**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 задания.

**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. Библиотекарю 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*».



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.

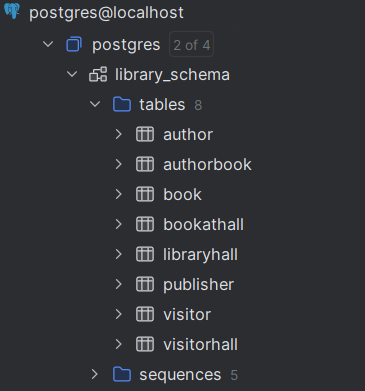


Рисунок 2 – Созданные таблицы.



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

Написаны запросы на вывод нужных сведений для Библиотекаря:

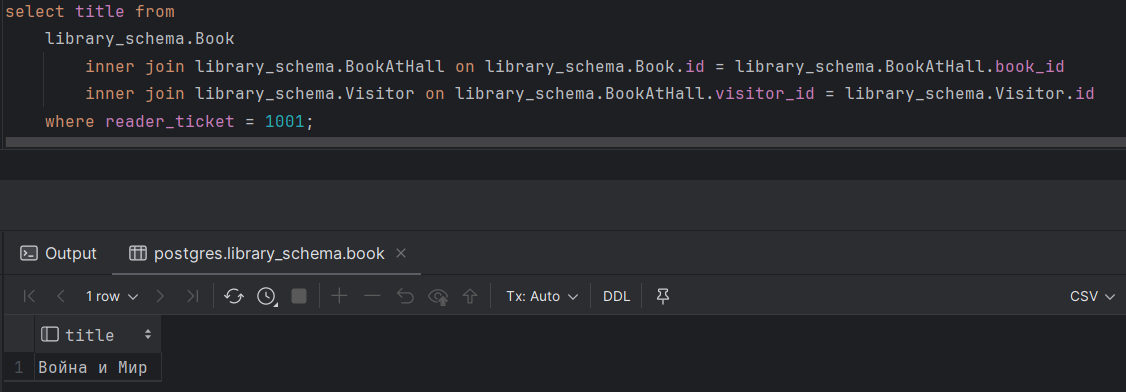


Рисунок 2 – Запрос на вывод: книги, закрепленные за определенным читателем.

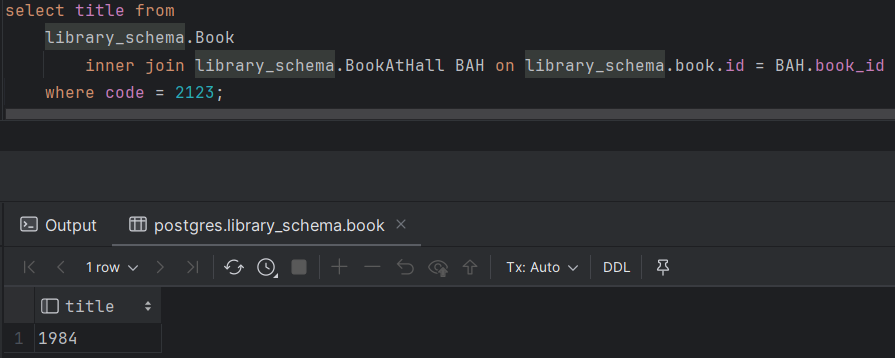


Рисунок 3 – Запрос на вывод: название книги с заданным шифром.

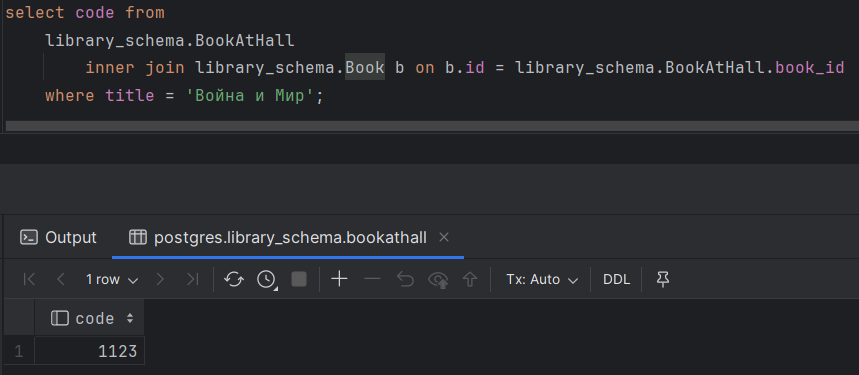


Рисунок 4 – Запрос на вывод: шифр у книги с заданным названием.

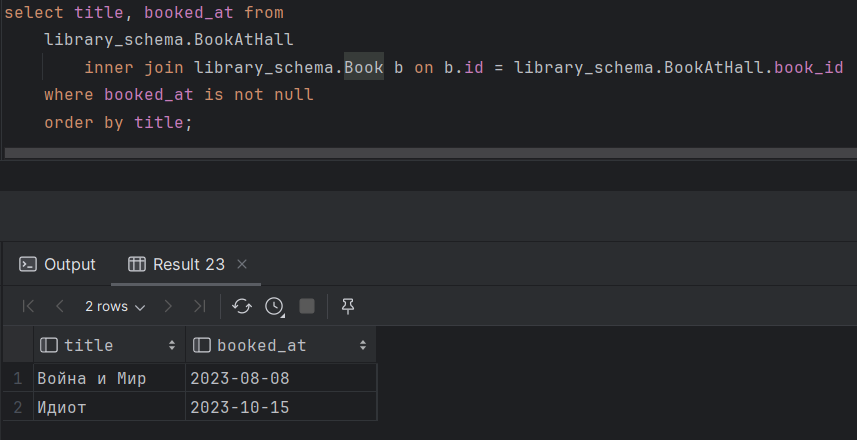


Рисунок 5 – Запрос на вывод: книга была закреплена за читателем.

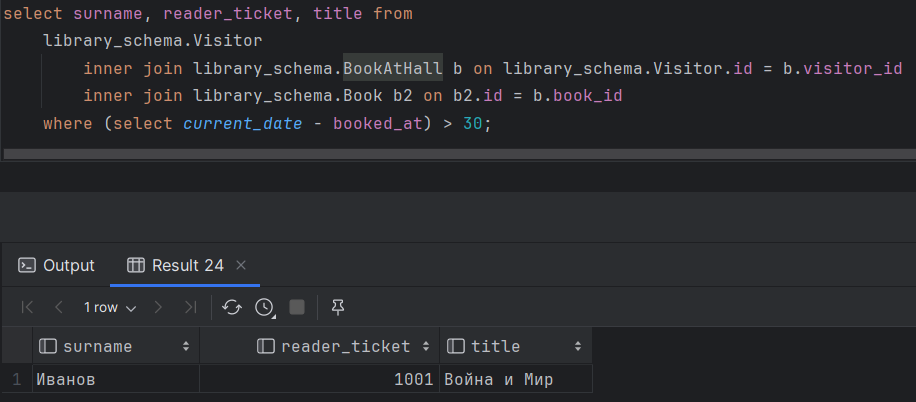


Рисунок 6 – Запрос на вывод: кто из читателей взял книгу более месяца тому назад.

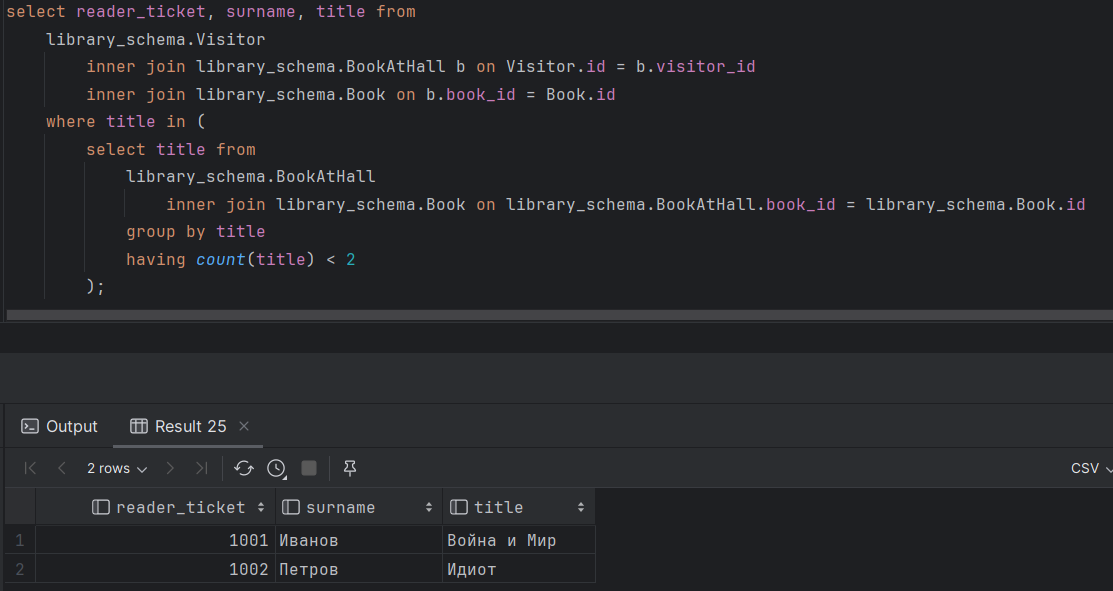


Рисунок 7 – Запрос на вывод: за кем из читателей закреплены книги, количество экземпляров которых в библиотеке не превышает 2.

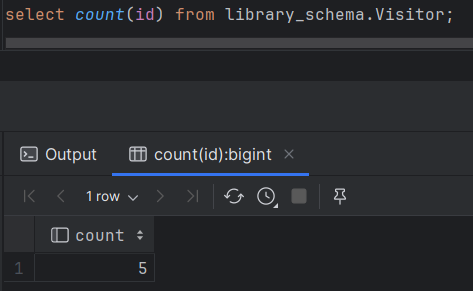


Рисунок 10 – Запрос на вывод: количество читателей, использующих библиотеку

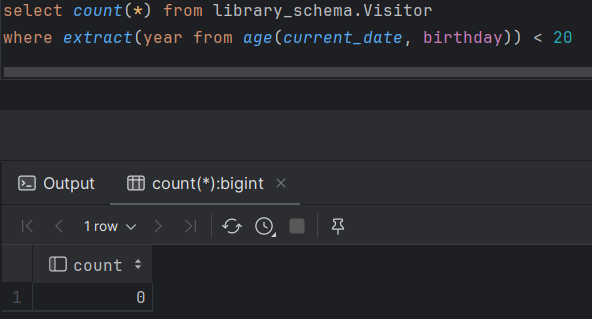


Рисунок 11 – Запрос на вывод: количество читателей младше 20 лет

**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 задания.

**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, 'Aнна 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