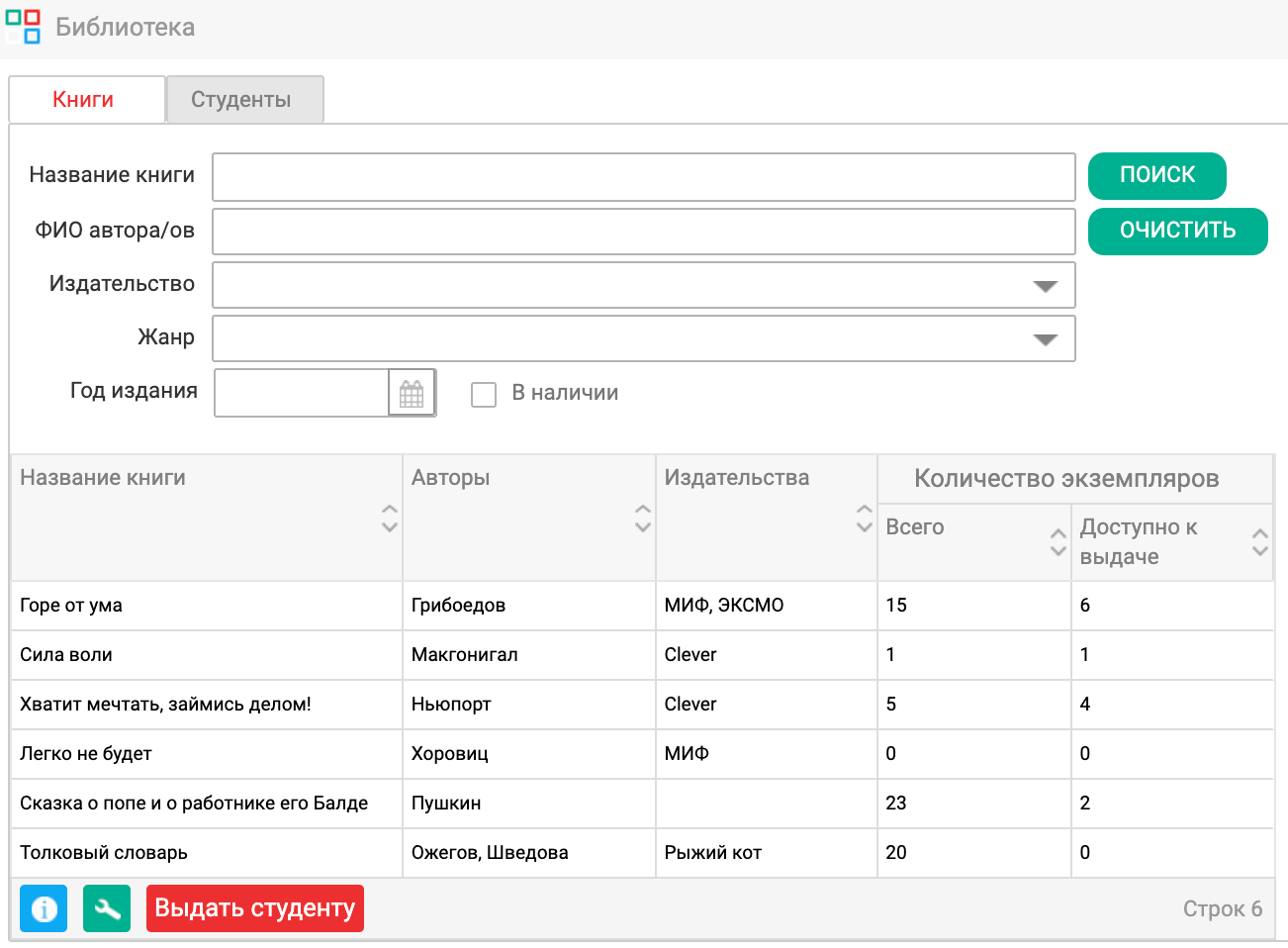
# **Тестовое задание**

# стажер аналитик-проектировщик

# ФИС

## Вводные данные

Перед вами скриншот из приложения для учета книг в библиотеке



Это приложение сделано на базе конструктора веб-приложений Case Platform. Ключевой функционал приложения: хранение названий книг, ФИО авторов, наименований издательств. Учет имеющихся в библиотеке экземпляров конкретной книги. Учет студентов, которым выдавалась конкретная книга. При каждой выдаче книги студенту, фиксация даты выдачи, а при возврате – даты возврата книги.

## Задания

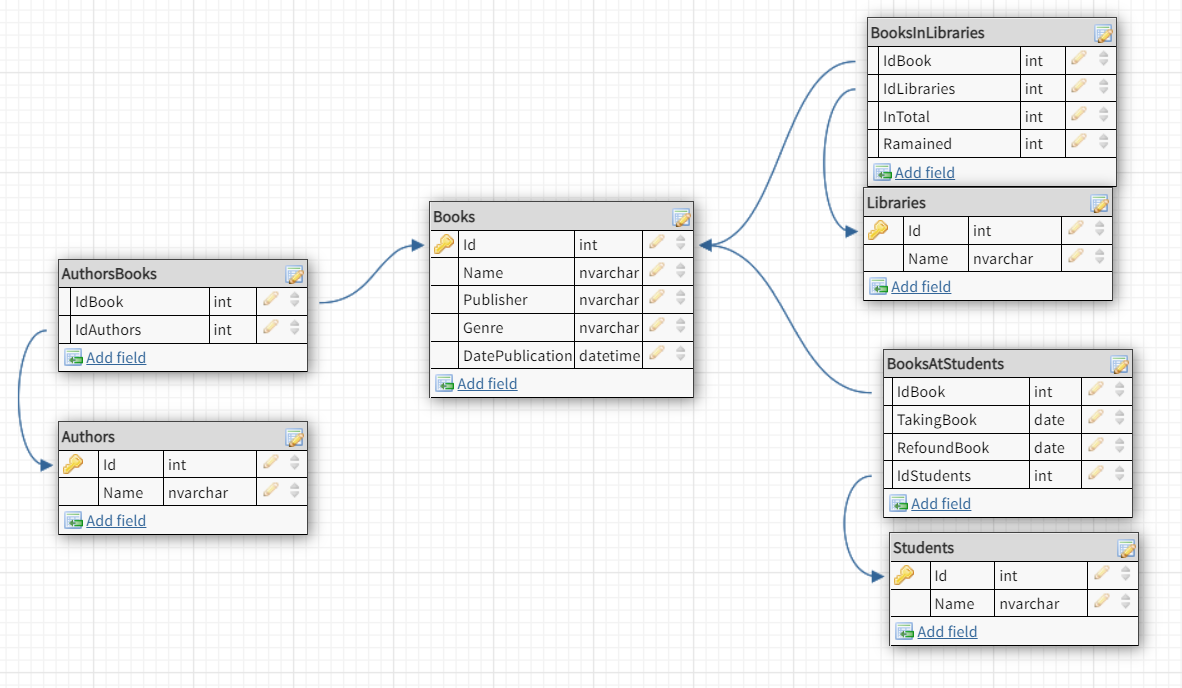
### Модель данных

Опишите модель данных, которая может лежать в основе данного приложения.

Ожидаемый результат: ER диаграмма с пояснениями о выборе сущностей и типов данных их атрибутов.

Результат:

Ссылка на диаграмму : https://dbdesigner.page.link/19MazcdViiFEZLyGA



Пояснение:

Сущность Books:

Создана полностью с картинки которая приложена, только добавлен идентификатор Id

Сущность

Сущность Libraries Students:

добавлена для избежания избыточности данных и реализации связи многие ко многим. Сущность Library можно было бы убрать, но простота ее добавления сейчас и проблемы добавления ее в будущем просто не соизмеримы

Сущности BookAtStudents и BookInLibraries

связь многие ко многим и дополнительные параметры для контроля

### Популярный автор

Напишите SQL-запрос, который бы возвращал самого популярного автора за год.

Ожидаемый результат: SQL-запрос, который основывается на модели данных из задания 1.

Результат:

select top 1 COUNT( a.Name) as CountAuthor, a.Name from BooksAtStudents bs

join AuthorsBooks ab on bs.IdBook=ab.IdBook

join Authors a on ab.IdAuthors=a.Id

where YEAR(bs.TakingBook)=2022

group by a.name

order by countAuthor desc

### Злостный читатель

Определите понятие «злостный читатель». Предложите алгоритм для поиска самого злостного читателя библиотеки, напишите функцию, возвращающую злостного читателя.

Ожидаемый результат: определение понятия “злостный читатель”, словесное описание логики алгоритма, функция поиска такого читателя на любом языке программирования (js будет плюсом). Алгоритм должен основываться на модели данных из задания 1.

Злостный читатель- тот студент который взял больше книг либо прочитал. Если студент прочитал книгу и вернул это ценится больше чем если он ее просто взял и не вернул.

Проходим по списку книг у студентов

смотрим учтен ли студент в временном списке

Если да:

получаем записи всех взятых и взятых-сданных книг у конкретного студента

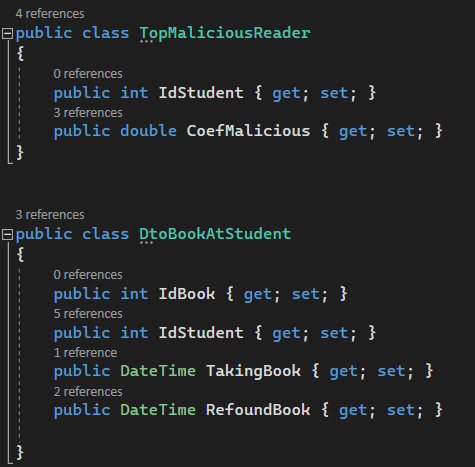
проходим по всем книгам у студента

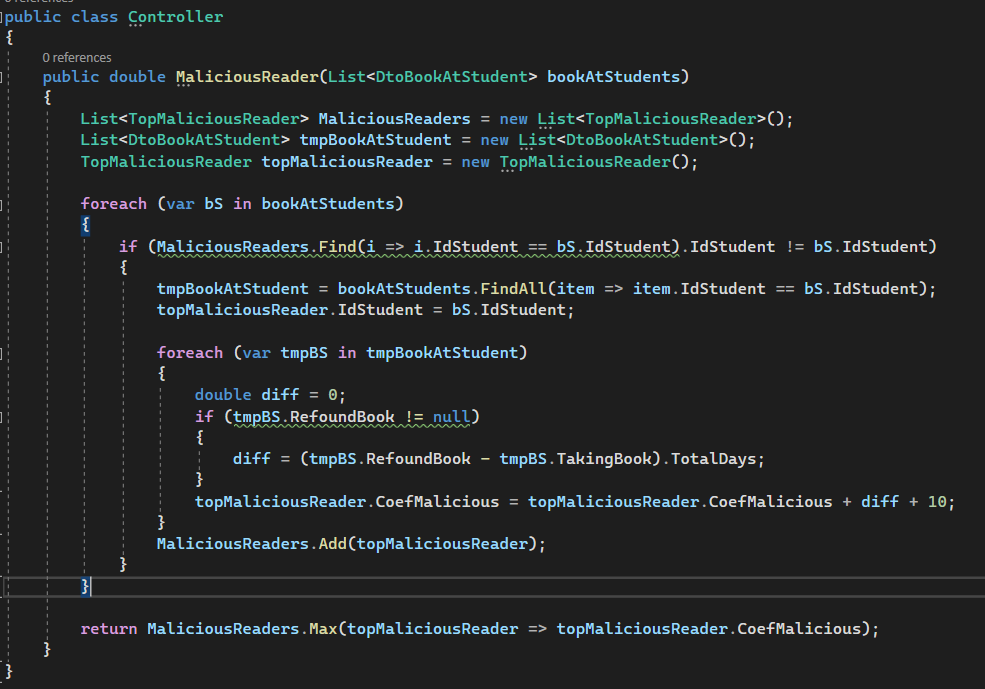
если студент вернул книгу, считаем сколько дней она у него была

вычисляем коэффициент злостного читателя

если нет: идем к следующей строчке книг у студентов

возвращаем Id студента у которого максимальый коэффициент злостного читателя





Необходимо выполнить все задания и сформировать один файл, или предоставить ссылку на ресурс, например, github. При необходимости, можете включить в работу дополнительные скриншоты, картинки и файлы.