

Министерство науки и высшего образования Российской
Федерации

федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет инфокоммуникационных технологий

Лабораторная работа №2

«ЗАПРОСЫ НА ВЫБОРКУ И МОДИФИКАЦИЮ ДАННЫХ, ПРЕДСТАВЛЕНИЯ И ИНДЕКСЫ В
POSTGRESQL»

Выполнил:

Чухонин Иван Андреевич
Группа К32421

Преподаватель:

Говорова Мария Михайловна

Санкт-Петербург
2023

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.

Индивидуальное задание - Вариант 3. БД «Библиотека»

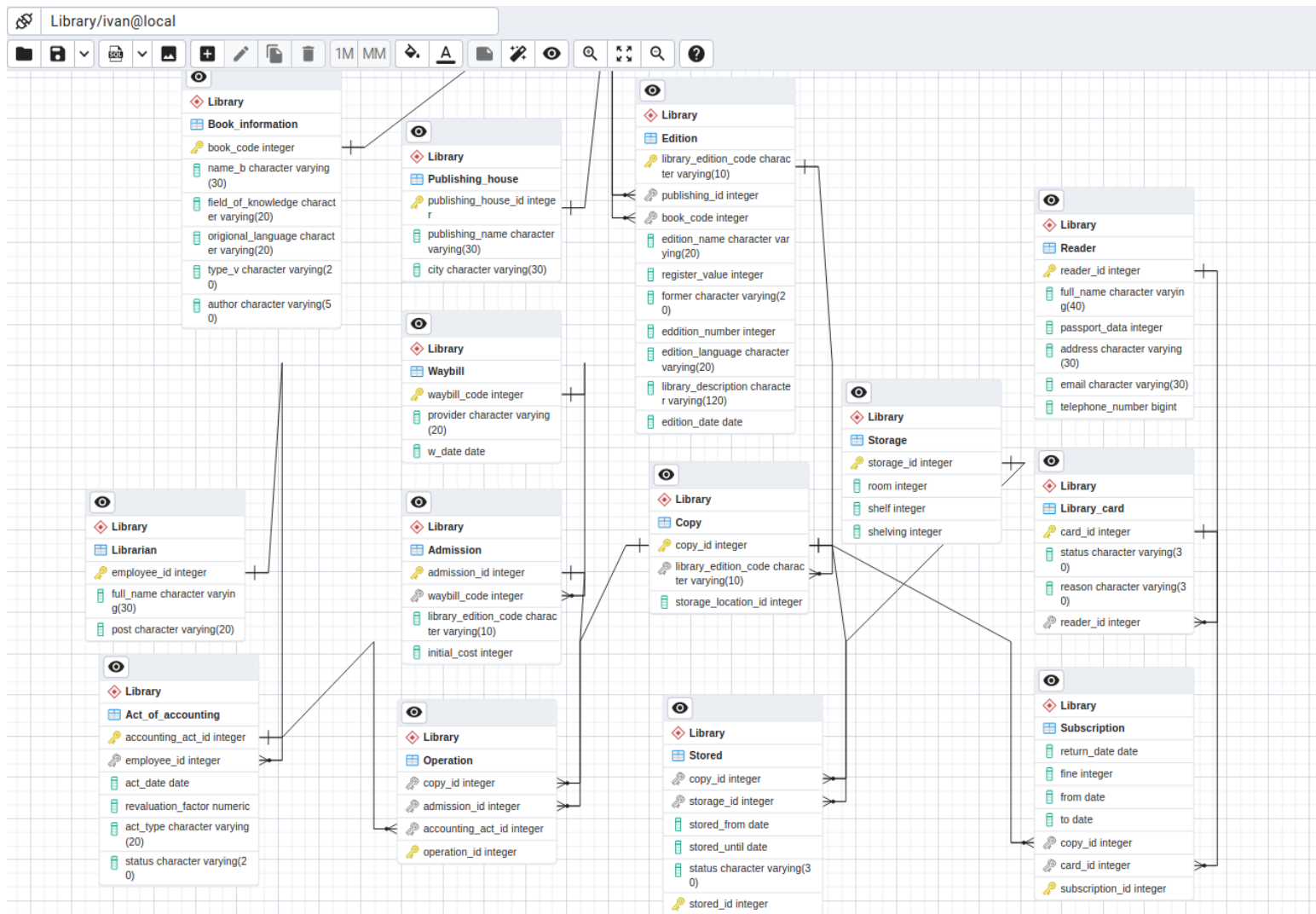


рисунок №1 - схема библиотеки

Задание 1. Создайте запросы:

Выполнение запросов:

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"."Library_card" t
      ON ("Reader".reader_id = t.reader_id)
LEFT JOIN "Library"."Subscription" a
      ON (t.card_id = a.card_id)
LEFT JOIN "Library"."Copy"
      ON ("Copy".copy_id = a.copy_id)
LEFT JOIN "Library"."Edition"
      ON ("Edition".library_edition_code = "Copy".library_edition_code)
LEFT JOIN "Library"."Book_information"
      ON ("Book_information".book_code = "Edition".book_code)
WHERE "Edition".edition_date > DATE(NOW())
      AND "Book_information".original_language = 'английский'
```

The screenshot shows a database query editor interface. The top toolbar includes icons for file operations, query execution, and help. The main area displays the SQL query from the previous block. Below the query, the 'Data Output' tab is active, showing a table with 10 columns: reader_id, full_name, passport_data, address, email, telephone_number, card_id, status, reason, and reader_id. The first row of data shows a reader named 'Женя' with passport number 20230322, address 'улица Яхтенная 15', email 'whale@gmail.com', telephone number 89112442323, card_id 103, and status 'действующий'.

	reader_id	full_name	passport_data	address	email	telephone_number	card_id	status	reason	reader_id
	integer	character varying (40)	integer	character varying (30)	character varying (30)	bigint	integer	character varying (30)	character varying (30)	integer
1	4	Женя	20230322	улица Яхтенная 15	whale@gmail.com	89112442323	103	действующий	-	

рисунок №2 - первый запрос

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"."Library_card" t  
           ON ("Reader".reader_id = t.reader_id)  
     LEFT JOIN "Library"."Subscription" a  
           ON (t.card_id = a.card_id)  
     WHERE return_date > to_d ) P1  
ON "Library"."Reader".reader_id=P1.reader_id  
INNER JOIN  
    (SELECT "Reader".reader_id,  
           COUNT(copy_id)  
     FROM "Library"."Reader"  
     INNER JOIN "Library"."Library_card"  
           ON "Reader".reader_id = "Library_card".reader_id  
     INNER JOIN "Library"."Subscription"  
           ON "Library_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
```

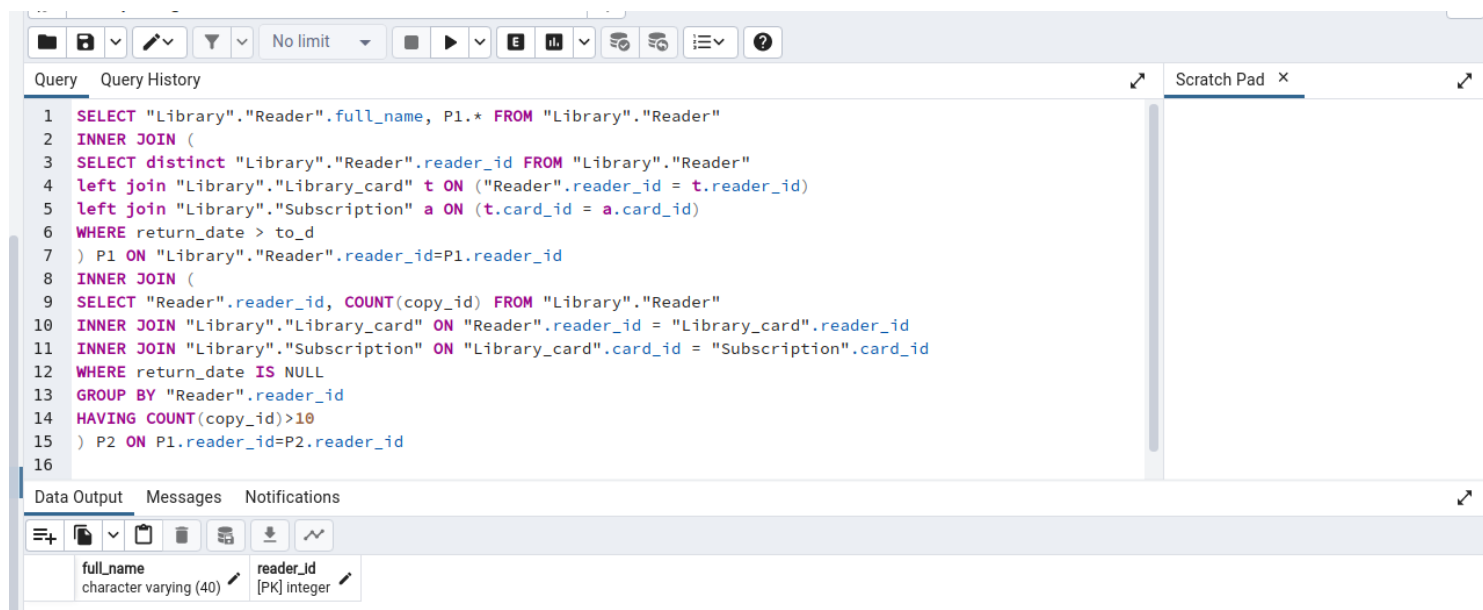


рисунок №3 - второй запрос

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

```

SELECT P2.*
FROM "Library"."Reader"
INNER JOIN
    (SELECT DISTINCT "Library"."Reader".reader_id
    FROM "Library"."Reader"
    LEFT JOIN "Library"."Library_card" t
        ON ("Reader".reader_id = t.reader_id)
    LEFT JOIN "Library"."Subscription" a
        ON (t.card_id = a.card_id)
    WHERE return_date > to_d ) P1
    ON "Library"."Reader".reader_id=P1.reader_id
INNER JOIN
    (SELECT "Reader".reader_id,
        COUNT(copy_id)
    FROM "Library"."Reader"
    INNER JOIN "Library"."Library_card"
        ON "Reader".reader_id = "Library_card".reader_id

```

```

INNER JOIN "Library"."Subscription"
      ON "Library_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

```

The screenshot shows a database query editor window titled "Library/ivan@local". The interface includes a toolbar with various icons for file operations, query execution, and settings. The main area is divided into two panes: "Query" and "Query History". The "Query" pane contains a complex SQL query with 15 lines of code. The "Query History" pane is empty. Below the query panes, there are tabs for "Data Output", "Messages", and "Notifications". The "Data Output" tab is active, showing a table with two columns: "reader_id" (PK Integer) and "count" (bigint). The table is currently empty.

```

1  SELECT P2.* FROM "Library"."Reader"
2  INNER JOIN (
3  SELECT distinct "Library"."Reader".reader_id FROM "Library"."Reader"
4  left join "Library"."Library_card" t ON ("Reader".reader_id = t.reader_id)
5  left join "Library"."Subscription" a ON (t.card_id = a.card_id)
6  WHERE return_date > to_d
7  ) P1 ON "Library"."Reader".reader_id=P1.reader_id
8  INNER JOIN (
9  SELECT "Reader".reader_id, COUNT(copy_id) FROM "Library"."Reader"
10 INNER JOIN "Library"."Library_card" ON "Reader".reader_id = "Library_card".reader_id
11 INNER JOIN "Library"."Subscription" ON "Library_card".card_id = "Subscription".card_id
12 WHERE return_date IS NULL
13 GROUP BY "Reader".reader_id
14 HAVING COUNT(copy_id)>10
15 ) P2 ON P1.reader_id=P2.reader_id

```

reader_id	count
[PK] Integer	bigint

рисунок №4 - запрос третий

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

```
SELECT name_b,  
       author  
FROM "Library"."Copy"  
LEFT JOIN "Library"."Edition"  
    ON ("Edition".library_edition_code = "Copy".library_edition_code)  
LEFT JOIN "Library"."Book_information"  
    ON ("Book_information".book_code = "Edition".book_code)  
GROUP BY name_b, author  
HAVING COUNT("Copy".copy_id) = 1
```

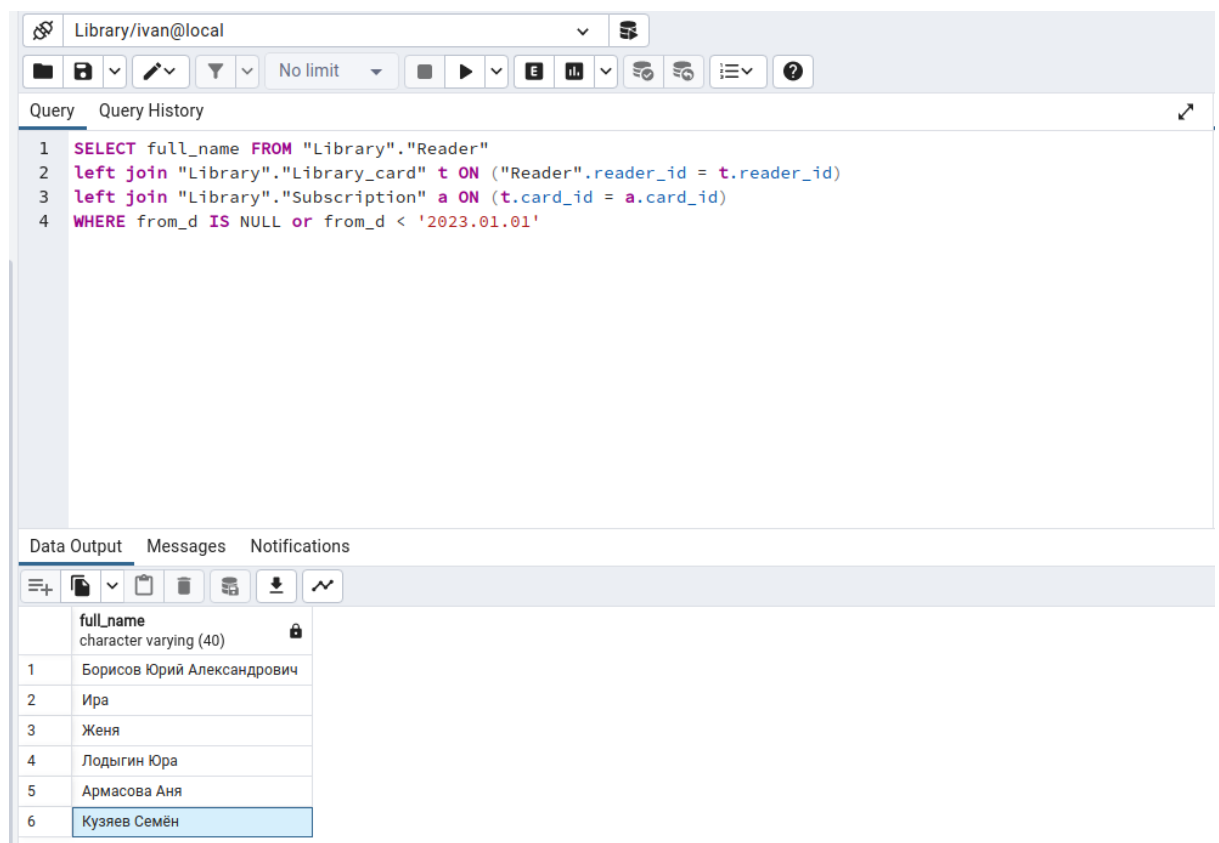
The screenshot shows a database management tool interface. On the left is a tree view of the database structure, including servers, local databases, and various schemas like Library, Casts, Catalogs, etc. The main window displays a SQL query in the 'Query' tab. Below the query editor is a 'Data Output' tab showing the results of the query as a table with two columns: 'name_b' and 'author'. The results list six books with their respective authors.

	name_b character varying (30)	author character varying (50)
1	Физика	д. Джанколи
2	Игрок	Ф.М. Достоевский
3	Преступление и наказание	Ф.М. Достоевский
4	Заводной апельсин	Энтони Берджесс
5	Бесы	Ф.М. Достоевский
6	Мартин Иден	Джек Лондон

рисунок №5 - четвертый запрос

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

```
SELECT full_name
FROM "Library"."Reader"
LEFT JOIN "Library"."Library_card" t
      ON ("Reader".reader_id = t.reader_id)
LEFT JOIN "Library"."Subscription" a
      ON (t.card_id = a.card_id)
WHERE from_d IS NULL
      OR from_d < '2023.01.01'
```



The screenshot shows a database client window titled "Library/ivan@local". The "Query" tab is active, displaying the following SQL query:

```
1 SELECT full_name FROM "Library"."Reader"
2 left join "Library"."Library_card" t ON ("Reader".reader_id = t.reader_id)
3 left join "Library"."Subscription" a ON (t.card_id = a.card_id)
4 WHERE from_d IS NULL or from_d < '2023.01.01'
```

The "Data Output" tab is also visible, showing the results of the query in a table with two columns: "full_name" (character varying (40)) and an unnamed column. The results are as follows:

	full_name	
1	Борисов Юрий Александрович	
2	Ира	
3	Женя	
4	Лодыгин Юра	
5	Армасова Аня	
6	Кузьяев Семён	

рисунок №6 - пятый запрос

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

```
SELECT knowledge_lvl,  
        COUNT(reader_id)  
FROM "Library"."Reader"  
GROUP BY knowledge_lvl
```



рисунок №7 - шестой запрос

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 name_b,  
       to_d  
FROM "Library"."Subscription"  
LEFT JOIN "Library"."Copy"  
      ON ("Copy".copy_id = "Subscription".copy_id)  
LEFT JOIN "Library"."Edition"  
      ON ("Edition".library_edition_code =  
"Copy".library_edition_code)  
LEFT JOIN "Library"."Book_information"  
      ON ("Book_information".book_code = "Edition".book_code)  
WHERE name_b LIKE '%Программирование на C#%'  
      AND storage_location_id IS NULL  
      AND to_d <= (date(now())+3)
```

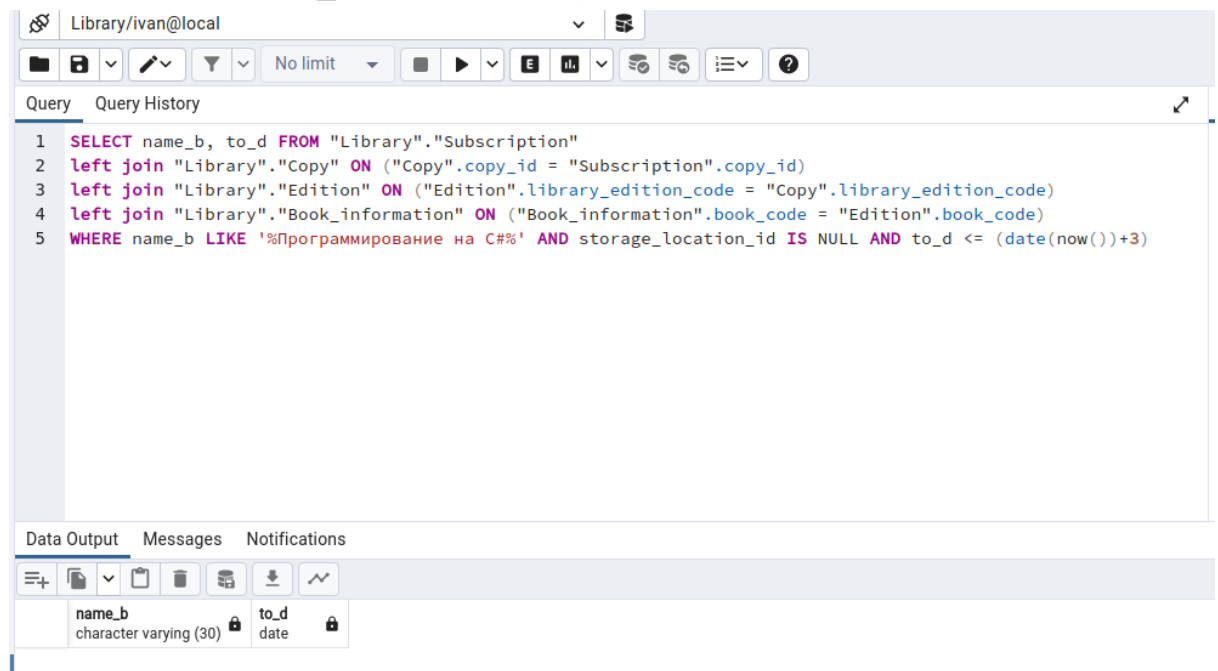


рисунок №8 - седьмой запрос

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"."Library_card" t
            ON ("Reader".reader_id = t.reader_id)
        LEFT JOIN "Library"."Subscription" a
            ON (t.card_id = a.card_id)
        WHERE return_date > to_d ) P1
    ON "Library"."Reader".reader_id=P1.reader_id
    INNER JOIN
        (SELECT "Reader".reader_id,
        COUNT(copy_id)
        FROM "Library"."Reader"
        INNER JOIN "Library"."Library_card"
            ON "Reader".reader_id = "Library_card".reader_id
        INNER JOIN "Library"."Subscription"
            ON "Library_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
```

The screenshot shows a database IDE interface. The top bar indicates the user is 'Library/ivan@local'. Below the toolbar, the 'Query' tab is active, displaying a SQL query that creates a view named 'debtors'. The query uses nested subqueries and joins to identify readers who have borrowed books. The 'Data Output' tab at the bottom shows the results of the query, which are two rows of data.

```

1 CREATE VIEW debtors AS
2 (SELECT "Library"."Reader".full_name, P2.* FROM "Library"."Reader"
3 INNER JOIN (
4 SELECT distinct "Library"."Reader".reader_id FROM "Library"."Reader"
5 left join "Library"."Library_card" t ON ("Reader".reader_id = t.reader_id)
6 left join "Library"."Subscription" a ON (t.card_id = a.card_id)
7 WHERE return_date > to_d
8 ) P1 ON "Library"."Reader".reader_id=P1.reader_id
9 INNER JOIN (
10 SELECT "Reader".reader_id, COUNT(copy_id) FROM "Library"."Reader"
11 INNER JOIN "Library"."Library_card" ON "Reader".reader_id = "Library_card".reader_id
12 INNER JOIN "Library"."Subscription" ON "Library_card".card_id = "Subscription".card_id
13 GROUP BY "Reader".reader_id
14 HAVING COUNT(copy_id)>=1
15 ) P2 ON P1.reader_id=P2.reader_id);
16
17 SELECT * FROM debtors
  
```

	full_name character varying (40)	reader_id integer	count bigint
1	Женя	4	1
2	Алла Владимировна	6	1

рисунок №9 - первое представление

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

```

CREATE VIEW popular_books AS
SELECT
  "Book_information".book_code,
  "Book_information".name_b
FROM "Library"."Book_information"
INNER JOIN
  (SELECT "Book_information".book_code,
    count(copy_id) AS total
  FROM "Library"."Book_information"
  INNER JOIN "Library"."Edition"
    ON "Book_information".book_code="Edition".book_code
  INNER JOIN "Library"."Copy"
    ON
      "Edition".library_edition_code="Copy".library_edition_code
  GROUP BY "Book_information".book_code ) ALL_BOOKS
ON ALL_BOOKS.book_code="Book_information".book_code
LEFT JOIN
  (SELECT "Book_information".book_code,
  
```

```

COUNT(subscription_id) AS given
FROM "Library"."Book_information"
INNER JOIN "Library"."Edition"
ON "Book_information".book_code="Edition".book_code
INNER JOIN "Library"."Copy"
ON
"Edition".library_edition_code="Copy".library_edition_code
INNER JOIN "Library"."Subscription"
ON "Copy".copy_id="Subscription".copy_id
WHERE "Subscription".return_date IS NULL
GROUP BY "Book_information".book_code )
GIVEN_BOOKS
ON ALL_BOOKS.book_code=GIVEN_BOOKS.book_code
WHERE total=coalesce(given,
0);SELECT *
FROM popular_books

```

The screenshot shows a database management tool interface. The top bar indicates the user is 'Library/ivan@local'. Below the toolbar, the 'Query' tab is active, displaying a SQL script. The script creates a view named 'popular_books' and then selects data from it. The 'Data Output' tab at the bottom shows the results of the query.

Query:

```

1 CREATE VIEW popular_books AS
2 SELECT "Book_information".book_code, "Book_information".name_b
3 FROM "Library"."Book_information"
4 INNER JOIN (
5 SELECT "Book_information".book_code, count(copy_id) AS total FROM "Library"."Book_information"
6 INNER JOIN "Library"."Edition" ON "Book_information".book_code="Edition".book_code
7 INNER JOIN "Library"."Copy" ON "Edition".library_edition_code="Copy".library_edition_code
8 GROUP BY "Book_information".book_code
9 ) ALL_BOOKS ON ALL_BOOKS.book_code="Book_information".book_code
10 LEFT JOIN (
11 SELECT "Book_information".book_code, COUNT(subscription_id) as given FROM "Library"."Book_information"
12 INNER JOIN "Library"."Edition" ON "Book_information".book_code="Edition".book_code
13 INNER JOIN "Library"."Copy" ON "Edition".library_edition_code="Copy".library_edition_code
14 INNER JOIN "Library"."Subscription" ON "Copy".copy_id="Subscription".copy_id
15 WHERE "Subscription".return_date IS NULL
16 GROUP BY "Book_information".book_code
17 ) GIVEN_BOOKS ON ALL_BOOKS.book_code=GIVEN_BOOKS.book_code
18 WHERE total=coalesce(given, 0);
19 SELECT * FROM popular_books

```

Data Output:

	book_code integer	name_b character varying (30)
1	7	Превращение

рисунок №10 - представление второе

Задание 3. Запросы на модификацию данных

3.1 Update с подзапросом

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

Код SQL:

```
UPDATE "Library"."Stored" SET status='занята'
WHERE copy_id IN
    (SELECT "Copy".copy_id
     FROM "Library"."Copy"
     WHERE "Copy".copy_id IN
        (SELECT "Subscription".copy_id
         FROM "Library"."Subscription"
         WHERE return_date IS NULL
         GROUP BY "Subscription".copy_id)
     GROUP BY "Copy".copy_id)
```

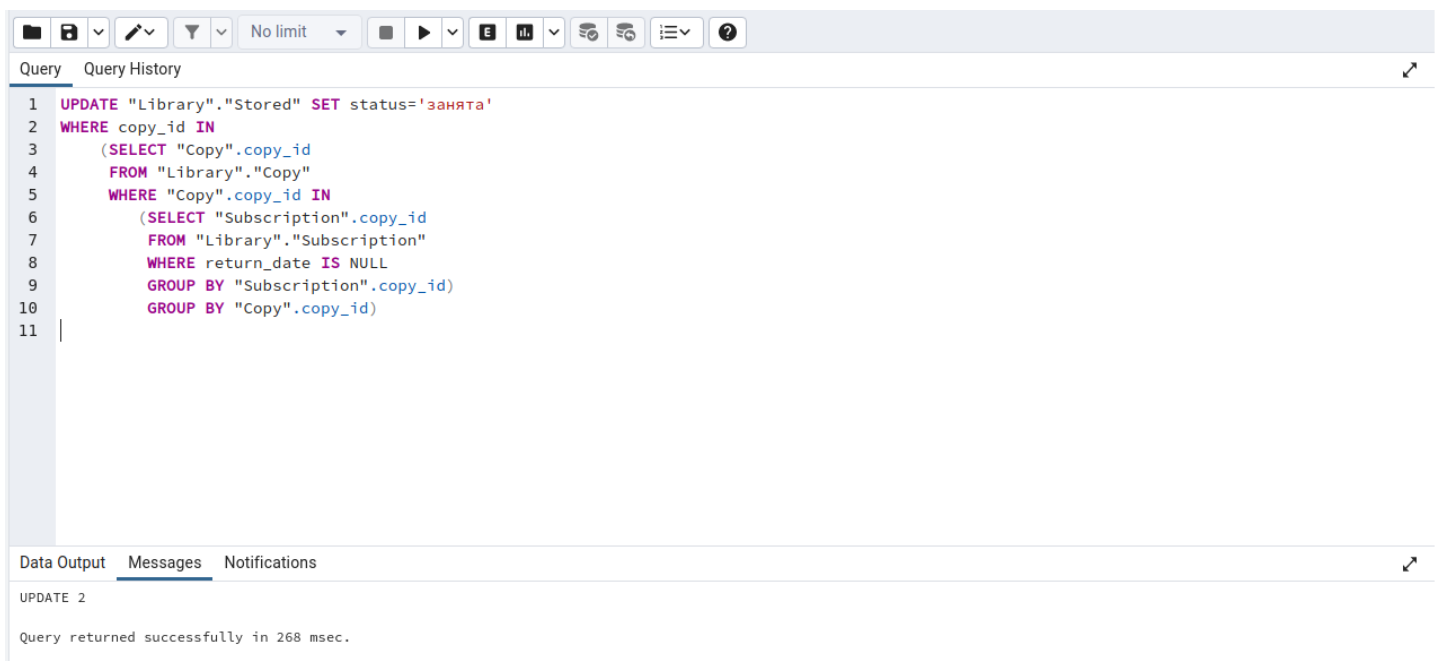


рисунок №11 - update с подзапросом

Query

Query History

1

SELECT * FROM "Library"."Stored"

2

ORDER BY stored_id ASC

Data Output

Messages

Notifications

	copy_id integer	storage_id integer	stored_from date	stored_until date	status character varying (30)	stored_id [PK] integer
1	1	1	2021-12-12	2021-12-31	в наличии	1
2	2	2	2021-10-12	2022-01-29	в наличии	2
3	3	3	2021-11-13	2022-02-05	занята	3
4	4	4	2021-09-01	2021-11-11	в наличии	4
5	39	5	2021-07-23	2025-07-23	в наличии	6
6	40	6	2021-06-21	2025-07-23	в наличии	7
7	42	7	2020-06-20	2025-06-13	занята	9
8	43	8	2020-05-10	2025-06-13	занята	10
9	44	9	2020-04-10	2025-08-13	в наличии	11
10	45	10	2020-02-10	2025-09-13	в наличии	12

рисунок №12 - таблица после update запроса

3.2 Delete с подзапросом

Запрос: Удалить карту читателя который суммарно получил штрафов больше 1000 рублей.

Код SQL:

DELETE

FROM "Library"."Library_card"

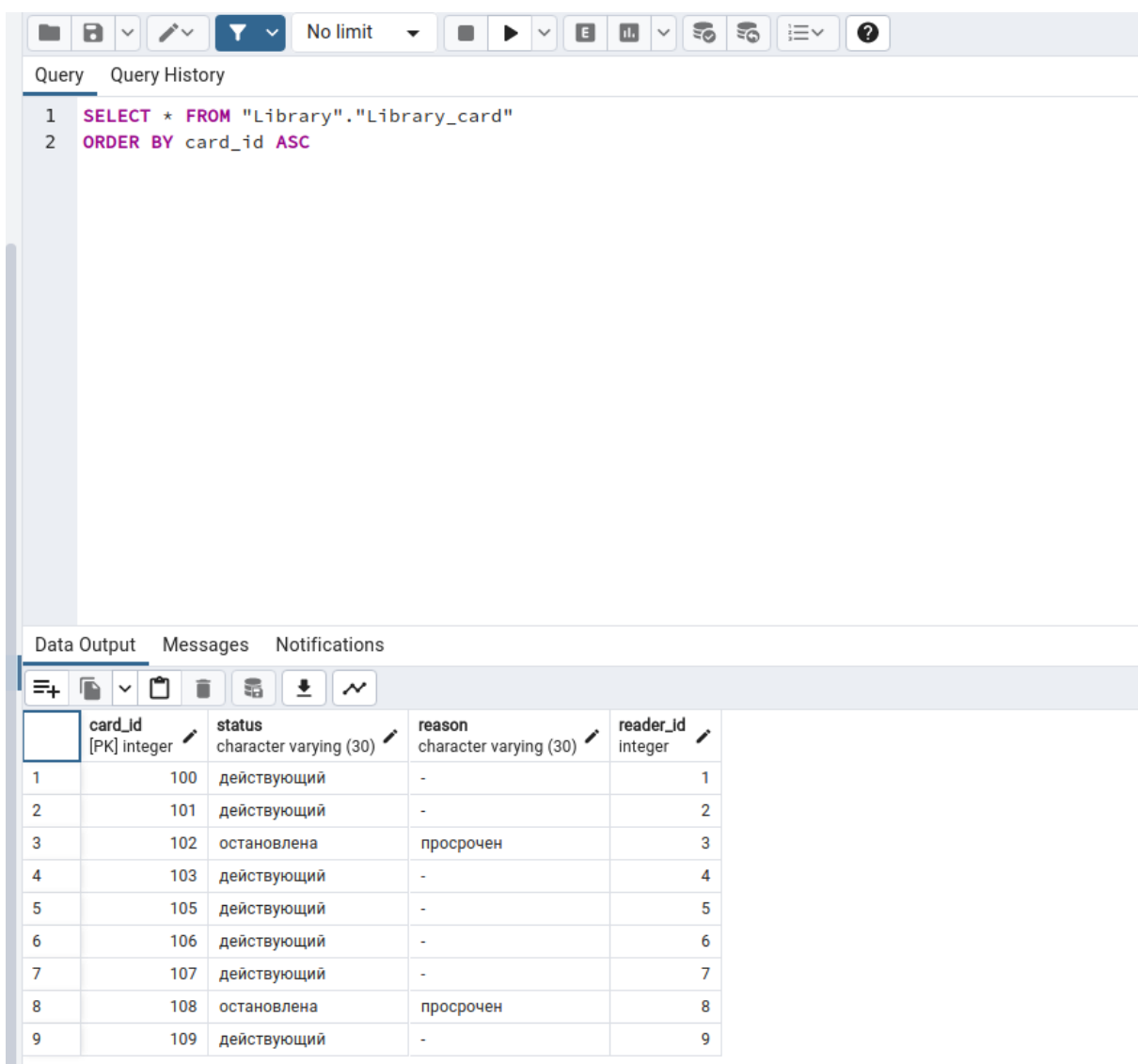
WHERE "Library_card".card_id IN

(SELECT "Subscription".card_id

FROM "Library"."Subscription"

GROUP BY "Subscription".card_id

HAVING SUM("Subscription".fine)>1000)



The screenshot shows a database management interface. At the top, there is a toolbar with various icons for file operations, query execution, and settings. Below the toolbar, the 'Query' tab is active, displaying a SQL query. The query is a DELETE statement that removes records from the 'Library_card' table where the 'card_id' is in a subquery. The subquery selects 'card_id' from the 'Subscription' table, grouped by 'card_id', and filtered by a sum of 'fine' greater than 1000. Below the query, the 'Data Output' tab is active, showing a table with 9 rows and 5 columns: 'card_id', 'status', 'reason', and 'reader_id'. The 'card_id' column is marked as a primary key. The data shows various card statuses and reasons for deletion.

	card_id [PK] integer	status character varying (30)	reason character varying (30)	reader_id integer
1	100	действующий	-	1
2	101	действующий	-	2
3	102	остановлена	просрочен	3
4	103	действующий	-	4
5	105	действующий	-	5
6	106	действующий	-	6
7	107	действующий	-	7
8	108	остановлена	просрочен	8
9	109	действующий	-	9

рисунок №13 - таблица до запроса

No limit

E

Query

Query History

1

SELECT * FROM "Library"."Library_card"

2

ORDER BY card_id ASC

Data Output

Messages

Notifications

	card_id [PK] integer	status character varying (30)	reason character varying (30)	reader_id integer
1	100	действующий	-	1
2	101	действующий	-	2
3	102	остановлена	просрочен	3
4	105	действующий	-	5
5	106	действующий	-	6
6	107	действующий	-	7
7	108	остановлена	просрочен	8
8	109	действующий	-	9

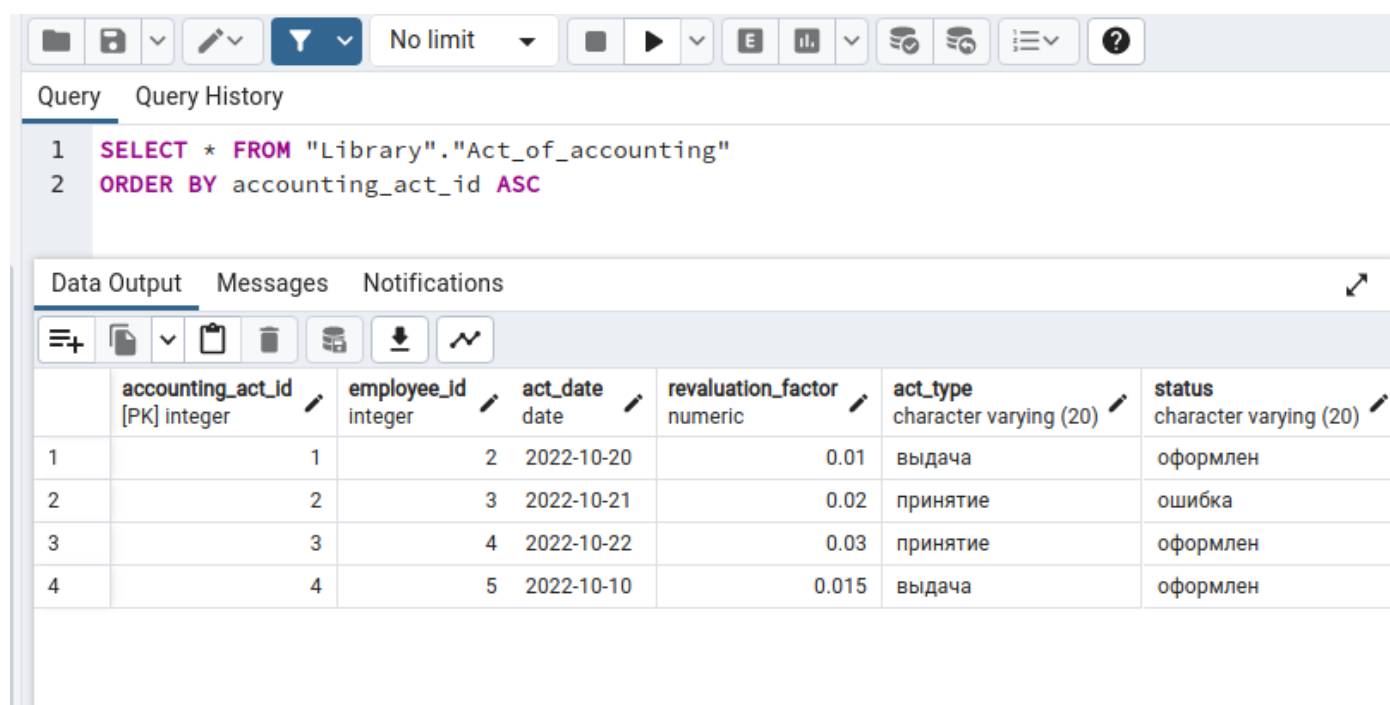
картинка №14 - таблица после delete запроса

3.3 Insert с подзапросом

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

Код SQL:

```
INSERT INTO "Library"."Act_of_accounting"( employee_id, act_date,
revaluation_factor, act_type, status)
SELECT employee_id,
        '2023-05-05', 0.002, 'принятие', 'оформлен'
FROM "Library"."Librarian"
WHERE "Librarian".full_name = 'Евгения'
```



The screenshot shows a database management interface. At the top, there is a toolbar with various icons for file operations, query execution, and settings. Below the toolbar, the 'Query' tab is active, displaying the following SQL query:

```
1 SELECT * FROM "Library"."Act_of_accounting"
2 ORDER BY accounting_act_id ASC
```

Below the query, the 'Data Output' tab is active, showing a table with 7 columns and 4 rows of data. The columns are: **accounting_act_id** (PK) integer, **employee_id** integer, **act_date** date, **revaluation_factor** numeric, **act_type** character varying (20), and **status** character varying (20). The data rows are as follows:

	accounting_act_id [PK] integer	employee_id integer	act_date date	revaluation_factor numeric	act_type character varying (20)	status character varying (20)
1	1	2	2022-10-20	0.01	выдача	оформлен
2	2	3	2022-10-21	0.02	принятие	ошибка
3	3	4	2022-10-22	0.03	принятие	оформлен
4	4	5	2022-10-10	0.015	выдача	оформлен

рисунок №15 - таблица до запроса

<div> <div> <div>📁</div> <div>💾</div> <div>▼</div> <div>✎</div> <div>▼</div> <div>🔍</div> <div>▼</div> <div>No limit</div> <div>▼</div> <div>📄</div> <div>▶</div> <div>▼</div> <div>E</div> <div>il.</div> <div>▼</div> <div>🔄</div> <div>🔄</div> <div>⋮</div> <div>?</div> </div> </div>							
<div>Query Query History</div>							
<div> <div>1</div> <div>SELECT * FROM "Library"."Act_of_accounting"</div> <div>2</div> <div>ORDER BY accounting_act_id ASC</div> </div>							
<div>Data Output Messages Notifications</div>							
<div> <div>⋮</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>🗑️</div> <div>🔄</div> <div>⬇️</div> <div>📈</div> </div>							
	accounting_act_id [PK] integer	employee_id integer	act_date date	revaluation_factor numeric	act_type character varying (20)	status character varying (20)	
1	1	2	2022-10-20	0.01	выдача	оформлен	
2	2	3	2022-10-21	0.02	принятие	ошибка	
3	3	4	2022-10-22	0.03	принятие	оформлен	
4	4	5	2022-10-10	0.015	выдача	оформлен	
5	5	4	2023-05-05	0.002	принятие	оформлен	

рисунок №16 - таблица после insert запроса

4. Запросы с индексами.

4.1 Запрос без индекса.

Код:

```

SELECT name_b,
       author
FROM "Library"."Copy"
LEFT JOIN "Library"."Edition"
  ON ("Edition".library_edition_code = "Copy".library_edition_code)
LEFT JOIN "Library"."Book_information"
  ON ("Book_information".book_code = "Edition".book_code)
GROUP BY name_b, author
HAVING COUNT("Copy".copy_id) = 1

```

План запроса без индекса:

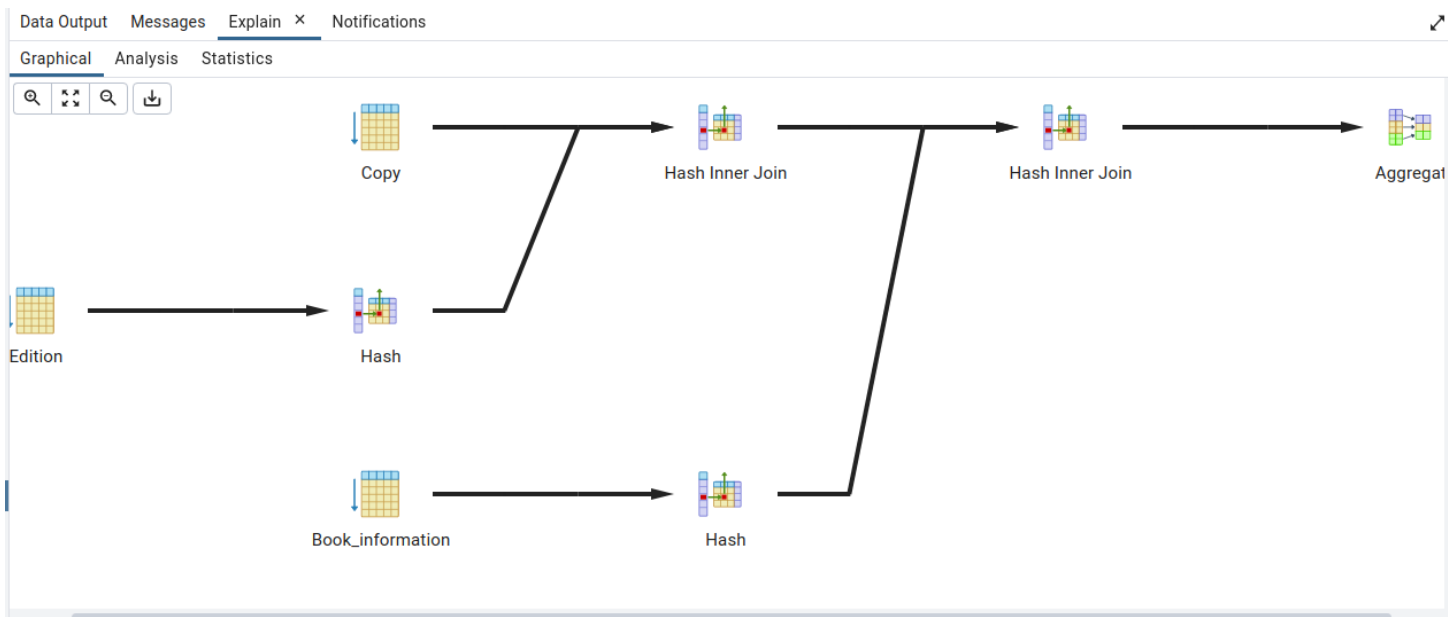


рисунок №17 - графический план запроса

```
{
  "Plan": {
    "Node Type": "Aggregate",
    "Strategy": "Hashed",
    "Partial Mode": "Simple",
    "Parallel Aware": false,
    "Async Capable": false,
    "Group Key": ["\"Book_information\".name_b", "\"Book_information\".author"],
    "Filter": "(count(\"Copy\".copy_id) = 1)",
    "Plans": [
      {
        "Node Type": "Hash Join",
        "Parent Relationship": "Outer",
        "Parallel Aware": false,
        "Async Capable": false,
        "Join Type": "Inner",
        "Inner Unique": true,
        "Hash Cond": "(\"Edition\".book_code = \"Book_information\".book_code)",
        "Plans": [
          {
            "Node Type": "Hash Join",
            "Parent Relationship": "Outer",
            "Parallel Aware": false,
            "Async Capable": false,
            "Join Type": "Inner",
            "Inner Unique": true,
            "Hash Cond": "(\"Copy\".library_edition_code)::text = (\"Edition\".library_edition_code)::text",
            "Plans": [
              {
                "Node Type": "Seq Scan",
                "Parent Relationship": "Outer",
                "Parallel Aware": false,
                "Async Capable": false,
                "Relation Name": "Copy",
                "Alias": "Copy"
              },
              {
                "Node Type": "Hash",
                "Parent Relationship": "Inner",
                "Parallel Aware": false,
                "Async Capable": false,
                "Plans": [
                  {
                    "Node Type": "Seq Scan",
                    "Parent Relationship": "Outer",
                    "Parallel Aware": false,
                    "Async Capable": false,
                    "Relation Name": "Edition",
                    "Alias": "Edition"
                  }
                ]
              }
            ]
          }
        ]
      }
    ]
  }
}
```

рисунок №18 - аналитический план запроса

Total rows: 1 of 1	Query complete 00:00:00.202
--------------------	-----------------------------

рисунок №19 - время запроса

Запрос с индексом.

Добавление индекса:

CREATE index "index_book_code" ON "Library"."Edition"(book_code)

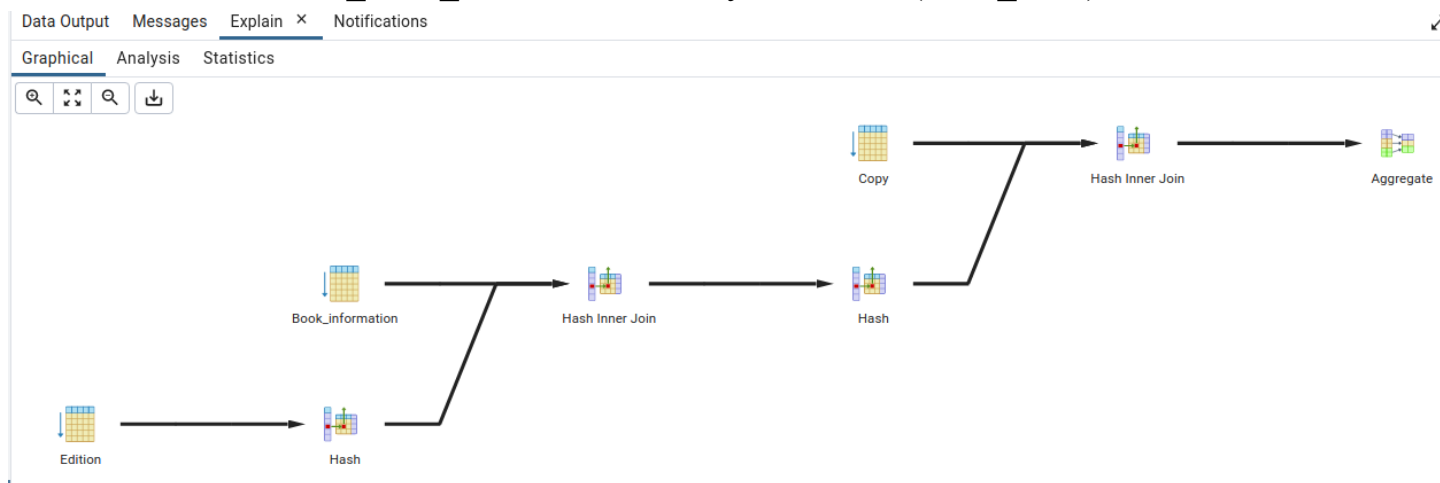


рисунок №20 - графический план запроса с индексом

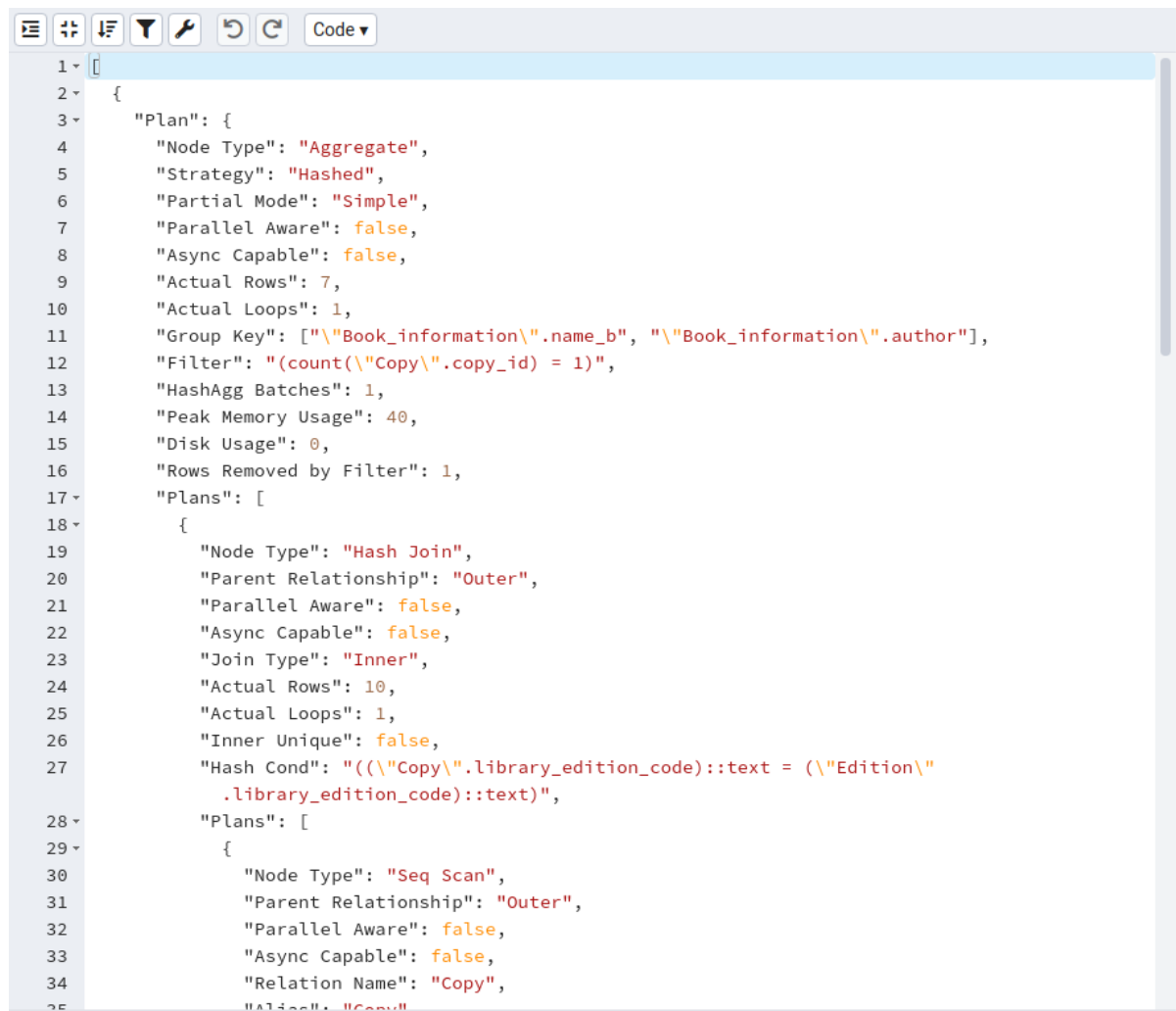


рисунок №21 - аналитический план запроса

Total rows: 1 of 1 Query complete 00:00:00.159

рисунок №22 - время запроса с индексом

4.2 Запрос с индексом №2

Код запроса:

```
SELECT knowledge_lvl,  
       COUNT(reader_id)  
FROM "Library"."Reader"  
GROUP BY knowledge_lvl
```

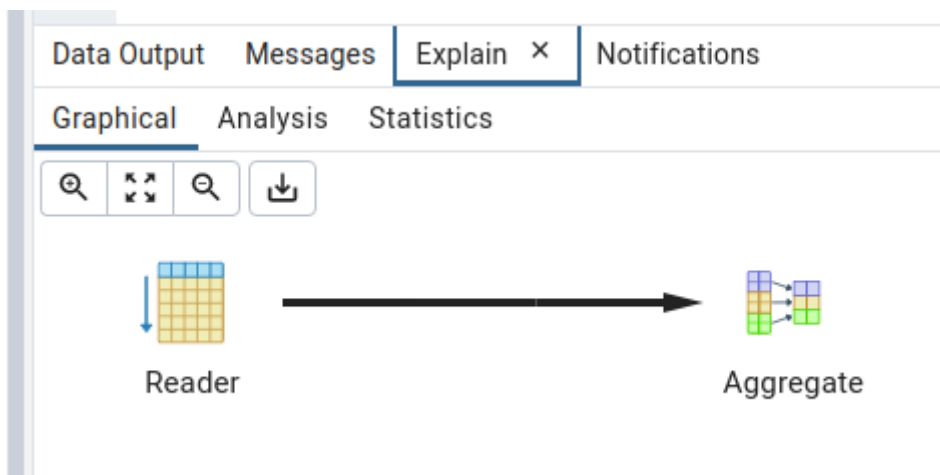


рисунок №23 - графическое отображение запроса без индекса



```
1 {
2   {
3     "Plan": {
4       "Node Type": "Aggregate",
5       "Strategy": "Hashed",
6       "Partial Mode": "Simple",
7       "Parallel Aware": false,
8       "Async Capable": false,
9       "Actual Rows": 3,
10      "Actual Loops": 1,
11      "Group Key": ["knowledge_lvl"],
12      "HashAgg Batches": 1,
13      "Peak Memory Usage": 24,
14      "Disk Usage": 0,
15      "Plans": [
16        {
17          "Node Type": "Seq Scan",
18          "Parent Relationship": "Outer",
19          "Parallel Aware": false,
20          "Async Capable": false,
21          "Relation Name": "Reader",
22          "Alias": "Reader",
23          "Actual Rows": 9,
24          "Actual Loops": 1
25        }
26      ]
27    },
28    "Triggers": [
29    ]
30  }
31 }
```

Ln: 1 Col: 1

рисунок №24 - аналитический вид запроса

Total rows: 1 of 1	Query complete 00:00:00.128
--------------------	-----------------------------

время запроса без индекса

Запрос с индексом.

Добавление индекса:

CREATE index "index_reader_id" ON "Library"."Reader"(reader_id)



рисунок №25 - графическое представление запроса

```
1 {  
2   {  
3     "Plan": {  
4       "Node Type": "Aggregate",  
5       "Strategy": "Hashed",  
6       "Partial Mode": "Simple",  
7       "Parallel Aware": false,  
8       "Async Capable": false,  
9       "Group Key": ["knowledge_lvl"],  
10      "Plans": [  
11        {  
12          "Node Type": "Seq Scan",  
13          "Parent Relationship": "Outer",  
14          "Parallel Aware": false,  
15          "Async Capable": false,  
16          "Relation Name": "Reader",  
17          "Alias": "Reader"  
18        }  
19      ]  
20    }  
21  }  
22 }
```

Ln: 1 Col: 1

X Cancel

рисунок №26 - аналитическое представление запроса

Total rows: 1 of 1	Query complete 00:00:00.099
--------------------	-----------------------------

время затраченное на запрос с индексом

Вывод: в ходе выполнения лабораторной работы были созданы запросы на выборку данных к PostgreSQL согласно индивидуальному заданию часть 2 и 3. Были созданы 3 запроса на модификацию данных (INSERT, UPDATE, DELETE) с использованием подзапросов. Были изучены графические представления запроса (EXPLAIN). Выполнены все шаги указанные в практическом задании. Получилось удачно познакомиться с представлениями, индексами, а также реализацией запросов на модификацию данных с подзапросом. Индексы в больших запросов значительно сокращают время выполнения.