

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»**

Автор: Морозов Артём

Факультет: ИКТ

Группа: K3241

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

Оглавление

Цель работы	3
Практическое задание	3
Выполнение практического задания.....	5
Запросы.....	5
Представления.....	12
Создание INSERT, UPDATE и DELETE запросов	14
Создание индексов.....	15
Вывод	18

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

Создать запросы и представления на выборку данных к базе данных PostgreSQL (согласно индивидуальному заданию, часть 2 и 3).

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

Изучить графическое представление запросов и просмотреть историю запросов.

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 3. БД «Библиотека»

Описание предметной области: Каждая книга может храниться в нескольких экземплярах. Для каждого экземпляра известно место его хранения (комната, стеллаж, полка). Читателю не может быть выдано более 3-х книг одновременно. Книги выдаются читателям на срок не более 10 дней. В случае просрочки читателю назначается денежный штраф.

Все издания, поступающие в библиотеку ставятся на библиотечный учет, согласно существующим требованиям. Необходимо хранить информацию, кто из сотрудников поставил экземпляр на учет.

Книги принимаются к учету на основании первичных учетных документов (накладной от поставщика, акта о приеме документов). Если документы поступают на безвозмездной основе (в результате передачи обязательных экземпляров и т. п.), оформляется акт о приеме документов. Документы, поступающие от читателей взамен утерянных и признанные равноценными утраченным, оформляются актом о приеме документов взамен утерянных.

Выбытие документов из библиотеки отражается в учете в связи с физической утратой либо утратой потребительских свойств (по причине ветхости, дефектности, устарелости по содержанию, непрофильности). Непрофильность издания определяется на основании профиля комплектования фонда или иного документа, утверждаемого руководителем библиотеки. При выбытии документов из библиотеки оформляется акт о списании исключенных объектов библиотечного фонда (далее – акт о списании), к которому прилагается список исключаемых объектов библиотечного фонда. В акте о списании отражаются сведения о количестве и общей стоимости исключаемых документов, а также причина списания и направление изданий после выбытия с учета. В прилагаемом к акту списке указываются:

- регистрационный номер и шифр хранения издания;
- краткое библиографическое описание;
- стоимость, зафиксированная в регистре индивидуального учета издания;

– коэффициент переоценки, стоимость после переоценки;

– общая стоимость исключаемых документов.

БД должна содержать следующий минимальный набор сведений: · Автор (фамилия и имя (инициалы) или псевдоним автора издания). · Название (заглавие) издания. · Номер тома (части, книги, выпуска). · Составитель (фамилия и имена (инициалы) каждого из составителей издания). · Язык, с которого выполнен перевод издания. · Вид издания (сборник, справочник, монография ...). · Область знания. · Переводчик (фамилия и инициалы переводчика). · Место издания (город). · Издательство (название издательства). · Год выпуска издания. · Библиотечный шифр (например, ББК 32.973). · Номер (инвентарный номер) экземпляра. · Номер комнаты (помещения для хранения экземпляров). · Номер стеллажа в комнате. · Номер полки на стеллаже. · Цена конкретного экземпляра. · Дата изъятия экземпляра с установленного места. · Номер читательского билета (формуляра). · Фамилия читателя. · Имя читателя. · Отчество читателя. · Паспортные данные. Адрес читателя (фактический). Телефон читателя. Электронная почта читателя.

Дополнить исходные данные информацией о читательском абонементе (выдаче книг).

Задание 1.1 (ЛР 1 БД). Выполните инфологическое моделирование базы данных системы. (Ограничения задать самостоятельно.)

Задание 1.2. Создайте логическую модель БД, используя ИЛМ (задание 1.1). Используйте необходимые средства поддержки целостности данных в СУБД.

Задание 2. Создайте запросы:

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

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

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

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

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

Подсчитать количество читателей библиотеки по уровню образования.

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

Задание 3. Создать представления для администрации библиотеки, содержащие:

сведения о должниках;

сведения о наиболее популярных книгах (все экземпляры находятся на руках у читателей).

Выполнение практического задания

Запросы

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

```
select
    reader.id_reader,
    reader.name,
    reader.surname
from
    reader,
    book_circulation,
    copy_of_book,
    publication,
    book
where
    reader.id_reader = book_circulation.id_reader and
    book_circulation.id_copy_of_book = copy_of_book.id_copy_of_book and
    copy_of_book.id_publication = publication.library_code and
    publication.id_book = book.id_book and
    book.language = 'English' and
    publication.language != 'English' and
    publication.year_of_release > 2000;
```



```

        ((book_circulation.return_date_plan < current_date and
book_circulation.return_date > book_circulation.return_date_plan and
book_circulation.fine = False) or
        book_circulation.fine = True) and
        reader.count_of_books > 9;
/* У меня стоит ограничение на количество книг 10, поэтому делаю запрос на >9

```

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

```

1 select
2     reader.id_reader,
3     reader.name,
4     reader.surname,
5     reader.count_of_books
6 from
7     reader,
8     book_circulation
9 where
10    reader.id_reader = book_circulation.id_reader and
11    ((book_circulation.return_date_plan < current_date and book_circulation.return_date > book_circulation.return_date_plan and book_circulation.fine = False) or
12     book_circulation.fine = True) and
13    reader.count_of_books > 9;
14 /* У меня стоит ограничение на количество книг 10, поэтому делаю запрос на >9

```

The results pane shows 9 rows of data:

	id_reader	name	surname
1	473	Krystalle	Coleridge
2	888	Christophorus	Blest
3	230	Osbourne	Gratrex
4	726	Eryn	Peterson
5	766	Ches	Frackiewicz
6	67	Hyacinth	Grote
7	615	Cord	Butlin
8	348	Artsair	Mac Giolla Pheadair
9	741	Jaclyn	Goulbourne

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

```

select
    reader.id_reader,
    reader.name,
    reader.surname,
    reader.count_of_books
from
    reader,
    book_circulation
where
    reader.id_reader = book_circulation.id_reader and
    ((book_circulation.return_date_plan < current_date and
book_circulation.return_date > book_circulation.return_date_plan and
book_circulation.fine = False) or
    book_circulation.fine = True) and
    reader.count_of_books > 9;
/* У меня стоит ограничение на количество книг 10, поэтому делаю запрос на >9

```

The screenshot shows a SQL IDE with a query editor and a results pane. The query is as follows:

```

1 select
2     reader.id_reader,
3     reader.name,
4     reader.surname,
5     reader.count_of_books
6 from
7     reader,
8     book_circulation
9 where
10    reader.id_reader = book_circulation.id_reader and
11    ((book_circulation.return_date_plan < current_date and book_circulation.return_date > book_circulation.return_date_plan and book_circulation.fine = False) or
12     book_circulation.fine = True) and
13    reader.count_of_books > 9;
14 /* У меня стоит ограничение на количество книг 10, поэтому делаю запрос на >9

```

The results pane shows 9 rows of data:

	id_reader	name	surname
1	473	Krystalle	Coleridge
2	888	Christophorus	Blest
3	230	Osbourne	Gratrex
4	726	Eryn	Peterson
5	766	Ches	Frackiewicz
6	67	Hyacinth	Grote
7	615	Cord	Butlin
8	348	Artair	Mac Giolla Pheadair
9	741	Jaclyn	Goulbourne

4. Вывести список книг, которые находятся в библиотеке в единственном экземпляре.

```

select
    distinct copy_of_book.id_publication
from
    copy_of_book
group by
    copy_of_book.id_publication
having count(id copy of book) = 1;

```


The screenshot shows a SQL playground with a dark theme. At the top, there are tabs for 3.sql, 4.sql, 5.sql, 6.sql, and 7.sql. Below the tabs is a toolbar with icons for running, saving, and other functions. The main area contains a SQL query:

```
1 select
2   distinct copy_of_book.id_publication
3 from
4   copy_of_book
5 group by
6   copy_of_book.id_publication
7 having count(id_copy_of_book) = 1;
```

Below the query editor, there is an 'Output' section showing the results of the query. The output is a table with two columns: 'id_publication' and a value. The table contains 6 rows of data:

	id_publication
1	3
2	6
3	8
4	11
5	14
6	16

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

```
select
  DISTINCT reader.id_reader,
  reader.name,
  reader.surname
from
  book_circulation,
  reader
where
  reader.id_reader = book_circulation.id_reader and
  book_circulation.issue_date < current_date - interval '1' year;
```

4.sql x 5.sql 6.sql 7.sql

Tx: Auto ✓ ↺ Playground

```

1 ✓ select
2     DISTINCT reader.id_reader,
3     reader.name,
4     reader.surname
5 from
6     book_circulation,
7     reader
8 where
9     reader.id_reader = book_circulation.id_reader and
10    book_circulation.issue_date < current_date - interval '1' year;

```

Output Library.public.reader Tx: ✓ ↺

114 rows < > ↺ ⌂ + - ↶ ↷ ⬆ ⬇ Tx: Auto DDL

	id_reader	name	surname
1	374	Kassandra	Kingsmill
2	210	Ellwood	Sleeford
3	791	Leonore	Raccio
4	706	Edgar	Leving
5	404	LeoLine	Voaden
6	30	Kalie	Casazza

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

```

select
    copy_of_book.*
from
    book_circulation,
    copy_of_book,
    publication,
    book
where
    book_circulation.id_copy_of_book = copy_of_book.id_copy_of_book and
    copy_of_book.id_publication = publication.library_code and
    publication.id_book = book.id_book and
    copy_of_book.availability = FALSE and
    book_circulation.return_date_plan <= current_date + interval '3' day and
    book_circulation.return_date_plan >= current_date and
    book.area_of_knowledge = 'Computer Science';

```

5.sql × 6.sql 7.sql

▶ ⌚ Ⓟ ⚙ Tx: Auto ✓ ↺ ■ Playground ▾

1 ✓
2
3
4
5
6
7
8
9
10
11
12
13
14
15

```
select
  copy_of_book.*
from
  book_circulation,
  copy_of_book,
  publication,
  book
where
  book_circulation.id_copy_of_book = copy_of_book.id_copy_of_book and
  copy_of_book.id_publication = publication.library_code and
  publication.id_book = book.id_book and
  copy_of_book.availability = FALSE and
  book_circulation.return_date_plan <= current_date + interval '3' day and
  book_circulation.return_date_plan >= current_date and
  book.area_of_knowledge = 'Computer Science';
```

Output Library.public.copy_of_book × Tx: ✓ ↺

◀ < 0 rows ▾ > ▶ ↺ ⌚ ■ + − ↶ ↷ ⬆

Tx: Auto ▾ DDL ⚙

🔍 id_copy_of_book ▾ 📄 availability ▾ 🔍 id_publication ▾ 📄 condition ▾

11

Представления

1. Сведения о должниках

```
select
    reader.id_reader,
    reader.name,
    reader.surname
from
    reader,
    book_circulation
where
    reader.id_reader = book_circulation.id_reader and
    book_circulation.fine = True;
```

The screenshot shows a SQL playground interface. At the top, there are tabs for '6.sql' and '7.sql'. Below the tabs is a toolbar with icons for running, saving, and other functions. The main area contains a SQL query that has been executed, indicated by a green checkmark on line 1. The query is as follows:

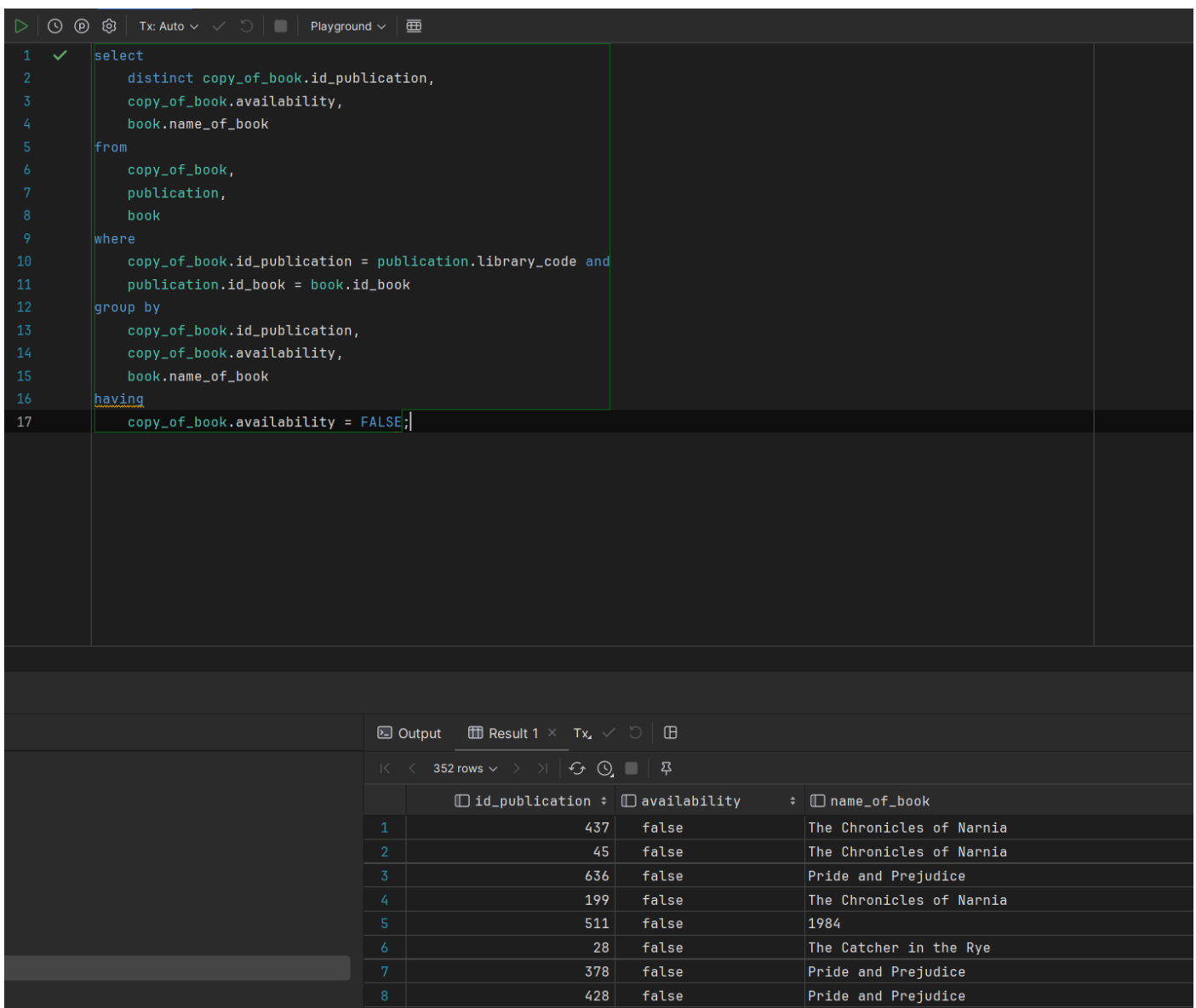
```
1 select
2     reader.id_reader,
3     reader.name,
4     reader.surname
5 from
6     reader,
7     book_circulation
8 where
9     reader.id_reader = book_circulation.id_reader and
10    book_circulation.fine = True;
```

Below the query editor, there is a section for the results. It shows a table with 489 rows. The table has three columns: 'id_reader', 'name', and 'surname'. The first five rows are displayed:

	id_reader	name	surname
1	71	Margalo	Delph
2	530	Silvanus	Ticksall
3	892	Robert	Seeds
4	300	Renaldo	Kasting
5	491	Jeremiah	Lapree

2. Сведения о наиболее популярных книгах (все экземпляры находятся на руках у читателей)

```
select
  distinct copy_of_book.id_publication,
  copy_of_book.availability,
  book.name_of_book
from
  copy_of_book,
  publication,
  book
where
  copy_of_book.id_publication = publication.library_code and
  publication.id_book = book.id_book
group by
  copy_of_book.id_publication,
  copy_of_book.availability,
  book.name_of_book
having
  copy_of_book.availability = FALSE;
```



The screenshot shows a SQL playground interface. The top section contains a SQL query. The bottom section shows the results of the query, which are 352 rows. The results are displayed in a table with columns: id_publication, availability, and name_of_book. The first 8 rows are visible, showing books like 'The Chronicles of Narnia', 'Pride and Prejudice', and 'The Catcher in the Rye'.

	id_publication	availability	name_of_book
1	437	false	The Chronicles of Narnia
2	45	false	The Chronicles of Narnia
3	636	false	Pride and Prejudice
4	199	false	The Chronicles of Narnia
5	511	false	1984
6	28	false	The Catcher in the Rye
7	378	false	Pride and Prejudice
8	428	false	Pride and Prejudice

Создание INSERT, UPDATE и DELETE запросов

UPDATE

```
update
  copy_of_book
set
  condition = 'Fair'
where
  copy_of_book.id_copy_of_book in
  (
    select
      copy_of_book.id_copy_of_book
    from
      copy_of_book, act_of_registration
    where
      copy_of_book.id_copy_of_book = act_of_registration.id_copy_of_book and
      act_of_registration.date_of_signing < date('2000-01-01')
  )
```

INSERT

```
INSERT INTO
  copy_of_book
VALUES
  (1000,
   true,
   (SELECT library_code FROM publication, publishing_house WHERE
publication.id_publishing_house = publishing_house.id_publishing_house and
publishing_house.name = 'Penguin Books' and
publishing_house.compilers = 'Manuel Buesden'),
   'New');
```

DELETE

```
delete from
  copy_of_book
where
  copy_of_book.id_copy_of_book in
  (
    select
      copy_of_book.id_copy_of_book
    from
      copy_of_book,
      act_of_debiting
    where
      copy_of_book.id_copy_of_book = act_of_debiting.id_copy_of_book
  )
```

Создание индексов

Оператор EXPLAIN ANALYZE в запросе без индексов:

Запрос

История запросов

1

explain analyze

2

select

3

reader.id_reader,

4

reader.name,

5

reader.surname,

6

reader.count_of_books

7

from

8

reader,

9

book_circulation

10

where

11

reader.id_reader = book_circulation.id_reader **and**

12

book_circulation.fine = True **and**

13

reader.count_of_books > 9;

Data Output

Сообщения

Notifications

≡+

▼

▼

	QUERY PLAN	
	text	
1	Hash Join (cost=31.79..51.07 rows=50 width=23) (actual time=1.839..2.179 rows=56 loops=1)	
2	Hash Cond: (book_circulation.id_reader = reader.id_reader)	
3	-> Seq Scan on book_circulation (cost=0.00..18.00 rows=489 width=4) (actual time=0.046..0.334 rows=489 loop...	
4	Filter: fine	
5	Rows Removed by Filter: 511	
6	-> Hash (cost=30.50..30.50 rows=103 width=23) (actual time=1.782..1.783 rows=103 loops=1)	
7	Buckets: 1024 Batches: 1 Memory Usage: 14kB	
8	-> Seq Scan on reader (cost=0.00..30.50 rows=103 width=23) (actual time=0.006..1.747 rows=103 loops=1)	
9	Filter: (count_of_books > 9)	
10	Rows Removed by Filter: 897	
11	Planning Time: 5.381 ms	
12	Execution Time: 2.208 ms	

Простые индексы

Запрос

История запросов

```
1 drop index idx_reader_circulation;
2
3 CREATE INDEX idx_reader_circulation ON book_circulation (id_reader);
4 CREATE INDEX idx_reader_circulation1 ON book_circulation (fine);
5
6 explain analyze
7 select
8     reader.id_reader,
9     reader.name,
10    reader.surname,
11    reader.count_of_books
12 from
13     reader,
14     book_circulation
15 where
16     reader.id_reader = book_circulation.id_reader and
17     book_circulation.fine = True and
18     reader.count_of_books > 9;
```

Data Output

Сообщения

Notifications

QUERY PLAN

text

1

Hash Join (cost=31.79..51.07 rows=50 width=23) (actual time=0.125..0.228 rows=56 loops=1)

2

Hash Cond: (book_circulation.id_reader = reader.id_reader)

3

-> Seq Scan on book_circulation (cost=0.00..18.00 rows=489 width=4) (actual time=0.010..0.086 rows=489 loop...

4

Filter: fine

5

Rows Removed by Filter: 511

6

-> Hash (cost=30.50..30.50 rows=103 width=23) (actual time=0.108..0.108 rows=103 loops=1)

7

Buckets: 1024 Batches: 1 Memory Usage: 14kB

8

-> Seq Scan on reader (cost=0.00..30.50 rows=103 width=23) (actual time=0.005..0.095 rows=103 loops=1)

9

Filter: (count_of_books > 9)

10

Rows Removed by Filter: 897

11

Planning Time: 1.574 ms

12

Execution Time: 0.249 ms

Составные индексы

Запрос

История запросов

```
1 drop index idx_reader_circulation;
2
3 CREATE INDEX idx_reader_circulation ON book_circulation (id_reader);
4 CREATE INDEX idx_reader_circulation1 ON book_circulation (fine);
5
6 explain analyze
7 select
8     reader.id_reader,
9     reader.name,
10    reader.surname,
11    reader.count_of_books
12 from
13     reader,
14     book_circulation
15 where
16     reader.id_reader = book_circulation.id_reader and
17     book_circulation.fine = True and
18     reader.count_of_books > 9;
```

Data Output

Сообщения

Notifications

≡+

QUERY PLAN

text

1	Hash Join (cost=31.79..51.07 rows=50 width=23) (actual time=0.125..0.228 rows=56 loops=1)
2	Hash Cond: (book_circulation.id_reader = reader.id_reader)
3	-> Seq Scan on book_circulation (cost=0.00..18.00 rows=489 width=4) (actual time=0.010..0.086 rows=489 loop...
4	Filter: fine
5	Rows Removed by Filter: 511
6	-> Hash (cost=30.50..30.50 rows=103 width=23) (actual time=0.108..0.108 rows=103 loops=1)
7	Buckets: 1024 Batches: 1 Memory Usage: 14kB
8	-> Seq Scan on reader (cost=0.00..30.50 rows=103 width=23) (actual time=0.005..0.095 rows=103 loops=1)
9	Filter: (count_of_books > 9)
10	Rows Removed by Filter: 897
11	Planning Time: 1.574 ms
12	Execution Time: 0.249 ms

Вывод

В ходе этой лабораторной я научился работать с SQL запросами.