

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»**
Вариант 3

Автор: Мосин З. И.
Факультет: Инфокоммуникационных технологий (ИКТ)
Группа: K32402
Преподаватель: Говорова М. М.
Дата: 10.08.2023

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

Программное обеспечение: СУБД PostgreSQL 1X, pgAdmin 4

Практическое задание:

Вариант 3. БД «Библиотека»

1. Вывести список читателей, имеющих на руках книги, переведенные с английского языка, изданные позднее 2000 года.
2. Вывести список читателей, не вернувших в срок книги и имеющих на руках более десяти книг.
3. Найти количество читателей, не вернувших в срок книги и имеющих на руках более десяти книг.
4. Вывести список книг, которые находятся в библиотеке в единственном экземпляре.
5. Подсчитать количество читателей, которые не обращались в библиотеку в течение года.
6. Подсчитать количество читателей библиотеки по уровню образования.
7. Вывести список книг по программированию на C#, экземпляры которых отсутствуют в библиотеке, и которые должны быть возвращены не позднее, чем через 3 дня.

Выполнение:

Задание 1. Создайте запросы.

Запрос №1 - Вывести список читателей, имеющих на руках книги, переведенные с английского языка, изданные позднее 2000 года.

```
SELECT full_name
FROM "library"."reader"
LEFT JOIN "library"."reader_card" t
ON ("reader".reader_id = t.reader_id)
LEFT JOIN "library"."subscription" a
ON (t.card_id = a.card_id)
LEFT JOIN "library"."book_copy"
ON ("book_copy".copy_id = a.copy_id)
LEFT JOIN "library"."edition"
ON ("edition".edition_id = "book_copy".edition_id)
LEFT JOIN "library"."book_info"
ON ("book_info".book_id = "edition".book_id)
WHERE "edition".edition_date > '2000-12-31'::date
AND "book_info".original_language = 'английский'
AND "edition".edition_language != "book_info".original_language
```

```
1  SELECT full_name
2  FROM "library"."reader"
3  LEFT JOIN "library"."reader_card" t
4  ON ("reader".reader_id = t.reader_id)
5  LEFT JOIN "library"."subscription" a
6  ON (t.card_id = a.card_id)
7  LEFT JOIN "library"."book_copy"
8  ON ("book_copy".copy_id = a.copy_id)
9  LEFT JOIN "library"."edition"
10 ON ("edition".edition_id = "book_copy".edition_id)
11 LEFT JOIN "library"."book_info"
12 ON ("book_info".book_id = "edition".book_id)
13 WHERE "edition".edition_date > '2000-12-31'::date
14 AND "book_info".original_language = 'английский'
15 AND "edition".edition_language != "book_info".original_language
```

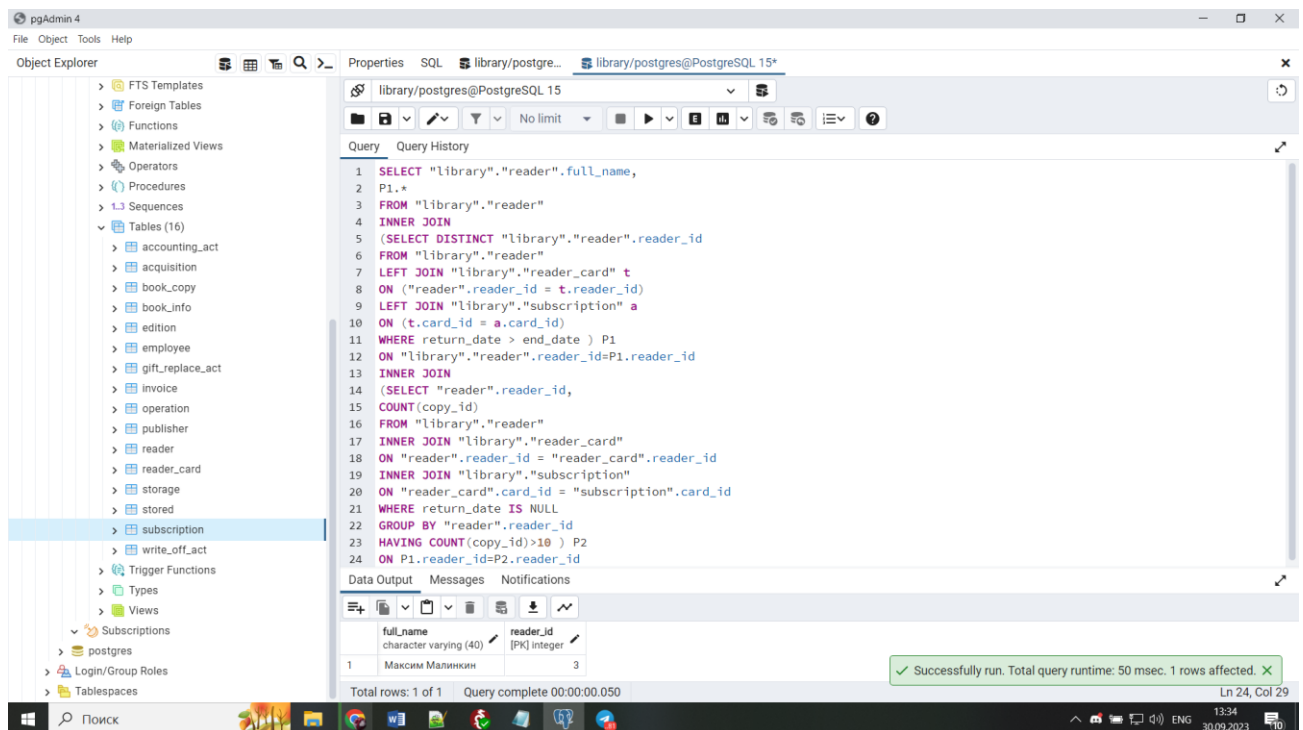
Data Output Messages Notifications



	full_name character varying (40) 🔒
1	Максим Малинкин

Запрос №2 - Вывести список читателей, не вернувших в срок книги и имеющих на руках более десяти книг

```
SELECT "library"."reader".full_name,  
P1.*  
FROM "library"."reader"  
INNER JOIN  
(SELECT DISTINCT "library"."reader".reader_id  
FROM "library"."reader"  
LEFT JOIN "library"."reader_card" t  
ON ("reader".reader_id = t.reader_id)  
LEFT JOIN "library"."subscription" a  
ON (t.card_id = a.card_id)  
WHERE return_date > end_date ) P1  
ON "library"."reader".reader_id=P1.reader_id  
INNER JOIN  
(SELECT "reader".reader_id,  
COUNT(book_copy_id)  
FROM "library"."reader"  
INNER JOIN "library"."reader_card"  
ON "reader".reader_id = "reader_card".reader_id  
INNER JOIN "library"."subscription"  
ON "reader_card".card_id = "subscription".card_id  
WHERE return_date IS NULL  
GROUP BY "reader".reader_id  
HAVING COUNT(book_copy_id)>10 ) P2  
ON P1.reader_id=P2.reader_id
```



Запрос №3 - Найти количество читателей, не вернувших в срок книги и имеющих на руках более десяти книг.

```

SELECT COUNT(DISTINCT "library"."reader".reader_id) AS count_debtors
FROM "library"."reader"
INNER JOIN(
    SELECT DISTINCT "library"."reader".reader_id
    FROM "library"."reader"
    LEFT JOIN "library"."reader_card" t
    ON ("library"."reader".reader_id = t.reader_id)
    LEFT JOIN "library"."subscription" a
    ON (t.card_id = a.card_id)
    WHERE return_date > end_date
) P1
ON "library"."reader".reader_id = P1.reader_id
INNER JOIN
(
    SELECT "reader".reader_id
    FROM "library"."reader"
    INNER JOIN "library"."reader_card"

```

```

ON "reader".reader_id = "reader_card".reader_id
INNER JOIN "library"."subscription"
ON "reader_card".card_id = "subscription".card_id
WHERE return_date IS NULL
GROUP BY "reader".reader_id
HAVING COUNT(copy_id) > 10) P2
ON P1.reader_id = P2.reader_id;

```

```

1  SELECT COUNT(DISTINCT "library"."reader".reader_id) AS count_debtors
2  FROM "library"."reader"
3  INNER JOIN
4  (
5      SELECT DISTINCT "library"."reader".reader_id
6      FROM "library"."reader"
7      LEFT JOIN "library"."reader_card" t
8      ON ("library"."reader".reader_id = t.reader_id)
9      LEFT JOIN "library"."subscription" a
10     ON (t.card_id = a.card_id)
11     WHERE return_date > end_date
12 ) P1
13 ON "library"."reader".reader_id = P1.reader_id
14 INNER JOIN
15 (
16     SELECT "reader".reader_id
17     FROM "library"."reader"
18     INNER JOIN "library"."reader_card"
19     ON "reader".reader_id = "reader_card".reader_id
20     INNER JOIN "library"."subscription"

```

Data Output Messages Notifications



	count_debtors bigint	
1		1

Запрос №4 - Вывести список книг, которые находятся в библиотеке в единственном экземпляре.

```
SELECT book_name, book_author
FROM "library"."book_book_copy"
LEFT JOIN "library"."edition"
ON ("edition".edition_id = "book_book_copy".edition_id)
LEFT JOIN "library"."book_info"
ON ("book_info".book_id = "edition".book_id)
GROUP BY book_name, book_author
HAVING COUNT("book_book_copy".book_copy_id) = 1
```

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer displays the database structure, including tables like 'book_book_copy', 'edition', and 'book_info'. The 'book_book_copy' table is selected, showing its columns: book_id, book_name, field_of_knowledge, original_language, book_author, and type. The main pane displays the SQL query:

```
1 SELECT book_name, book_author
2 FROM "library"."book_book_copy"
3 LEFT JOIN "library"."edition"
4 ON ("edition".edition_id = "book_book_copy".edition_id)
5 LEFT JOIN "library"."book_info"
6 ON ("book_info".book_id = "edition".book_id)
7 GROUP BY book_name, book_author
8 HAVING COUNT("book_book_copy".copy_id) = 1
```

Below the query, the Data Output pane shows the results of the query:

	book_name	book_author
1	Три товарища	Эрих Мария Ремарк
2	Преступление и наказание	Ф.М. Достоевский

At the bottom, a status bar indicates: 'Total rows: 2 of 2', 'Query complete 00:00:00.132', and a green message: 'Successfully run. Total query runtime: 132 msec. 2 rows affected.'

Запрос №5 - Подсчитать количество читателей, которые не обращались в библиотеку в течение года.

```
SELECT COUNT(DISTINCT full_name) AS nonactive_readers
FROM (
    SELECT full_name
    FROM "library"."reader"
    LEFT JOIN "library"."reader_card" t
    ON ("reader".reader_id = t.reader_id)
    LEFT JOIN "library"."subscription" a
    ON (t.card_id = a.card_id)
    WHERE start_date IS NULL
    OR start_date < DATE_TRUNC('YEAR', CURRENT_DATE)
) AS subquery;
```

```
1  SELECT COUNT(DISTINCT full_name) AS nonactive_readers
2  FROM (
3      SELECT full_name
4      FROM "library"."reader"
5      LEFT JOIN "library"."reader_card" t
6      ON ("reader".reader_id = t.reader_id)
7      LEFT JOIN "library"."subscription" a
8      ON (t.card_id = a.card_id)
9      WHERE start_date IS NULL
10     OR start_date < DATE_TRUNC('YEAR', CURRENT_DATE)
11 ) AS subquery;
```

Data Output Messages Notifications

	nonactive_readers bigint	
1		1

Запрос №6 - Подсчитать количество читателей библиотеки по уровню образования.

SELECT education_level,

COUNT(reader_id)

FROM "library"."reader"

GROUP BY education_level

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer shows the database structure, with the 'reader' table selected. The main pane displays the SQL query:

```
1 SELECT education_level,  
2 COUNT(reader_id)  
3 FROM "library"."reader"  
4 GROUP BY education_level
```

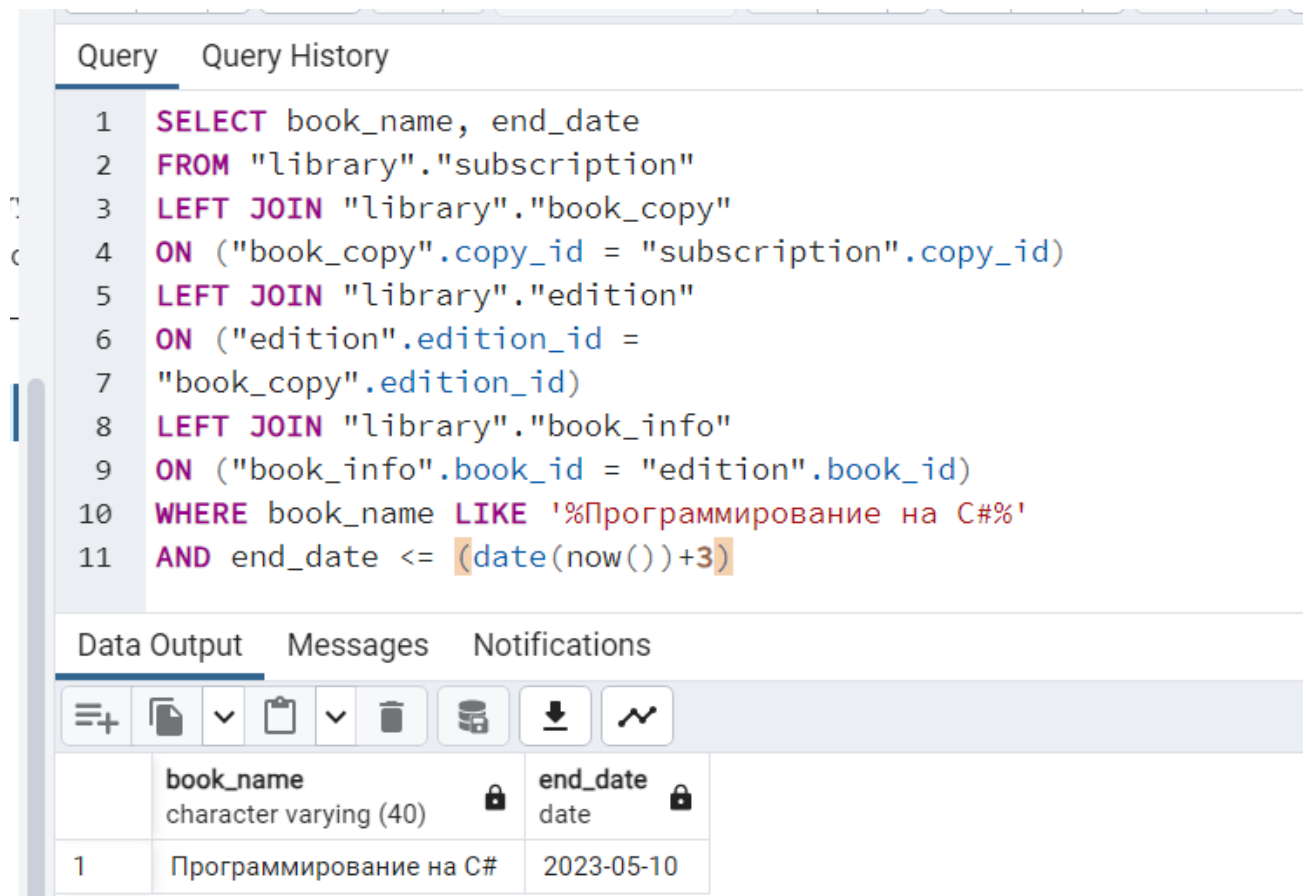
Below the query, the Data Output tab shows the results of the query:

education_level	count
1	3
2	5
3	7

At the bottom, a status bar indicates: "Total rows: 3 of 3 Query complete 00:00:00.048". A green message box at the bottom right states: "Successfully run. Total query runtime: 48 msec. 3 rows affected."

Запрос №7 - Вывести список книг по программированию на C#, экземпляры которых отсутствуют в библиотеке, и которые должны быть возвращены не позднее, чем через 3 дня.

```
SELECT book_name, end_date
FROM "library"."subscription"
LEFT JOIN "library"."book_copy"
ON ("book_copy".copy_id = "subscription".copy_id)
LEFT JOIN "library"."edition"
ON ("edition".edition_id =
"book_copy".edition_id)
LEFT JOIN "library"."book_info"
ON ("book_info".book_id = "edition".book_id)
WHERE book_name LIKE '%Программирование на C#%'
AND end_date <= (date(now())+3)
```



The screenshot shows a database query editor interface. The top section is titled "Query" and "Query History". The query text is displayed in a monospaced font with syntax highlighting. Below the query, there are tabs for "Data Output", "Messages", and "Notifications". The "Data Output" tab is active, showing a table with two columns: "book_name" and "end_date". The table has one row of data.

	book_name character varying (40)	end_date date
1	Программирование на C#	2023-05-10

Задание 2. Создать представления для администрации библиотеки.

Представление №1 - сведения о должниках.

```
CREATE VIEW debtors AS
(SELECT "library"."reader".full_name,
P2.*
FROM "library"."reader"
INNER JOIN
(SELECT DISTINCT "library"."reader".reader_id
FROM "library"."reader"
LEFT JOIN "library"."reader_card" t
ON ("reader".reader_id = t.reader_id)
LEFT JOIN "library"."subscription" a
ON (t.card_id = a.card_id)
WHERE return_date > end_date ) P1
ON "library"."reader".reader_id=P1.reader_id
INNER JOIN
(SELECT "reader".reader_id,
COUNT(copy_id)
FROM "library"."reader"
INNER JOIN "library"."reader_card"
ON "reader".reader_id = "reader_card".reader_id
INNER JOIN "library"."subscription"
ON "reader_card".card_id = "subscription".card_id
GROUP BY "reader".reader_id
HAVING COUNT(copy_id)>=1 ) P2
ON P1.reader_id=P2.reader_id);SELECT *
FROM debtors
```

pgAdmin 4

File Object Tools Help

Object Explorer

- Servers
 - PostgreSQL 15
 - Databases (2)
 - library
 - Aggregates
 - Collations
 - Domains
 - FTS Configurations
 - FTS Dictionaries
 - FTS Parsers
 - FTS Templates
 - Foreign Tables
 - Functions
 - Materialized Views
 - Operators
 - Procedures
 - Sequences
 - Tables (16)
 - accounting_act
 - acquisition
 - book_copy
 - book_info

Properties SQL library/postgres@PostgreSQL 15*

library/postgres@PostgreSQL 15

No limit

Query Query History

```
1 CREATE VIEW debtors AS
2 (SELECT "library"."reader".full_name,
3 P2.*
4 FROM "library"."reader"
5 INNER JOIN
6 (SELECT DISTINCT "library"."reader".reader_id
7 FROM "library"."reader"
8 LEFT JOIN "library"."reader_card" t
9 ON ("reader".reader_id = t.reader_id)
10 LEFT JOIN "library"."subscription" a
11 ON (t.card_id = a.card_id)
12 WHERE return_date > end_date ) P1
13 ON "library"."reader".reader_id=P1.reader_id
14 INNER JOIN
15 (SELECT "reader".reader_id,
16 COUNT(copy_id)
17 FROM "library"."reader"
18 INNER JOIN "library"."reader_card"
19 ON "reader".reader_id = "reader_card".reader_id
20 INNER JOIN "library"."subscription"
21 ON "reader_card".card_id = "subscription".card_id
22 GROUP BY "reader".reader_id
23 HAVING COUNT(copy_id)>=1 ) P2
24 ON P1.reader_id=P2.reader_id);SELECT *
```

Data Output Messages Notifications

	full_name character varying (40)	reader_id integer	count bigint
1	Максим Малинкин	3	12

Total rows: 1 of 1 Query complete 00:00:00.184 Ln 25, Col 13

Поиск

1:16 03.10.2023

Представление №2 - сведения о наиболее популярных книгах (все экземпляры находятся на руках у читателей).

```
CREATE VIEW popular_books AS
SELECT
    bi.book_id,
    bi.book_name
FROM
    "library"."book_info" bi
WHERE
    bi.book_id NOT IN (
        SELECT e.book_id FROM "library"."edition" e
        LEFT JOIN "library"."book_copy" c
        ON e.edition_id = c.edition_id
        LEFT JOIN "library"."subscription" s
        ON c.copy_id = s.copy_id
        WHERE s.return_date IS NULL
    );
```

```
1 CREATE VIEW popular_books AS
2 SELECT
3     bi.book_id,
4     bi.book_name
5 FROM
6     "library"."book_info" bi
7 WHERE
8     bi.book_id NOT IN (
9         SELECT e.book_id FROM "library"."edition" e
10        LEFT JOIN "library"."book_copy" c
11        ON e.edition_id = c.edition_id
12        LEFT JOIN "library"."subscription" s
13        ON c.copy_id = s.copy_id
14        WHERE s.return_date IS NULL
15    );
```

Data Output Messages Notifications

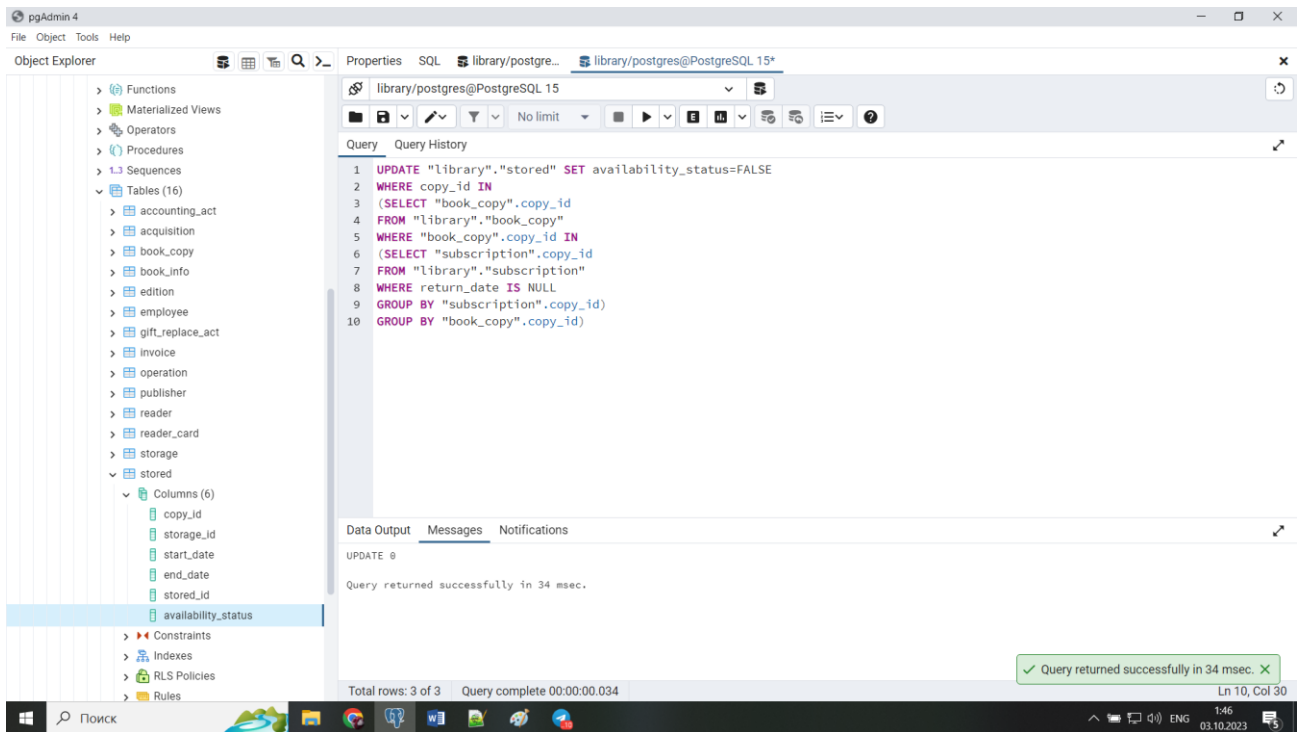
	book_id integer	book_name character varying (40)
1	1	Преступление и наказание
2	8	Над пропастью во ржи

Задание 3. Запросы на модификацию данных

Запрос 1. Update с подзапросом

Запрос: изменить статус на занято у тех книг, которые находятся на руках.

```
UPDATE "library"."stored" SET availability_status=FALSE
WHERE copy_id IN
(SELECT "book_copy".copy_id
FROM "library"."book_copy"
WHERE "book_copy".copy_id IN
(SELECT "subscription".copy_id
FROM "library"."subscription"
WHERE return_date IS NULL
GROUP BY "subscription".copy_id)
GROUP BY "book_copy".copy_id)
```



pgAdmin 4

File Object Tools Help

Object Explorer

- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (16)
 - accounting_act
 - acquisition
 - book_copy
 - book_info
 - edition
 - employee
 - gift_replace_act
 - invoice
 - operation
 - publisher
 - reader
 - reader_card
 - storage
 - stored
 - subscription
 - write_off_act
- Trigger Functions
- Types
- Views
- Subscriptions
- postgres
- Login/Group Roles
- Tablespaces

Properties SQL library/postgres@PostgreSQL 15* library.storage/... library.stored/li... library.subscrip...

library/postgres@PostgreSQL 15

Query Query History

```
1 SELECT * FROM "library"."stored"
2 ORDER BY stored_id ASC
```

Data Output Messages Notifications

copy_id	storage_id	start_date	end_date	stored_id	availability_status
integer	integer	date	date	[PK] integer	boolean
1	17	2010-01-01	2025-01-01	1	true

Total rows: 1 of 1 Query complete 00:00:00.054

Successfully run. Total query runtime: 54 msec. 1 rows affected.

Ln 2, Col 23

1:52 03.10.2023

Запрос 2. Delete с подзапросом

Запрос: удалить карту читателя, который суммарно получил штрафов больше 1000р.

DELETE

FROM "library"."reader_card"

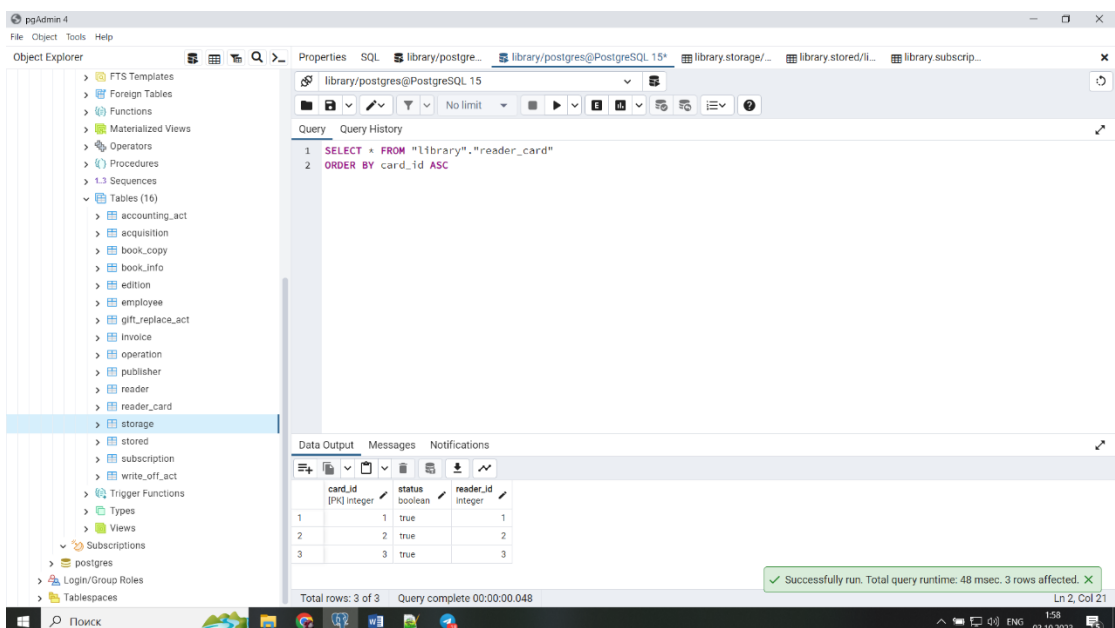
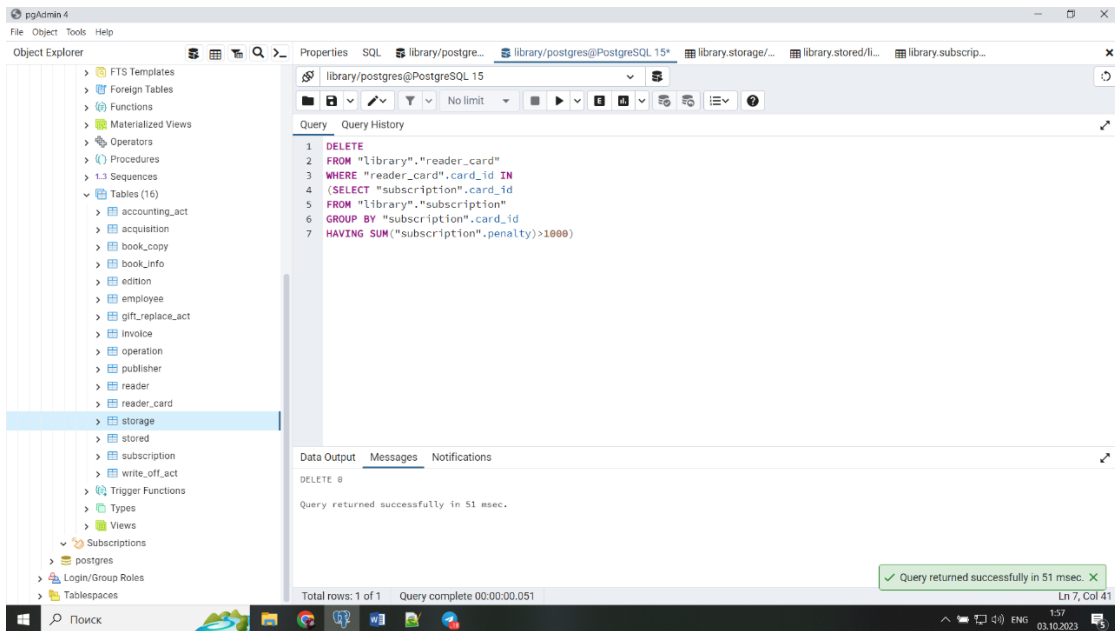
WHERE "reader_card".card_id IN

(SELECT "subscription".card_id

FROM "library"."subscription"

GROUP BY "subscription".card_id

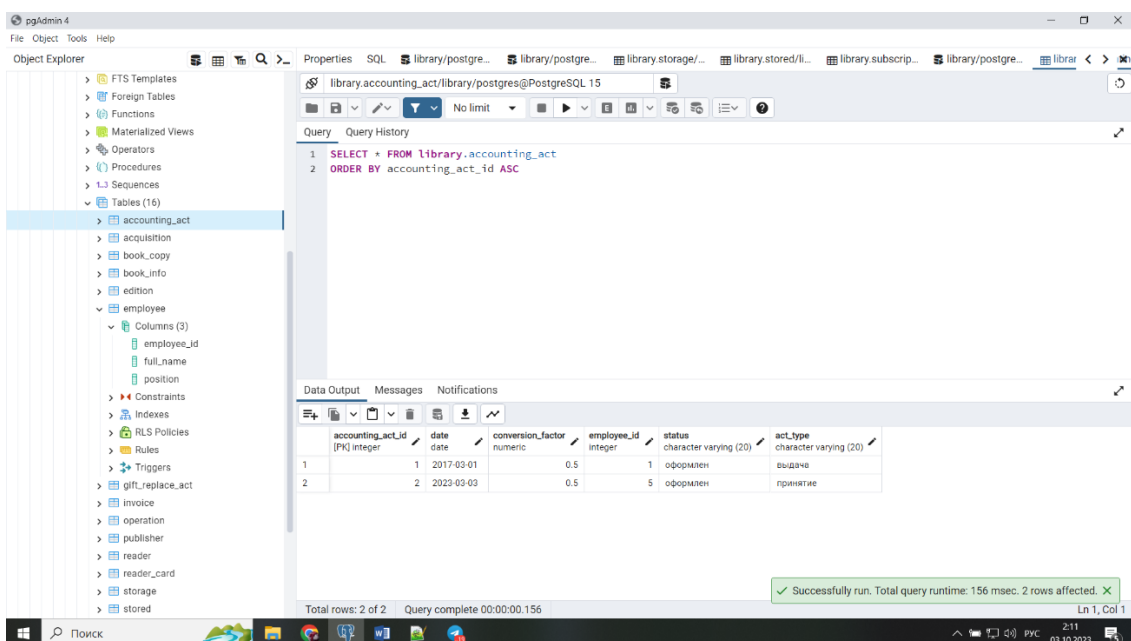
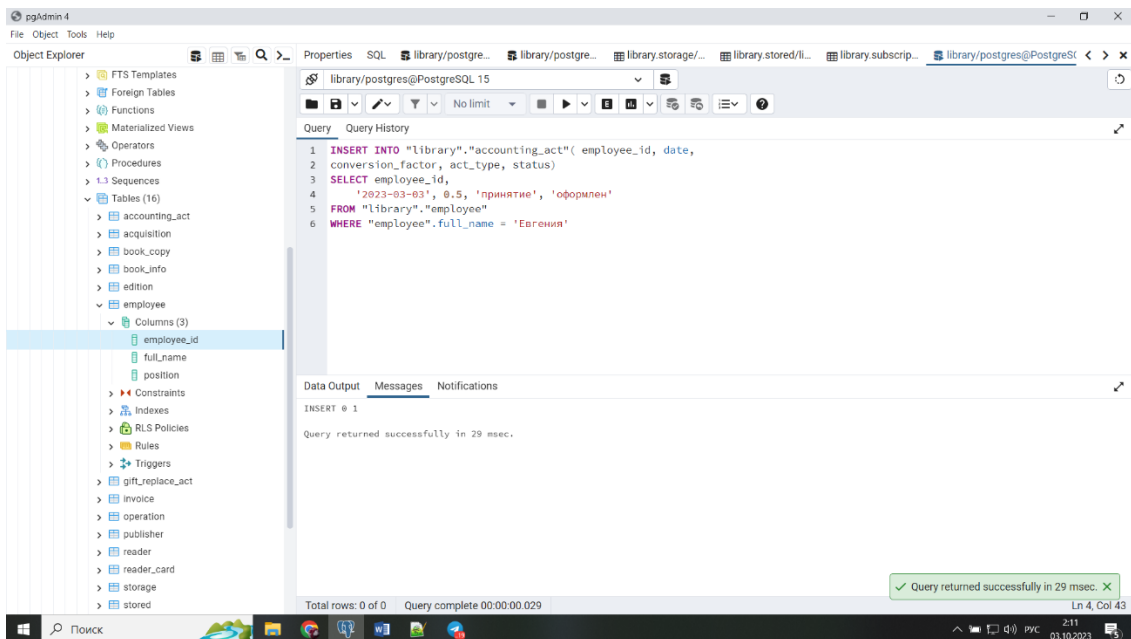
HAVING SUM("subscription".penalty)>1000)



Запрос 3. Insert с подзапросом

Запрос: добавить новый акт принятия книги, оформленный библиотекарем Евгенией сегодня.

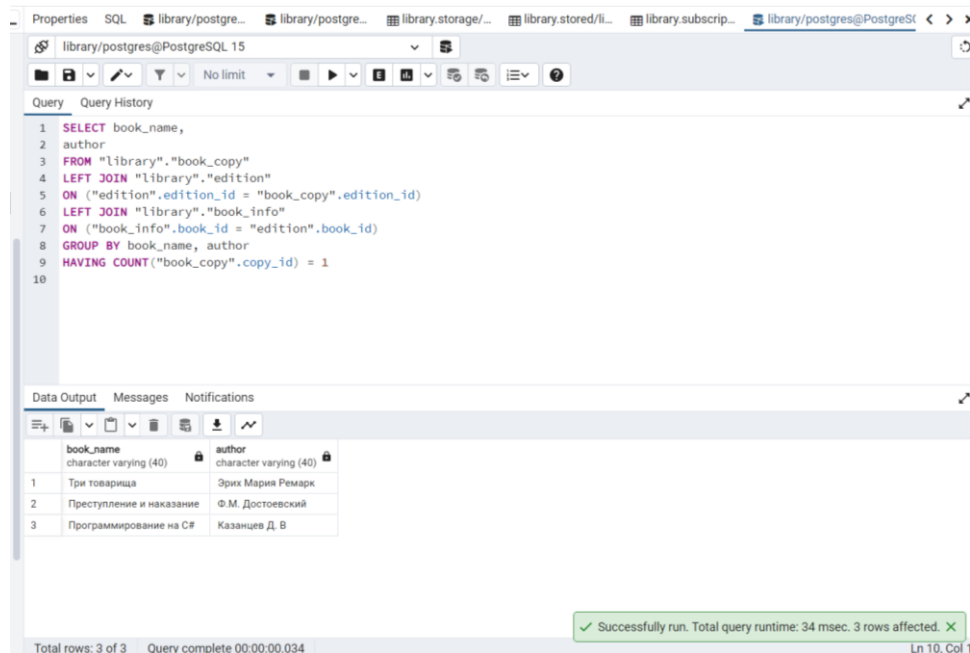
```
INSERT INTO "library"."accounting_act"( employee_id, date,  
conversion_factor, type, status)  
SELECT employee_id,  
'2023-03-10', 0.002, 'принятие', 'оформлен'  
FROM "library"."employee"  
WHERE "employee".full_name = 'Евгения'
```



Запрос 4. Индексы

1) Запрос без индекса.

```
SELECT book_name,  
author  
FROM "library"."book_copy"  
LEFT JOIN "library"."edition"  
ON ("edition".edition_id = "book_copy".edition_id)  
LEFT JOIN "library"."book_info"  
ON ("book_info".book_id = "edition".book_id)  
GROUP BY book_name, author  
HAVING COUNT("book_copy".copy_id) = 1
```



The screenshot shows a PostgreSQL query editor with the following query:

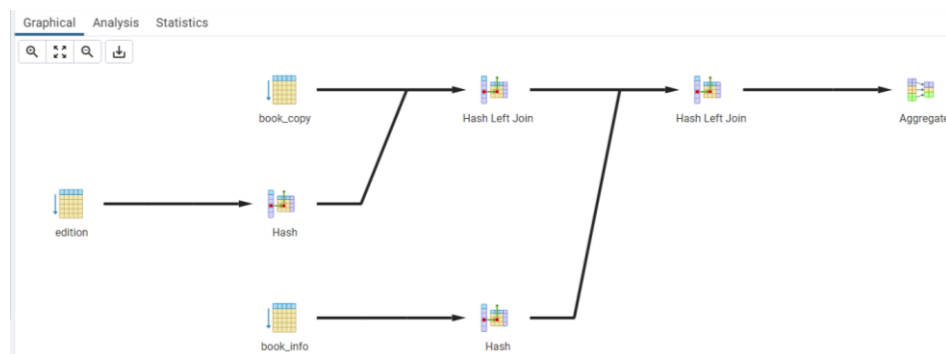
```
1 SELECT book_name,  
2 author  
3 FROM "library"."book_copy"  
4 LEFT JOIN "library"."edition"  
5 ON ("edition".edition_id = "book_copy".edition_id)  
6 LEFT JOIN "library"."book_info"  
7 ON ("book_info".book_id = "edition".book_id)  
8 GROUP BY book_name, author  
9 HAVING COUNT("book_copy".copy_id) = 1  
10
```

The 'Data Output' tab shows the results of the query:

book_name	author
Три товарища	Эрих Мария Ремарк
Преступление и наказание	Ф. М. Достоевский
Программирование на C#	Казанцев Д. В

At the bottom, a status bar indicates: 'Total rows: 3 of 3 Query complete 00:00:00.034' and a green message box says 'Successfully run. Total query runtime: 34 msec. 3 rows affected.'

План запроса в графическом виде:



План запроса в аналитическом виде:

Query		Query History	
1	SELECT book_name,		
2	author		
3	FROM "library"."book_copy"		
4	LEFT JOIN "library"."edition"		
5	ON ("edition".edition_id = "book_copy".edition_id)		
Data Output Messages Explain x Notifications			
Graphical		Analysis	
#	Node	Rows Actual	Loops
1.	→ Aggregate (rows=3 loops=1) Filter: (count(book_copy.copy_id) = 1) Rows Removed by Filter: 4 Buckets: Batches: Memory Usage: 121 kB	3	1
2.	→ Hash Left Join (rows=18 loops=1) Hash Cond: (edition.book_id = book_info.book_id)	18	1
3.	→ Hash Left Join (rows=18 loops=1) Hash Cond: (book_copy.edition_id = edition.edition_id)	18	1
4.	→ Seq Scan on book_copy as book_copy (rows=18 loops=1)	18	1
5.	→ Hash (rows=7 loops=1) Buckets: 1024 Batches: 1 Memory Usage: 9 kB	7	1
6.	→ Seq Scan on edition as edition (rows=7 loops=1)	7	1
7.	→ Hash (rows=7 loops=1) Buckets: 1024 Batches: 1 Memory Usage: 9 kB	7	1
8.	→ Seq Scan on book_info as book_info (rows=7 loops=1)	7	1
Total rows: 1 of 1 Query complete 00:00:00.042 Ln 10, Col 1			

Создание индекса:

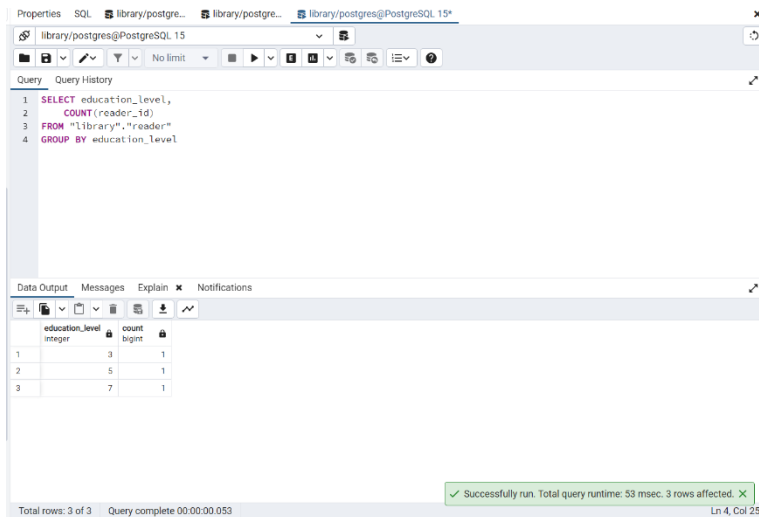
```
Query  Query History
1 CREATE INDEX "index_book_id" ON "library"."edition"(book_id)

Data Output  Messages  Explain  Notifications
CREATE INDEX

Query returned successfully in 150 msec.
```

2) Запрос с индексом 2

```
SELECT education_level,  
       COUNT(reader_id)  
  
FROM "library"."reader"  
  
GROUP BY education_level
```



Query

```
1 SELECT education_level,  
2     COUNT(reader_id)  
3 FROM "library"."reader"  
4 GROUP BY education_level
```

education_level	count
1	3
2	5
3	7

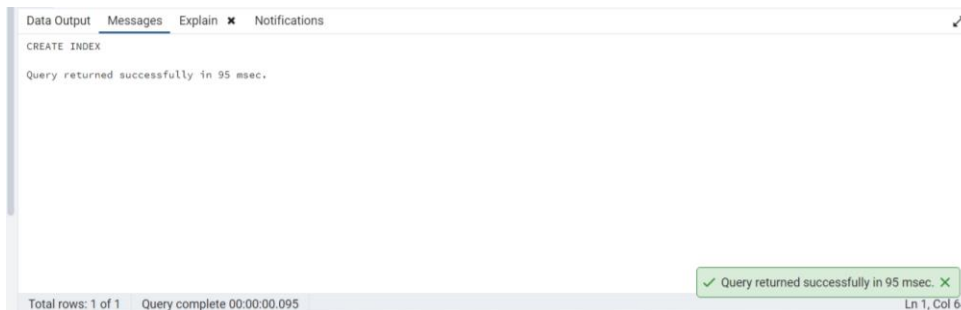
Total rows: 3 of 3 Query complete 00:00:00.053

Successfully run. Total query runtime: 53 msec. 3 rows affected.

План запроса:



Время выполнения:



CREATE INDEX

Query returned successfully in 95 msec.

Total rows: 1 of 1 Query complete 00:00:00.095

Query returned successfully in 95 msec.

Выводы:

В процессе выполнения лабораторной работы были созданы запросы для извлечения данных из базы данных PostgreSQL в соответствии с индивидуальными заданиями части 2 и 3. Также были разработаны три запроса для изменения данных (INSERT, UPDATE, DELETE), которые включали использование подзапросов. Мы также провели анализ графических представлений запросов с помощью EXPLAIN. Мы успешно выполнили все этапы, указанные в практическом задании, и приобрели опыт работы с представлениями, индексами и

выполнением запросов на изменение данных с использованием подзапросов. Особенно стоит отметить, что использование индексов в больших запросах значительно сокращает время их выполнения.

