

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

Чубан 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 телефона, 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 «*librarySchema*».

```
create schema if not exists librarySchema;
```

Рисунок 1 – Создание схемы БД «*librarySchema*».

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

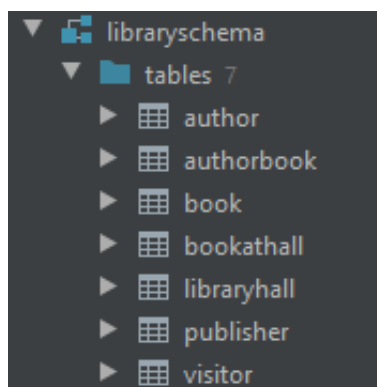


Рисунок 2 – Созданные таблицы.

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

create table librarySchema.Author(
    author_id serial primary key,
    name varchar(30)
);

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

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

Написаны запросы на вывод нужных сведений для Библиотекаря:

```
select title from
  librarySchema.Book
    inner join librarySchema.BookAtHall on librarySchema.Book.book_id = librarySchema.BookAtHall.book_id
    inner join librarySchema.Visitor on librarySchema.BookAtHall.visitor_id = librarySchema.Visitor.visitor_id
  where library_ticket = 1001;
```

Output postgres.libraryschema.book

	title
1	Война и мир

Рисунок 4— Запрос на вывод: книги, закрепленные за определенным читателем.

```
select title from
  librarySchema.Book
    inner join librarySchema.BookAtHall BAH on librarySchema.book.book_id = BAH.book_id
  where code = 1755;
```

Output postgres.libraryschema.book

	title
1	1984

Рисунок 5 – Запрос на вывод: название книги с заданным шифром.

```
select code from
  librarySchema.BookAtHall
    inner join librarySchema.Book b on b.book_id = librarySchema.BookAtHall.book_id
  where title = '1984';
```

Output postgres.libraryschema.bookathall

	code
1	1755

Рисунок 6 – Запрос на вывод: шифр у книги с заданным названием.

```

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

```

Output Result 41

	title	received
1	1984	2023-10-19
2	Анна Каренина	2023-10-20
3	Братья Карамазовы	2023-10-22
4	Война и мир	2023-08-08
5	Преступление и наказание	2023-10-15

Рисунок 7 – Запрос на вывод: когда книга была закреплена за читателем.

```

select surname, library_ticket, title from
  librarySchema.Visitor
  inner join librarySchema.BookAtHall b on librarySchema.Visitor.visitor_id = b.visitor_id
  inner join librarySchema.Book b2 on b2.book_id = b.book_id
where (select current_date - received) > 30;

```

Output Result 46

	surname	library_ticket	title
1	Иванов	1001	Война и мир
2	Козлов	1004	Братья Карамазовы

Рисунок 8 – Запрос на вывод: кто из читателей взял книгу более месяца тому назад.

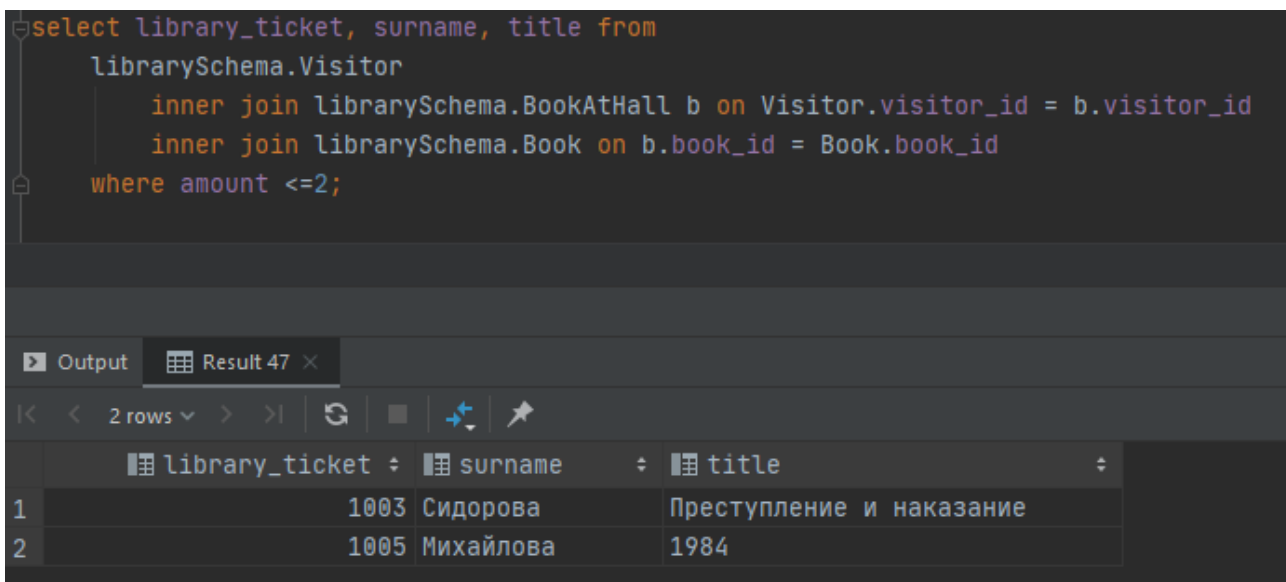


Рисунок 9 – Запрос на вывод: за кем из читателей закреплены книги, количество экземпляров которых в библиотеке не превышает 2.

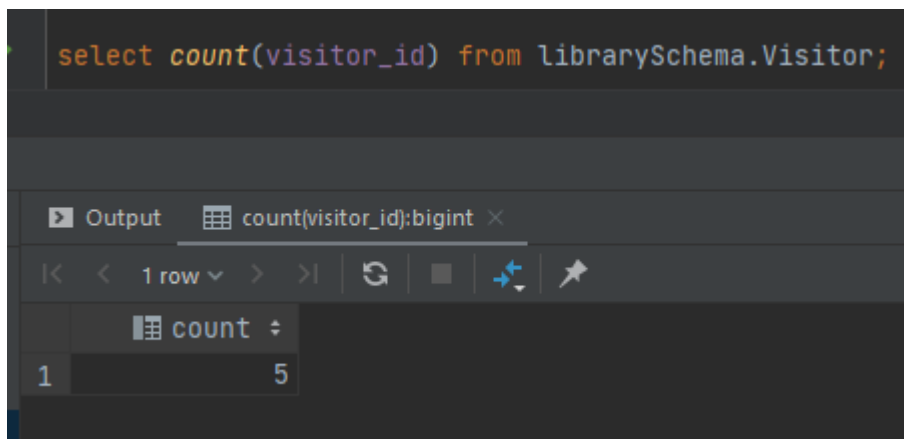


Рисунок 10 – Запрос на вывод: количество читателей, использующих библиотеку

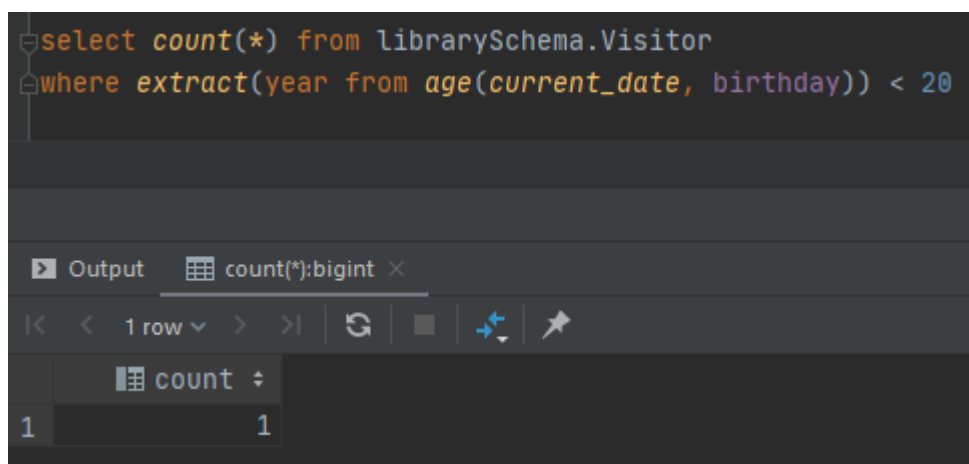


Рисунок 11 – Запрос на вывод: количество читателей младше 20 лет

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/35>

DB Fiddle: <https://www.db-fiddle.com/f/94so9UwkBEfT8wASko4fc/0>

Приложение Б

Исходный код

Файл lab2.sql:

```
create schema if not exists librarySchema;

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

create table librarySchema.Author(
    author_id serial primary key,
    name varchar(30)
);

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

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

```

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

create table librarySchema.Visitor(
    visitor_id serial primary key,
    library_ticket int,
    surname varchar(30),
    passport varchar(11),
    birthday date,
    address varchar(50),
    phone varchar(11),
    education varchar(30),
    academic_degree bool,
    hall_id integer,
                                foreign      key      (hall_id)      references
librarySchema.LibraryHall(hall_id)
);

create table librarySchema.BookAtHall(
    hall_id int,
    book_id int,
                                foreign      key      (hall_id)      references
librarySchema.LibraryHall(hall_id) on delete cascade,
    foreign key (book_id) references librarySchema.Book(book_id)
on delete cascade,
    visitor_id int,
                                foreign      key      (visitor_id)    references
librarySchema.Visitor(visitor_id) on delete set null,
    code int,
    received date,
    returned date

```

```
);
```

```
insert into librarySchema.Publisher(name) values
    ('Азбука-Аттикус'), ('Эксмо'), ('АСТ'), ('Альпина Паблишер'),
    ('МИФ');
```

```
insert into librarySchema.Author(name) values
    ('Толстой Л.Н.'),
    ('Достоевский Ф.М.'),
    ('Оруэлл Д.'),
    ('Остин Д.'),
    ('Хемингуэй Э.');
```

```
insert into librarySchema.Book(publisher_id, title, year, amount)
values
    (1, 'Война и мир', '1869', 10),
    (1, 'Анна Каренина', '1877', 20),
    (2, 'Преступление и наказание', '1866', 2),
    (2, 'Братья Карамазовы', '1880', 5),
    (3, 'Гордость и предубеждение', '1813', 8),
    (3, 'Эмма', '1815', 14),
    (4, '1984', '1949', 1),
    (4, 'Скотный двор', '1945', 9),
    (5, 'Старик и море', '1952', 13),
    (5, 'Прощай, оружие', '1929', 17);
```

```
insert into librarySchema.AuthorBook(author_id, book_id) values
    (1,1), (1,2), (2,3), (2,4), (3,5), (3,6), (4,7), (4,8), (5,9),
    (5,10);
```

```
insert into librarySchema.LibraryHall(name, capacity) values
    ('Красный', 25), ('Синий', 20), ('Зеленый', 25);
```

```

insert into librarySchema.Visitor(library_ticket, surname,
passport, birthday, address, phone, education, academic_degree,
hall_id) values
    (1001, 'Иванов', '1234 678901', '2010-05-15', 'ул. Ленина,
123', '89205678901', 'Высшее', true, 1),
    (1002, 'Петров', '2345 789012', '1985-12-10', 'ул. Пушкина,
45', '89066789012', 'Среднее', false, 2),
    (1003, 'Сидорова', '3456 890123', '1995-03-25', 'пр. Гагарина,
67', '89217890123', 'Высшее', true, 3),
    (1004, 'Козлов', '4567 901234', '1982-08-03', 'ул.
Маяковского, 56', '89208901234', 'Среднее', false, 1),
    (1005, 'Михайлова', '5678 012345', '1998-07-18', 'ул. Кирова,
89', '89219012345', 'Высшее', false, 2);

```

```

insert into librarySchema.BookAtHall(hall_id, book_id, visitor_id,
code, received) values
    (1, 1, 1, 1121, '2023-08-08'),
    (2, 2, 2, 2222, '2023-10-20'),
    (3, 3, 3, 3333, '2023-10-15'),
    (2, 4, 4, 2444, '2023-08-22'),
    (1, 7, 5, 1755, '2023-10-19');

```

```

select title from
    librarySchema.Book
        inner join librarySchema.BookAtHall on
librarySchema.Book.book_id = librarySchema.BookAtHall.book_id
        inner join librarySchema.Visitor on
librarySchema.BookAtHall.visitor_id =
librarySchema.Visitor.visitor_id
    where library_ticket = 1001;

```

```

select title from
    librarySchema.Book
        inner join librarySchema.BookAtHall BAH on
librarySchema.book.book_id = BAH.book_id

```

```
where code = 1755;
```

```
select code from
    librarySchema.BookAtHall
        inner join librarySchema.Book b on b.book_id =
librarySchema.BookAtHall.book_id
    where title = '1984';
```

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

```
select surname, library_ticket, title from
    librarySchema.Visitor
        inner join librarySchema.BookAtHall b on
librarySchema.Visitor.visitor_id = b.visitor_id
        inner join librarySchema.Book b2 on b2.book_id = b.book_id
    where (select current_date - received) > 30;
```

```
select library_ticket, surname, title from
    librarySchema.Visitor
        inner join librarySchema.BookAtHall b on
Visitor.visitor_id = b.visitor_id
        inner join librarySchema.Book on b.book_id = Book.book_id
    where amount <=2;
```

```
select count(visitor_id) from librarySchema.Visitor;
```

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