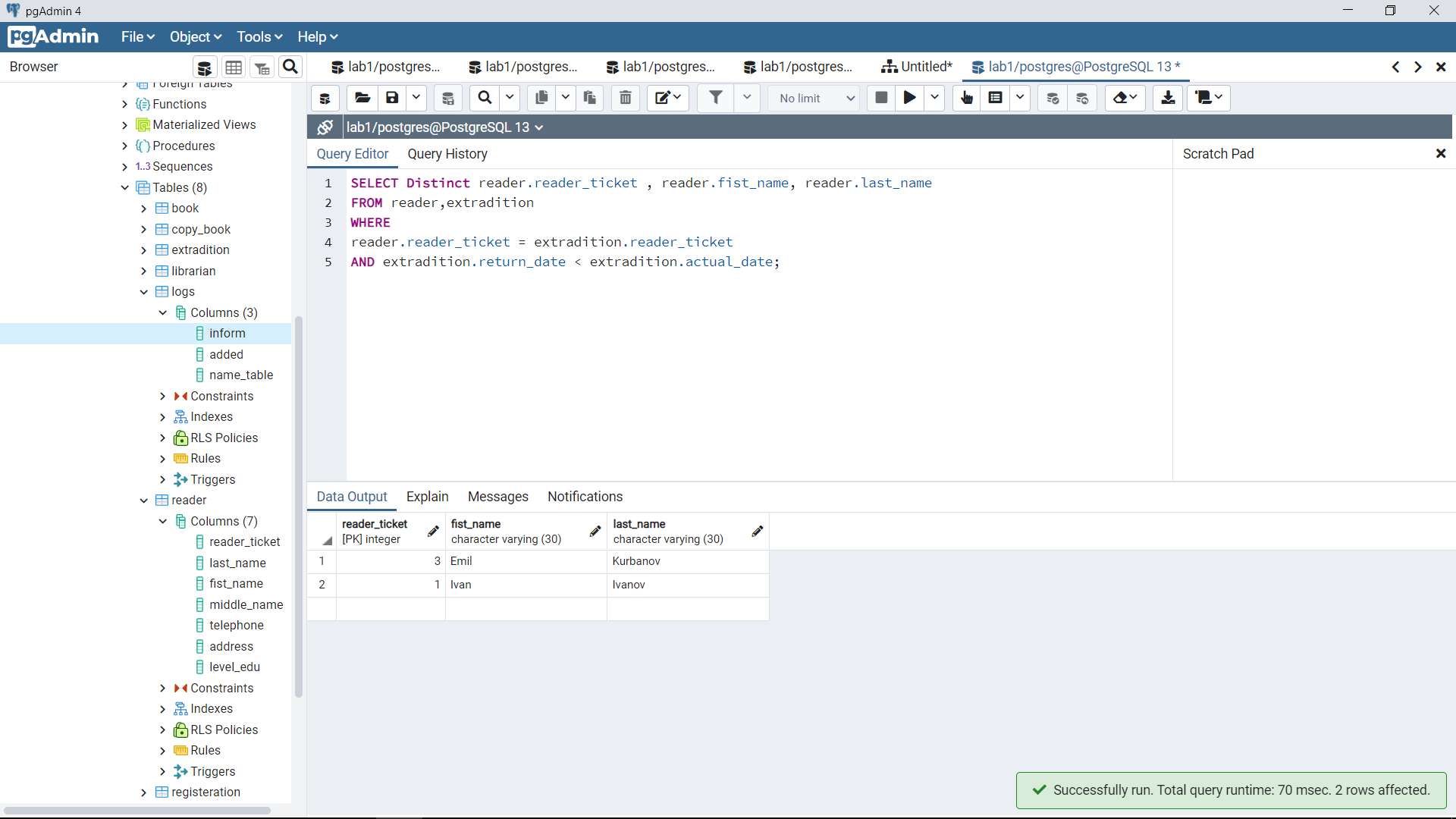
WHERE

reader.reader\_ticket = extradition.reader\_ticket

AND extradition.return\_date < extradition.actual\_date;

Изображение выглядит как текст

Автоматически созданное описание



**Cведения о наиболее популярных книгах**

CREATE VIEW popular AS

SELECT t4.book\_name, t4.writer, t3.book\_id, COUNT(t3.book\_id) FROM "book" AS t4 RIGHT

JOIN (SELECT \* FROM "copy\_book" AS t2 RIGHT JOIN (

SELECT \* FROM "extradition" WHERE return\_date IS NULL AND extradition.status=1) AS t1

ON t1.copy\_id=t2.copy\_id) AS t3

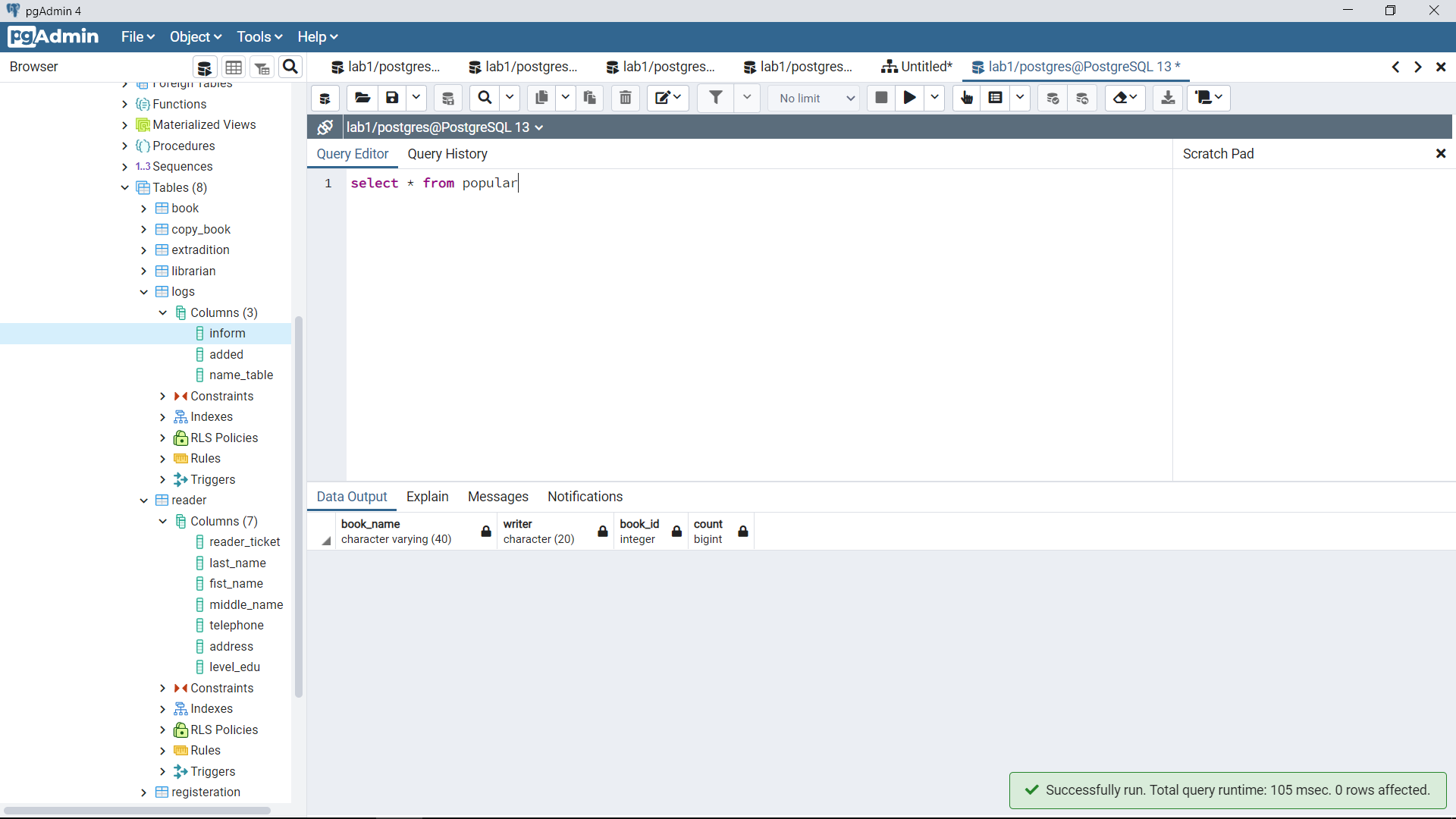
ON t3.book\_id=t4.book\_id GROUP BY (t3.book\_id, t4.writer, t4.book\_name) ORDER BY

count DESC;

ПЕРЕДЕЛАТЬ!!!!!!!!!!!!!!11Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание 

**Модификация данных**

1. **Запрос на модификацию данных с INSERT**

INSERT INTO public.book(

book\_id, translator, publishment\_place, original\_language, knowledge\_area, volume, book\_name,

writer, publish\_date)

VALUES (515, 'translator\_515', 'address\_515', 'English', 'Unknown',

(SELECT volume FROM book WHERE book.book\_name ='Harry potter and the prisoner of azkaban '), 'Moonbook', 'writer322', '1973-08-12');

Изображение выглядит как текст

Автоматически созданное описание

1. **Запрос на модификацию данных с UPDATE**

UPDATE "extradition" SET return\_date = return\_date + INTERVAL '7 DAYS' WHERE actual\_date IN ( SELECT actual\_date FROM extradition WHERE actual\_date IS NULL)

Изображение выглядит как текст

Автоматически созданное описание

1. **Запрос на модификацию данных с DELETE**

**ПЕРЕДЕЛАТЬ!**

DELETE FROM "storage\_address" where address\_id IN (SELECT storage\_address.address\_id FROM "storage\_address"

AS storage RIGHT JOIN "storage\_address" AS store ON

storage.address\_id = storage\_address.address\_id

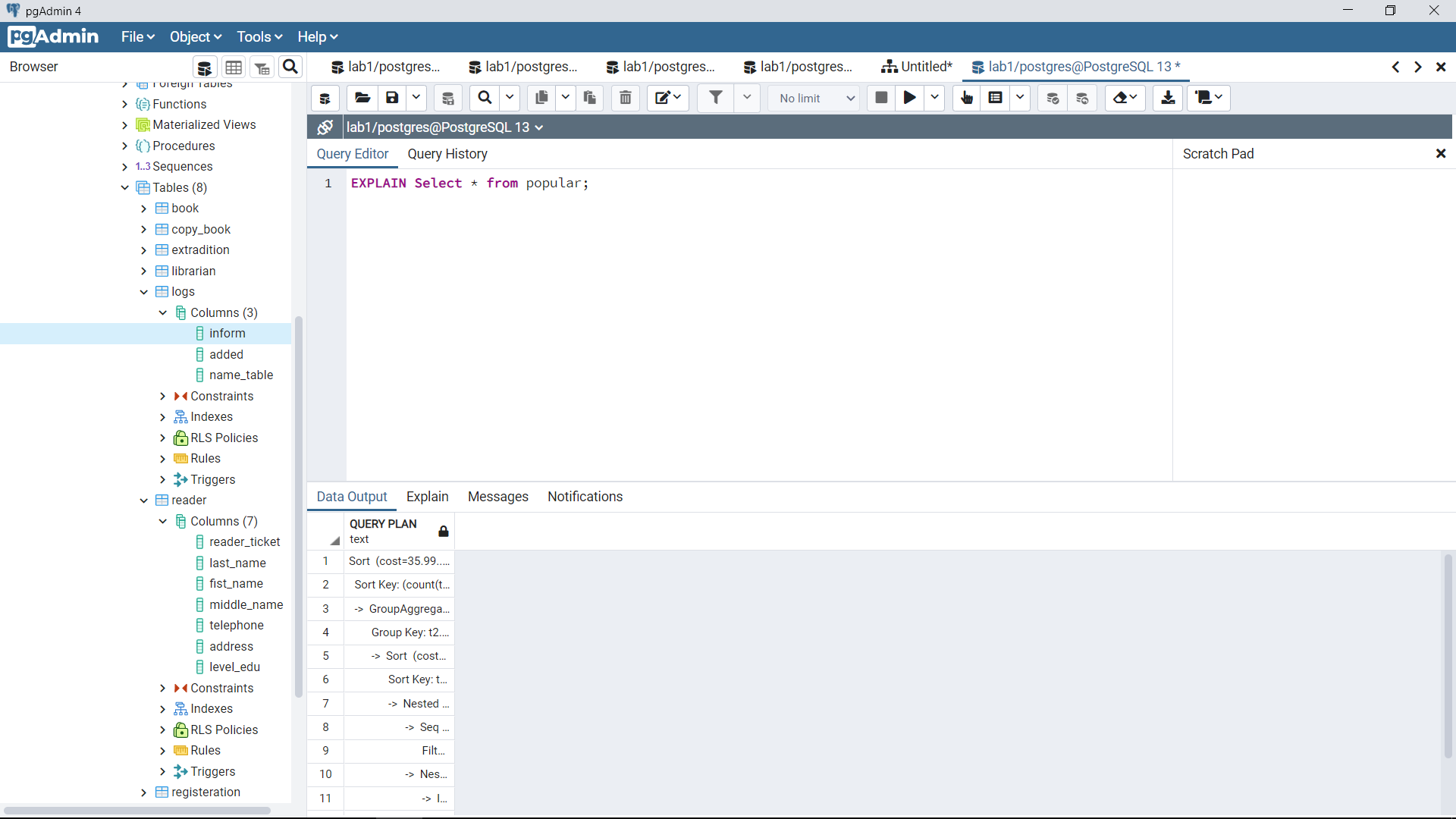
WHERE storage\_address.address\_id IS NULL);

Изображение выглядит как текст

Автоматически созданное описание

**Графическое представление запроса SELECT \* FROM popular:**

EXPLAIN Select \* from popular;



**Индексы**

Простой индекс

До создания:

Изображение выглядит как текст

Автоматически созданное описание

После:

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Составной индекс:

До создания:

Изображение выглядит как текст

Автоматически созданное описание

После:

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

**Удвление:**

Изображение выглядит как текст

Автоматически созданное описание

**ВЫВОДЫ**

SQL запросы позволяют изменять, добавлять или удалять данные, а также составлять различные выборки, подсчитывать числовые характеристики. Сравнив время выполнения запросов с индексами и без, можно сделать вывод, что с индексами запросы выполнялись немного медленнее. Это связано с небольшим количеством данных в таблице. Однако благодаря им, можно проще структурировать данные.