

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «Базы данных»**  
**Тема: Реализация базы данных в СУБД PostgreSQL**

Студент гр. 1303

\_\_\_\_\_

Беззубов Д.В.

Преподаватель

\_\_\_\_\_

Заславский М.М.

Санкт-Петербург

2023

## **Цель работы.**

Развернуть локально PostgreSQL, написать запросы для создания и заполнения таблиц, написать запросы к БД, отвечающие на вопросы в заданиях.

## **Задание.**

### Вариант 2

Пусть требуется создать программную систему, предназначенную для работников библиотеки. Такая система должна обеспечивать хранение сведений об имеющихся в библиотеке книгах, о читателях библиотеки и читальных залах. Для каждой книги в БД должны храниться следующие сведения: название книги, автор (ы), издательство, год издания, число экземпляров этой книги в каждом зале библиотеки, а также шифр книги и дата закрепления книги за читателем. Сведения о читателях библиотеки должны включать номер читательского билета, фамилию читателя, номер паспорта, дату рождения, адрес, номер телефон, образование, наличие ученой степени. Читатели закрепляются за определенным залом и могут записываться и выписываться из библиотеки. Библиотека имеет несколько читальных залов, которые характеризуются номером, названием и вместимостью, то есть количеством людей, которые могут одновременно работать в зале. Библиотека может получать новые книги и списывать старые. Шифр книги может измениться в результате переклассификации, а номер читательского билета в результате перерегистрации. Библиотекаря могут потребоваться следующие сведения о текущем состоянии библиотеки:

- Какие книги закреплены за определенным читателем?
- Как называется книга с заданным шифром?
- Какой шифр у книги с заданным названием?
- Когда книга была закреплена за читателем?
- Кто из читателей взял книгу более месяца тому назад?
- За кем из читателей закреплены книги, количество экземпляров которых в библиотеке не превышает 2?

- Какое число читателей пользуется библиотекой?
- Сколько в библиотеке читателей младше 20 лет?

### Выполнение работы.

Создана схема базы данных под названием «*library\_schema*».

```
1 create schema if not exists library_schema;
```

Рисунок 1 – Создание схемы БД «*library\_schema*».

Созданы таблицы «*author*», «*authorbook*», «*book*», «*bookathall*», «*libraryhall*», «*publisher*», «*visitor*», «*visitorhall*» в соответствии со структурой БД и написаны запросы на их заполнение.

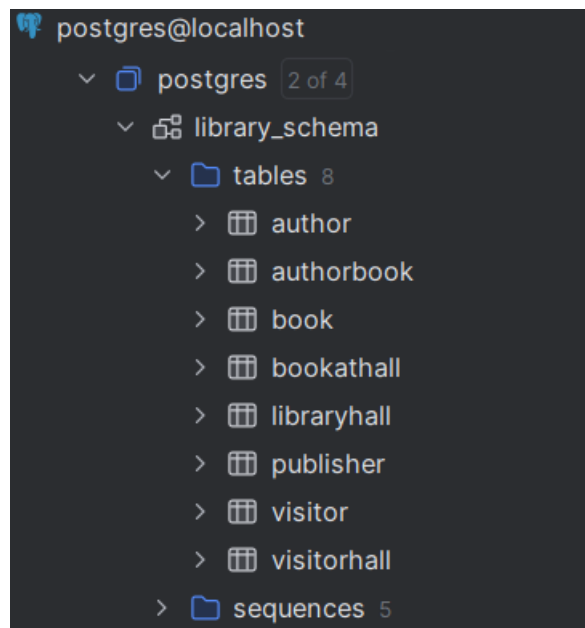


Рисунок 2 – Созданные таблицы.

```

3  create table library_schema.Publisher(
4      id serial primary key,
5      name varchar(50)
6  );
7
8  create table library_schema.Author(
9      id serial primary key,
10     name varchar(30),
11     surname varchar(30)
12 );
13
14 create table library_schema.Book(
15     id serial primary key,
16     publisher_id int not null,
17     foreign key (publisher_id) references library_schema.Publisher(id) on delete cascade,
18     title varchar(50),
19     year varchar(4)
20 );

```

Рисунок 3 – пример DDL для таблиц «*publisher*», «*author*», «*book*»

Написаны запросы на вывод нужных сведений для Библиотекаря:

```

select title from
  library_schema.Book
  inner join library_schema.BookAtHall on library_schema.Book.id = library_schema.BookAtHall.book_id
  inner join library_schema.Visitor on library_schema.BookAtHall.visitor_id = library_schema.Visitor.id
where reader_ticket = 1001;

```

Output postgres.library\_schema.book x

1 row | Tx: Auto | DDL | CSV

	title
1	Война и Мир

Рисунок 2 – Запрос на вывод: книги, закрепленные за определенным читателем.

```
select title from
  library_schema.Book
    inner join library_schema.BookAtHall BAH on library_schema.book.id = BAH.book_id
where code = 2123;
```

Output postgres.library\_schema.book ×

1 row

title

1	1984
---	------

Рисунок 3 – Запрос на вывод: название книги с заданным шифром.

```
select code from
  library_schema.BookAtHall
    inner join library_schema.Book b on b.id = library_schema.BookAtHall.book_id
where title = 'Война и Мир';
```

Output postgres.library\_schema.bookathall ×

1 row

code

1	1123
---	------

Рисунок 4 – Запрос на вывод: шифр у книги с заданным названием.

```
select title, booked_at from
  library_schema.BookAtHall
  inner join library_schema.Book b on b.id = library_schema.BookAtHall.book_id
where booked_at is not null
order by title;
```

Output Result 23 ×

2 rows

	title	booked_at
1	Война и Мир	2023-08-08
2	Идиот	2023-10-15

Рисунок 5 – Запрос на вывод: книга была закреплена за читателем.

```
select surname, reader_ticket, title from
  library_schema.Visitor
  inner join library_schema.BookAtHall b on library_schema.Visitor.id = b.visitor_id
  inner join library_schema.Book b2 on b2.id = b.book_id
where (select current_date - booked_at) > 30;
```

Output Result 24 ×

1 row

	surname	reader_ticket	title
1	Иванов	1001	Война и Мир

Рисунок 6 – Запрос на вывод: кто из читателей взял книгу более месяца тому назад.

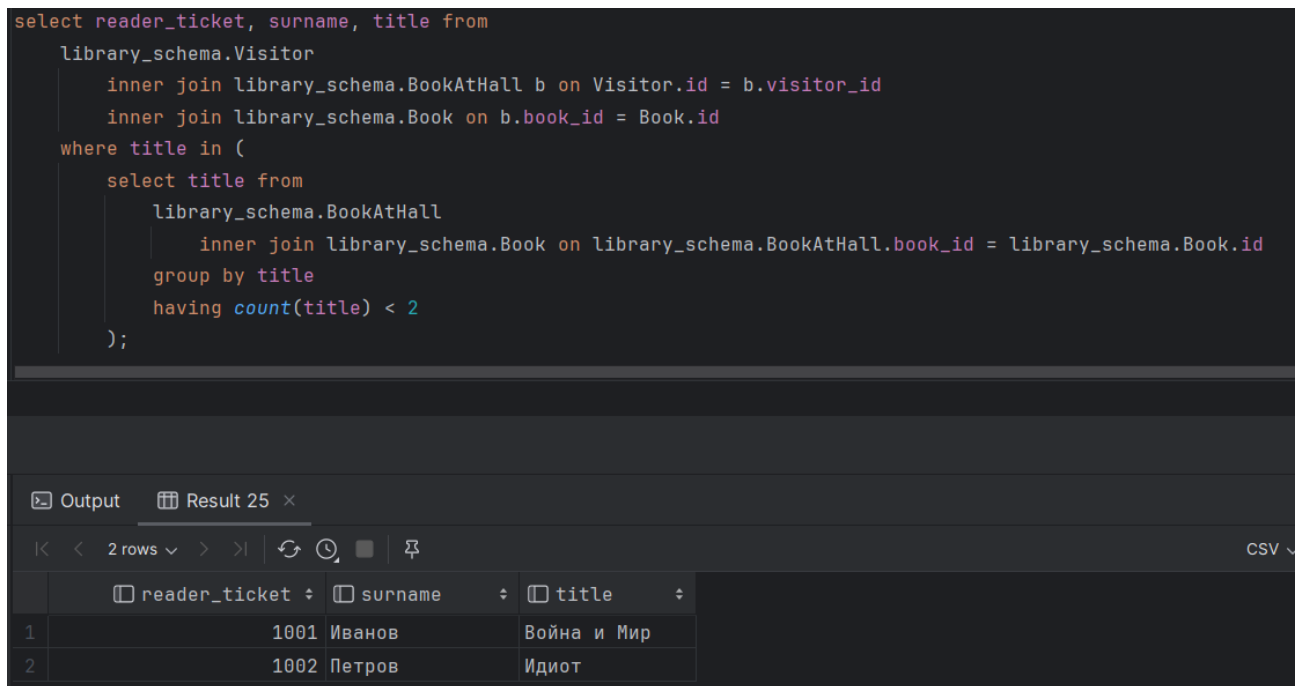


Рисунок 7 – Запрос на вывод: за кем из читателей закреплены книги, количество экземпляров которых в библиотеке не превышает 2.

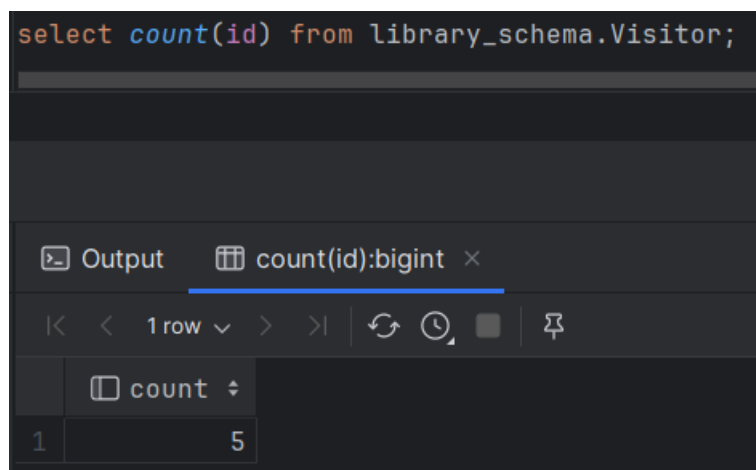
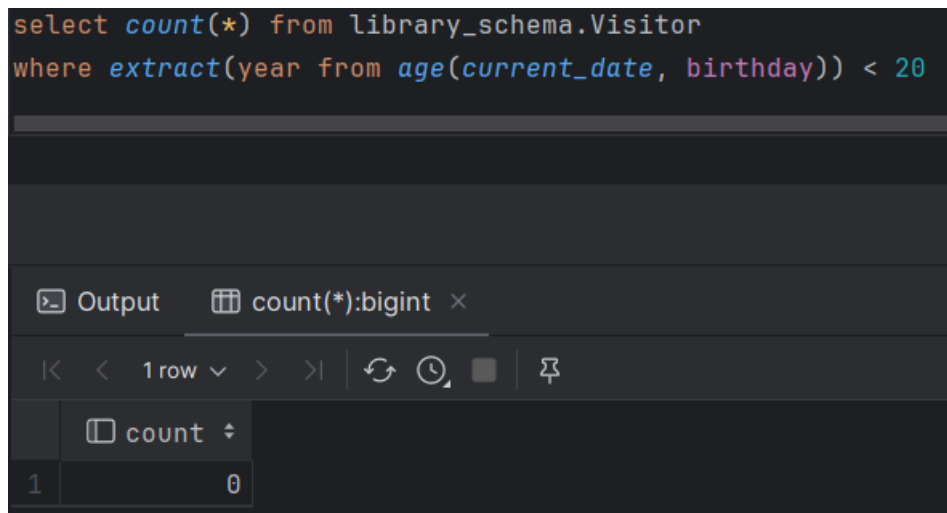


Рисунок 10 – Запрос на вывод: количество читателей, использующих библиотеку

```
select count(*) from library_schema.Visitor
where extract(year from age(current_date, birthday)) < 20
```



The screenshot shows a PostgreSQL query editor with a dark theme. The top pane contains the SQL query: `select count(*) from library_schema.Visitor where extract(year from age(current_date, birthday)) < 20`. The bottom pane is titled "Output" and shows the result of the query. It includes a table with one row and one column labeled "count". The value in the row is 0. The interface also shows a tab for the query, "count(\*) : bigint", and a toolbar with navigation and execution icons.

count
0

Рисунок 11 – Запрос на вывод: количество читателей младше 20 лет

### **Вывод.**

В ходе выполнения работы был развернут локально PostgreSQL, написаны запросы для создания и заполнения таблиц в соответствии со структурой БД, написаны запросы к БД, отвечающие на вопросы в заданиях.



## Приложение А

### Ссылки

Pull Request: <https://github.com/moevm/sql-2023-1303/pull/22>

DB Fiddle: <https://www.db-fiddle.com/f/kaa7snWaTqMz8wkDUSNpwH/0>

## Приложение Б

### Исходный код

Файл lab2.sql:

```
create schema if not exists library_schema;

create table library_schema.Publisher(
    id serial primary key,
    name varchar(50)
);

create table library_schema.Author(
    id serial primary key,
    name varchar(30),
    surname varchar(30)
);

create table library_schema.Book(
    id serial primary key,
    publisher_id int not null,
    foreign key (publisher_id) references
library_schema.Publisher(id) on delete cascade,
    title varchar(50),
    year varchar(4)
);

create table library_schema.AuthorBook(
    author_id int,
    book_id int,
    foreign key (author_id) references library_schema.Author(id)
on delete cascade,
    foreign key (book_id) references library_schema.Book(id) on
delete cascade,
    primary key (author_id, book_id)
);

create table library_schema.LibraryHall(
    id serial primary key,
    name varchar(30),
    capacity int
);

create table library_schema.Visitor(
    id serial primary key,
    reader_ticket int,
    surname varchar(30),
    passport varchar(11),
    birthday date,
    address varchar(50),
    phone varchar(11),
```

```

        educational_stage varchar(30),
        academic_degree bool
    );

create table library_schema.VisitorHall(
    visitor_id int,
    hall_id int,
    foreign key (visitor_id) references library_schema.Visitor(id)
on delete cascade,
    foreign key (hall_id) references
library_schema.LibraryHall(id) on delete cascade,
    primary key (visitor_id, hall_id)
);

create table library_schema.BookAtHall(
    hall_id int,
    book_id int,
    foreign key (hall_id) references
library_schema.LibraryHall(id) on delete cascade,
    foreign key (book_id) references library_schema.Book(id) on
delete cascade,
    visitor_id int,
    foreign key (visitor_id) references library_schema.Visitor(id)
on delete set null,
    code int,
    booked_at date,
    deleted_at date
);

insert into library_schema.Publisher(name) values
    ('Питер'), ('Эксмо'), ('АСТ'), ('Манн, Иванов и Фербер'),
    ('Центрполиграф');

insert into library_schema.Author(name, surname) values
    ('Фёдор', 'Достоевский'),
    ('Лев', 'Толстой'),
    ('Джордж', 'Оруэлл'),
    ('Агата', 'Кристи'),
    ('Джоан', 'Роулинг');

insert into library_schema.Book(publisher_id, title, year) values
    (1, 'Война и Мир', '1869'),
    (1, 'Преступление и наказание', '1866'),
    (2, 'Идиот', '1869'),
    (4, '1984', '1949'),
    (5, 'Анна Каренина', '1877');

insert into library_schema.AuthorBook(author_id, book_id) values
    (1, 2), (1, 3), (2, 1), (2, 5), (3, 4);

insert into library_schema.LibraryHall(name, capacity) values
    ('Классика', 25), ('Зарубежная', 20), ('Фэнтези', 25);

```

```

insert into library_schema.Visitor(reader_ticket, surname,
passport, birthday, address, phone, educational_stage,
academic_degree) values
    (1001, 'Иванов', '1234 678901', '1990-05-15', 'ул. Ленина,
123', '89205678901', 'Высшее', true),
    (1002, 'Петров', '2345 789012', '1985-12-10', 'ул. Пушкина,
45', '89066789012', 'Среднее', false),
    (1003, 'Сидорова', '3456 890123', '1995-03-25', 'пр. Гагарина,
67', '89217890123', 'Высшее', true),
    (1004, 'Козлов', '4567 901234', '1982-08-03', 'ул.
Маяковского, 56', '89208901234', 'Среднее', false),
    (1005, 'Михайлова', '5678 012345', '1998-07-18', 'ул. Кирова,
89', '89219012345', 'Высшее', false);

insert into library_schema.VisitorHall(visitor_id, hall_id) values
    (1, 1), (2, 1), (3, 3), (4, 2), (5, 2);

insert into library_schema.BookAtHall(hall_id, book_id,
visitor_id, code, booked_at) values
    (1, 1, 1, 1123, '2023-08-08'),
    (1, 2, null, 1124, null),
    (1, 3, 2, 1125, '2023-10-15'),
    (2, 4, null, 2123, null),
    (1, 5, null, 1126, null);

select title from
    library_schema.Book
        inner join library_schema.BookAtHall on
library_schema.Book.id = library_schema.BookAtHall.book_id
        inner join library_schema.Visitor on
library_schema.BookAtHall.visitor_id = library_schema.Visitor.id
    where reader_ticket = 1001;

select title from
    library_schema.Book
        inner join library_schema.BookAtHall BAH on
library_schema.book.id = BAH.book_id
    where code = 2123;

select code from
    library_schema.BookAtHall
        inner join library_schema.Book b on b.id =
library_schema.BookAtHall.book_id
    where title = 'Война и Мир';

select title, booked_at from
    library_schema.BookAtHall
        inner join library_schema.Book b on b.id =
library_schema.BookAtHall.book_id
    where booked_at is not null
    order by title;

select surname, reader_ticket, title from

```

```

        library_schema.Visitor
            inner join library_schema.BookAtHall b on
library_schema.Visitor.id = b.visitor_id
            inner join library_schema.Book b2 on b2.id = b.book_id
        where (select current_date - booked_at) > 30;

select reader_ticket, surname, title from
    library_schema.Visitor
        inner join library_schema.BookAtHall b on Visitor.id =
b.visitor_id
        inner join library_schema.Book on b.book_id = Book.id
    where title in (
        select title from
            library_schema.BookAtHall
                inner join library_schema.Book on
library_schema.BookAtHall.book_id = library_schema.Book.id
            group by title
            having count(title) < 2
    );

select count(id) from library_schema.Visitor;

select count(*) from library_schema.Visitor
where extract(year from age(current_date, birthday)) < 20

```