**Тестовое отборочное задание (КОРУС Консалтинг, ДАР) для системных аналитиков**

Привет! Любая активность, связанная с развитием, должна чему-то учить. Нам очень понравились две присланные работы, и мы бы хотели показать ответы из них, а также привести частые ошибки, которые нам встречались в других работах. Для задания 3.4 есть много способов написать запрос, советуем также заглянуть в комментарий для разработчиков и посмотреть задание 1 – там приведена еще одна классная работа.

**Задание №1**

Из числа жителей большого города были случайно выбраны 1000 человек. Их спросили, довольны ли они качество воздуха в городе. 78% опрошенных ответили "да". Что можно с уверенностью утверждать, исходя из данных опроса?

1. 78% жителей этого города довольны качеством воздуха в городе.
2. Если случайно выбрать другую тысячу человек из этого же города, 78% из них ответят, что довольны качеством воздуха в городе.
3. Если случайно выбрать 1000 человек из другого города, процент довольных качеством воздуха будет таким же

**Выберите ответ**:

1. Ничего из вышеперечисленного
2. только 1
3. 1 и 2
4. 2 и 3

**Хороший ответ:** a) Ничего из перечисленного с уверенностью нельзя утверждать, для утверждения 3 нет вообще никаких оснований, а для 1 и 2 недостаточно точности опроса, так как 1000 человек для крупного города — это слишком малый показатель, и если выбрать другую 1000 человек, то показатели могут сильно различаться.

**Частые ошибки:** b)

**Задание №2**

Вам даны две базы данных с информацией о студентах и их оценках. Вам нужно выяснить, какие студенты имеют средний балл выше определенного порога и составить рейтинг студентов по этому критерию. Опишите, как вы бы проанализировали эти данные и какие шаги бы вы предприняли для выполнения этой задачи.

**Хороший ответ:**

Для решения данной задачи я бы принял следующие шаги:

1. Изучил бы, в каком виде представлена информация об оценках, для решения, как нужно рассчитать средний балл.
2. Создал бы третью таблицу с двумя колонками – студент и его средний балл
3. Выбрал бы студентов из таблицы, у которых средний балл выше «определенного порога» и отсортировал список студентов по среднему баллу

Предположим, что таблица Студенты состоит из атрибутов id студента, Фамилия, Имя, Отчество и Информация о студенте; а таблица Оценки состоит из 3 атрибутов, id студента, id предмета (который находится в третьей таблице) и Оценка.

Создадим таблицу Рейтинг

CREATE TABLE Рейтинг (

id INT PRIMARY KEY,

Фамилия VARCHAR(255),

Имя VARCHAR(255),

Отчество VARCHAR(255),

Средний\_балл DECIMAL(5, 2)

);

Внесем в нее необходимые данные

INSERT INTO Рейтинг (id, Фамилия, Имя, Отчество, Средний\_балл)

SELECT

Студенты.id,

Студенты.Фамилия,

Студенты.Имя,

Студенты.Отчество,

AVG(Оценки.Оценка) AS Средний\_балл

FROM

Студенты

JOIN Оценки ON Студенты.id = Оценки.id\_студента

GROUP BY

Студенты.id,

Студенты.Фамилия,

Студенты.Имя,

Студенты.Отчество;

Пускай «Определенный порог» будет равен 4, тогда данный код выведет необходимую нам информацию

SELECT id, Фамилия, Имя, Отчество, Средний\_балл

FROM Рейтинг

WHERE Средний\_балл > 4

ORDER BY Средний\_балл DESC;

**Частые ошибки:**

Скудное описание рассуждений. Встречались ответы, которые как будто состояли из самого задания (синим – текст из задания, красным – дописанный ответ)

~~Вам даны~~ *Соединю* две базы данных с информацией о студентах и их оценках. ~~Вам нужно выяснить, какие студенты имеют~~ *Посчитаю* средний балл выше определенного порога и ~~составить рейтинг студентов по этому критерию.~~ *отсортирую по получившемуся баллу.*

**Задание №3 (кейс)**

У вас есть модель фрагмента базы данных библиотеки и семплы данных, соответствующие данной модели.

|  |  |  |
| --- | --- | --- |
| **Книги** | **Выдачи книг** | **Выдачи книг** |
| * ID книги (int) * Название (text) * Автор (text) * Издательство (text) * Год издания (int) * Город издания (text) * Количество страниц (int) * ID экземпляра (int) * Дата поступления в библиотеку | * ID экземпляра (int) * Дата выдачи (date) * Дата возврата (date) * № читательского билета (int) | * № читательского билета (int) * Фамилия (text) * Имя (text) * Отчество (text) * Дата рождения (date) * Пол (text) * Адрес (text) * Телефон (text) |

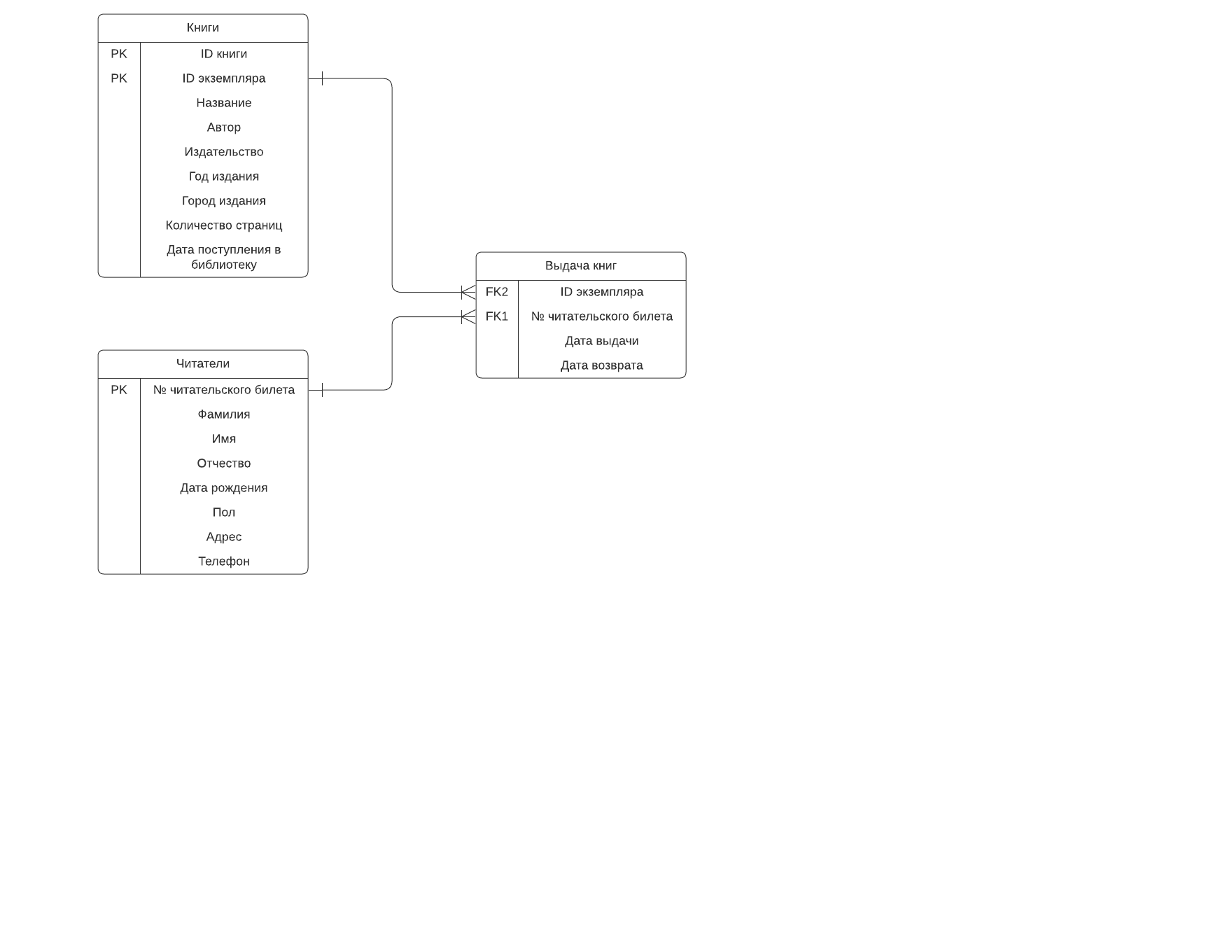
**Задание** **3.1**

Нарисуйте ER-диаграмму модели данных. Определите соответствует ли модель данных 3 нормальной форме.

**Хороший ответ:**

В задании не были приведены ключи, поэтому при построении диаграммы я определил их самостоятельно. Для таблицы «Книги» был определен составнонй ключ: «ID книги, ID экземпляра». Однако из полученных семплов данных невозможно точно определить ключ ввиду недостатка информации. Предполагается что ключом может быть только ID экземпляра (подробнее в пункте 3.3). Для таблицы «Читатели» ключом является «№ читательского билета». Для «Выдача книг» не удалось определить первичный ключ (подробнее в пункте 3.3). Внешним ключом в данной таблице являются «ID экземпляра» и «№ читательского билета».

ER-диаграмма:



Определение третьей нормальной формы гласит: таблица должна быть во второй нормальной форме и каждый не ключевой атрибут нетранзитивно зависит от первичного ключа. При этом вторая нормальная форма говорит, что соответствует первой нормальной форме и каждый неключевой атрибут неприводимо зависит от Первичного Ключа. В свою очередь первая нормальная форма - когда все его атрибуты являются простыми, все используемые домены должны содержать только скалярные значения и не должно быть повторений строк в таблице.

Рассмотрим наш пример. Таблица «Выдача книг» содержит данные «ID экземляра», «№ читательского билета», «Дата выдачи» и «Дата возврата». В предложенных семплах данных «ID экземляра» не является ункальным полем, что приведет к дублированию строк в данной таблице.

Так же рассмотрим таблицу «Книги»: в моем понимании поле «ID экземляра», должно быть уникальным и быть функционально связано с «ID книги», но при этом «ID книги» не уникально и может повторяться, так как библиотека может иметь несколько одинаковых книг. Следовательно можно разделить эту таблицу на две:

|  |  |  |
| --- | --- | --- |
| «ID экземляра» | «ID книги» | Дата поступления в библиотеку |
|  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| «ID книги» | Название | Автор | Издательство | Год издания | Город издания | Кол-во страниц |
|  |  |  |  |  |  |  |

Кроме того, у одной книги может быть несколько Авторов, это так же может быть нарушением первой нормальной формы и для корректности хранения данных вынесено в отдельную таблицу «ID книги» - «Автор».

Таким образом я делаю вывод, что данная диаграмма не соответствует третьей нормальной форме.

**Частые ошибки:**

1. ER-диаграмма – это буквально схема «сущность-связь». Если на вашей схеме не указаны связи, это не ER-диаграмма.
2. На ER-диаграмме не указаны в явном виде ключи (например, просто названия выделены жирным шрифтом)
3. Некорректно определенные ключи в ER-диаграмме. Больше всего пострадала таблица «Выдача книг». Чаще всего ключом становился только «id читательского билета» либо «id экземпляра», но по данному ключу невозможно точно идентифицировать запись сущности «Выдача книг». В данной задаче могут быть разные варианты решения: оно зависит от правил выдачи книг, которые приняты в данной библиотеке, а также от изначальной идеи ведения данной таблицы.

* Простой вариант: составной первичный ключ из двух внешних ключей - «id читательского билета» + «id экземпляра».
* Ключ состоит сразу из трех полей: «id читательского билета» + «id экземпляра» + «дата выдачи», если предположить, что один читатель может брать один и тот же экземпляр несколько раз
* Но лучше всего создать новый атрибут «id выдачи», т.к. один читатель может брать один и тот же экземпляр несколько раз в один и тот же день: например, на перемене между парами студент берет книгу, которую нельзя выносить из читального зала.

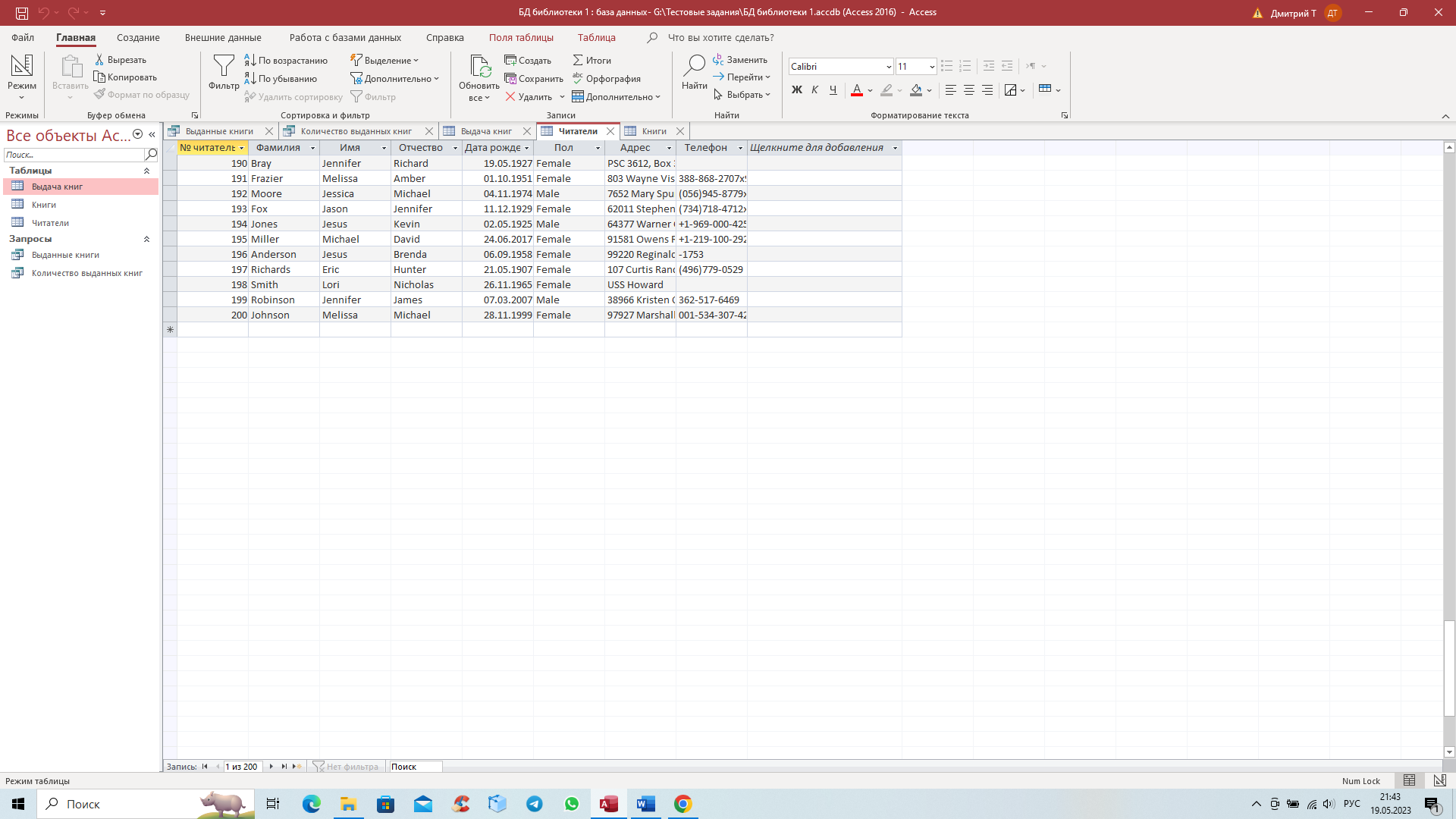
**Задание 3.2**

Создайте базу данных и заведите таблицы\*. Загрузите предоставленные семплы данных. (Приложите скрин, каждой загруженной таблицы).

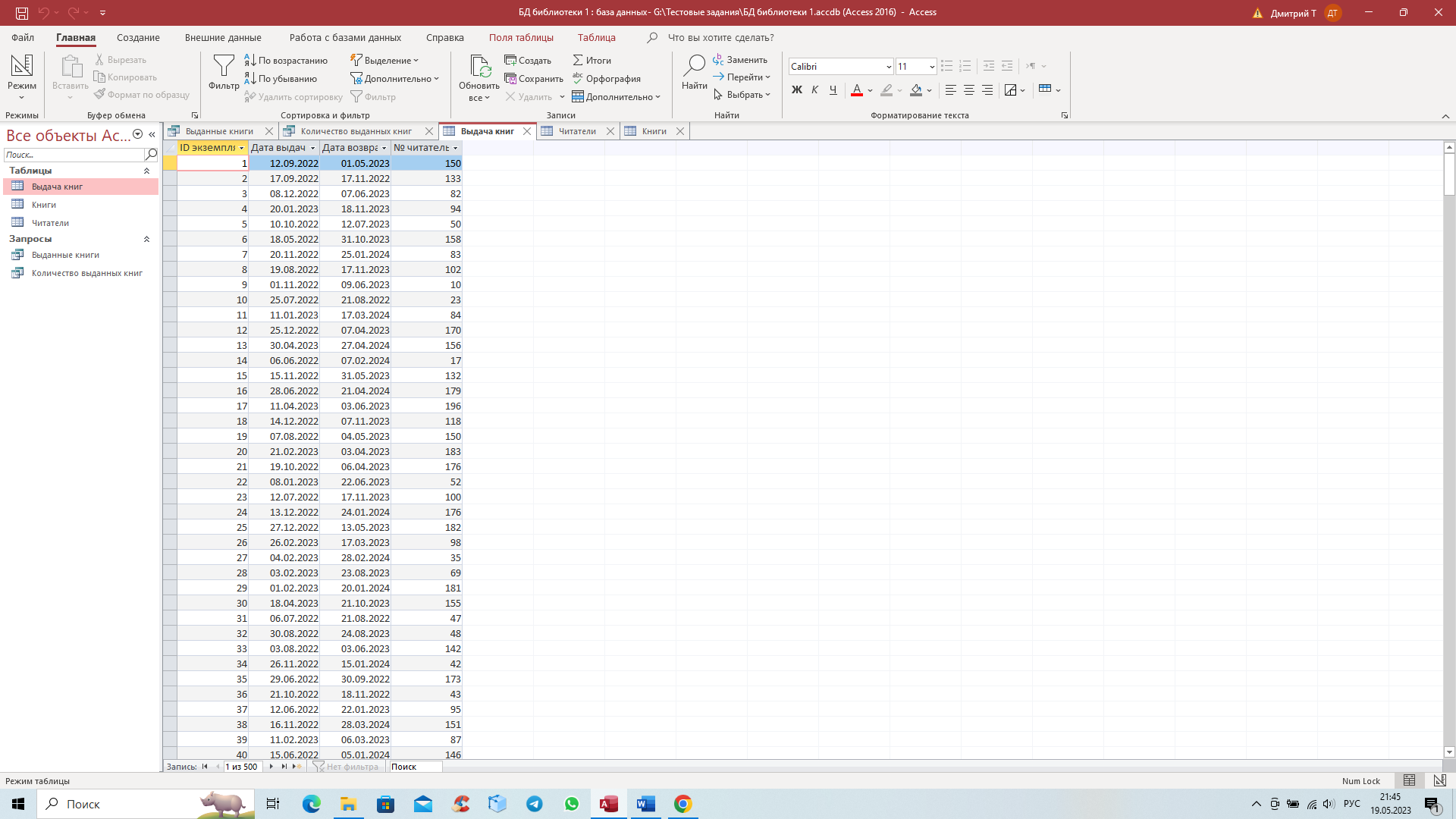
\* Базу данных можно создать используя любое удобное клиентское приложение. (Например: DBvear, MS access, Pl/sql developer).

**Хороший ответ:**

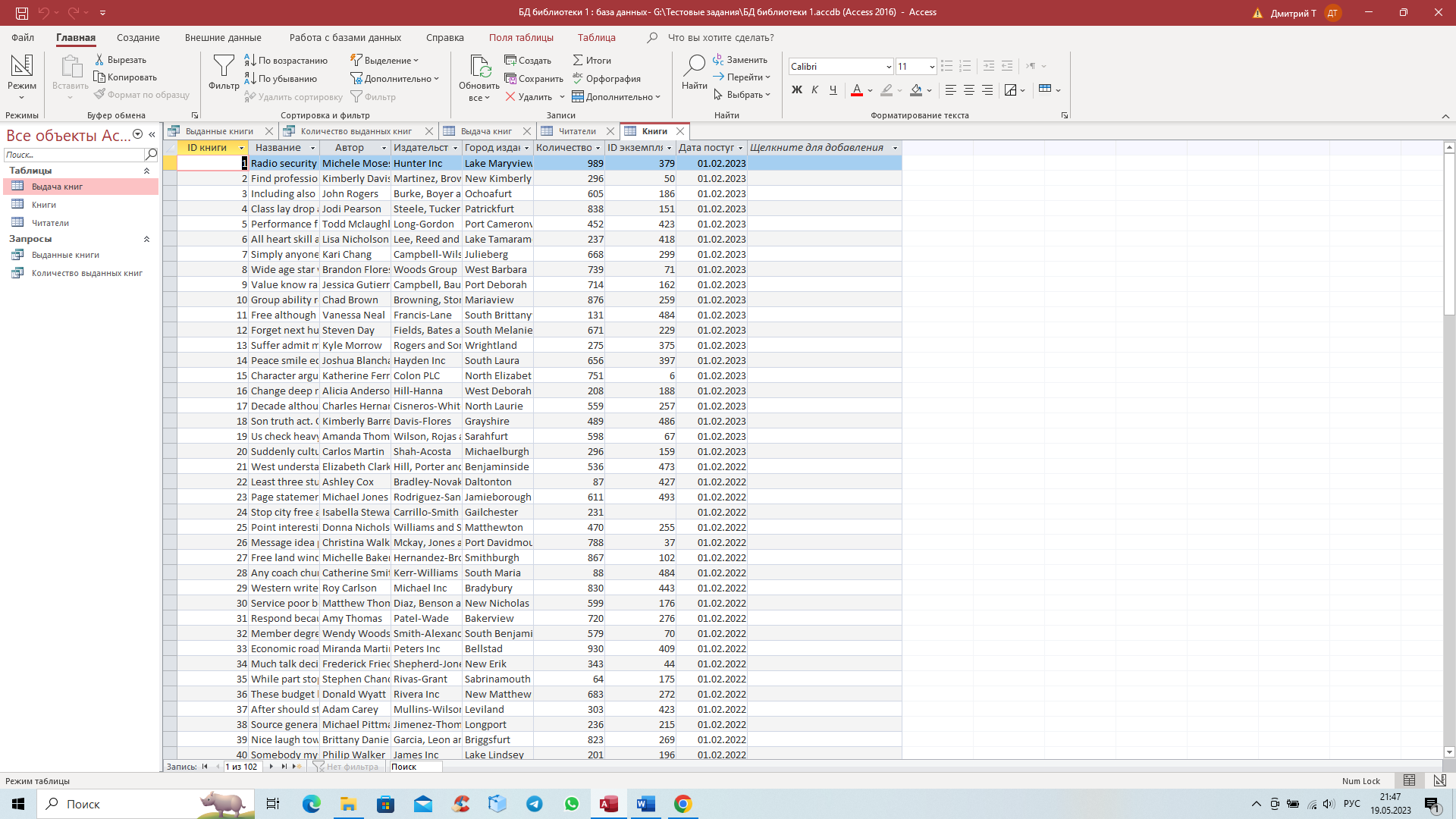
Читатели:



Выдачи книг:



Книги:



**Частые ошибки:** Остановиться перед трудностями: «Не удалось загрузить из-за ошибок в данных»

**Задание** **3.3**

Оцените качество/корректность предоставленных сэмплов данных\*\*. Зафиксируйте письменно результаты, приведите примеры по каждой из найденных проблем.

\*\* В качестве основных проблем, вызывающих снижение качества данных, обычно выделяют следующие:

* пропущенные значения;
* дубликаты;
* противоречия;
* шум;
* некорректные форматы и представления данных;
* фиктивные значения;
* нарушения структуры.

Задание под звёздочкой (не обязательное, но можно получить дополнительные баллы):

Напишите sql - запросы с помощью которых мы можем выявить ошибки в данных (количество запросов не ограничено, необходимо приложить текст запроса/запросов).

**Хороший ответ:**

Одной из ключевых проблем является то, что из за недостатка информации, в таблицах невозможно определить первичный ключ. Так, например в таблице «Книги» непонятно, что является ключем. В случае если «ID экземпляра» определяется для каждой конкретной книги с уникальным «ID книги», то составным ключем являются поля «ID книги» и «ID экземпляра». Отсюда вытекает проблема для идентификации данных в таблице «Выдача книг». Во втором случае, если «ID экземпляра» является уникальным для каждой книги в библиотеке, тогда оно же и является ключем.

Далее вытекает следующая ошибка, в случае если «ID экземпляра» является первичным ключем, то в данных, предоставленных в семпле, присутствуют ошибки с повторяющимися значениями, проверить это можно с помощью следующего запроса:

SELECT \*

FROM Книги

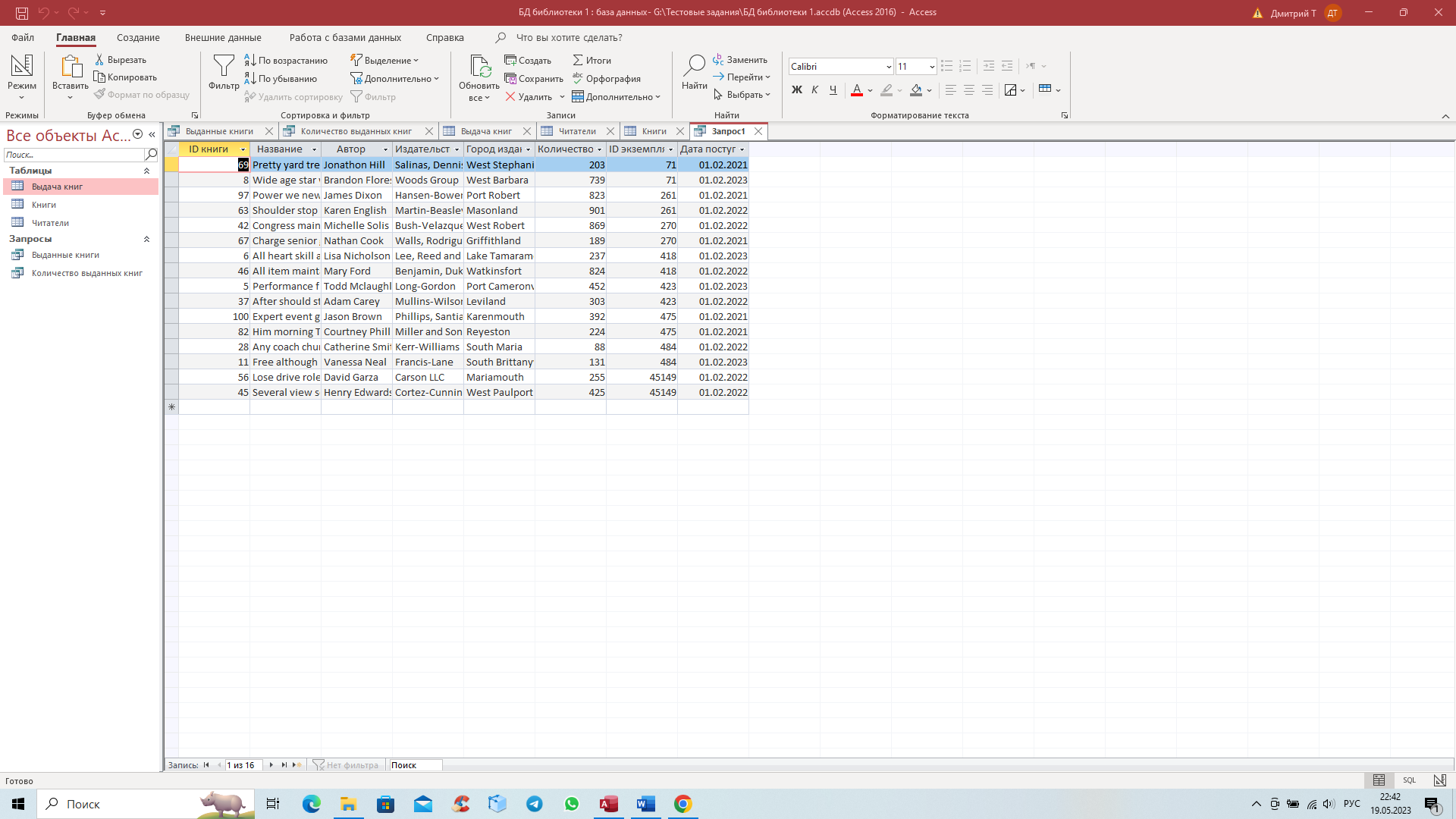
WHERE (((Книги.[ID экземпляра]) In (SELECT [ID экземпляра] FROM (SELECT [ID экземпляра], COUNT(\*)

FROM Книги

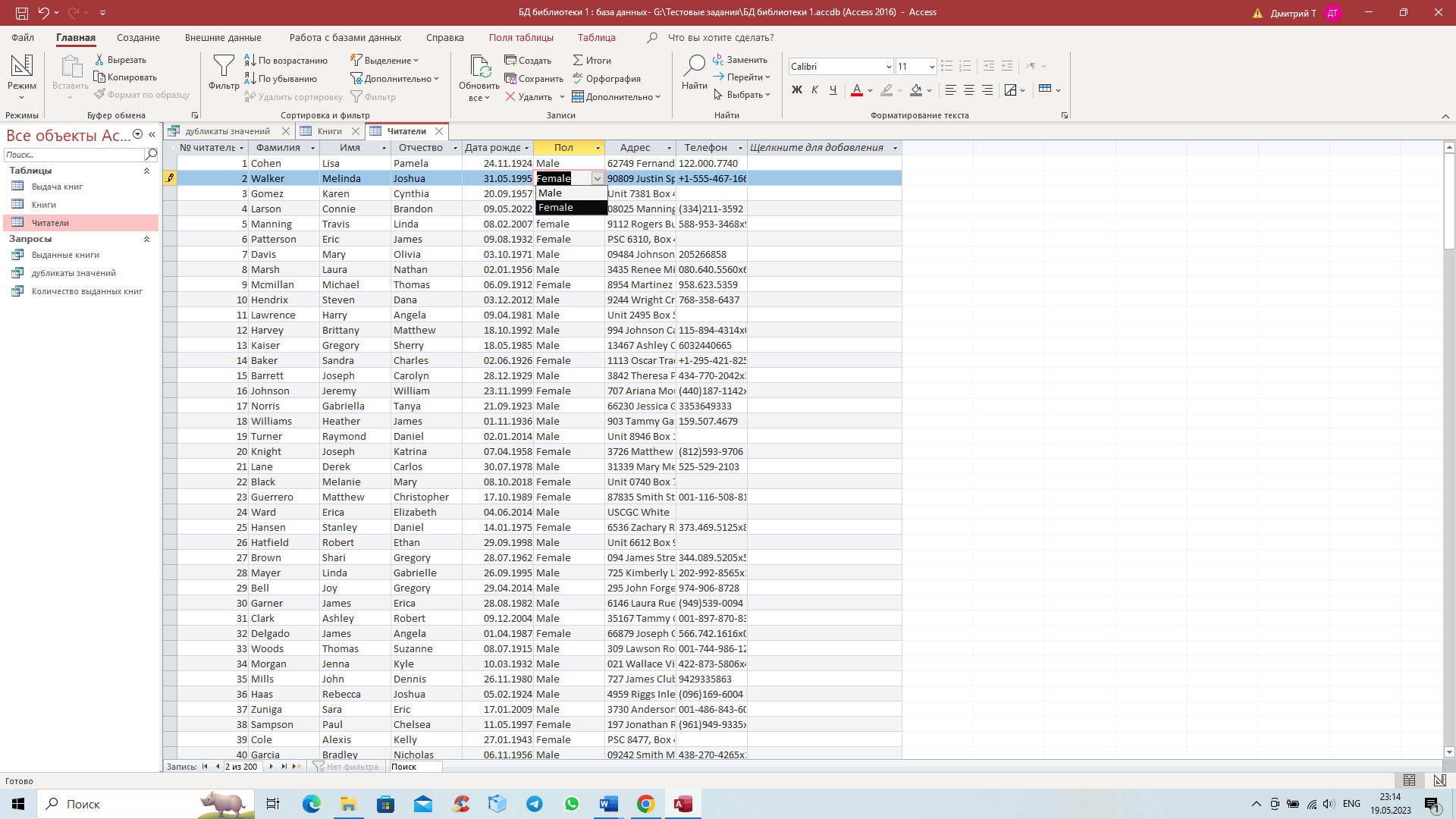
GROUP BY [ID экземпляра]

HAVING COUNT(\*) > 1))))

ORDER BY [ID экземпляра];



Так же для большего удобства в таблицу читатели тип данных можно указать список из которого могут быть выбраны варианты «Male» или «Female» с целью избежать лишних ошибок в правописании:



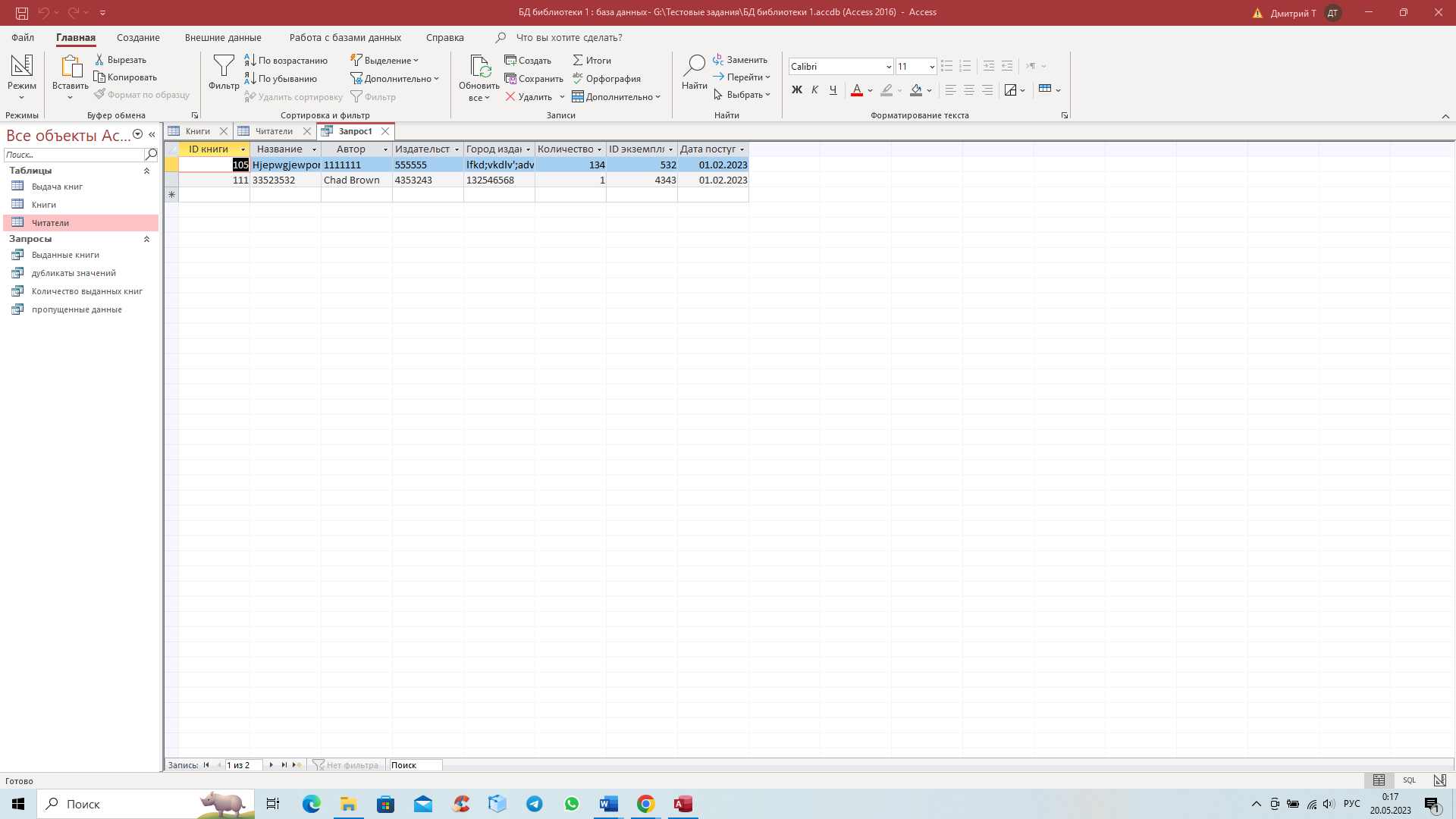
Другим улучшением будет вынос списка издательств, авторов и городов в отдельные таблицы, так при заведении данных можно допустить грамматические ошибки или опечатки и лучше ссылаться на отдельные таблицы. Такой подход позволит уменьшить количество ошибок в наименованиях.

Далее ошибкой является, что в некоторых таблицах встречаются неправильные типы данных, например в названии книги, или имени автора стоят цифр. Проверить это можно с помощью запроса:

SELECT Книги.\*

FROM Книги

WHERE ((Название Like "\*[0-9]\*")) OR ((Автор Like "\*[0-9]\*")) OR ((Издательство Like "\*[0-9]\*")) OR (([Город издания] Like "\*[0-9]\*"))



Также подобным образом можно выделить строки с недопустимыми символами:

SELECT \*

FROM Книги

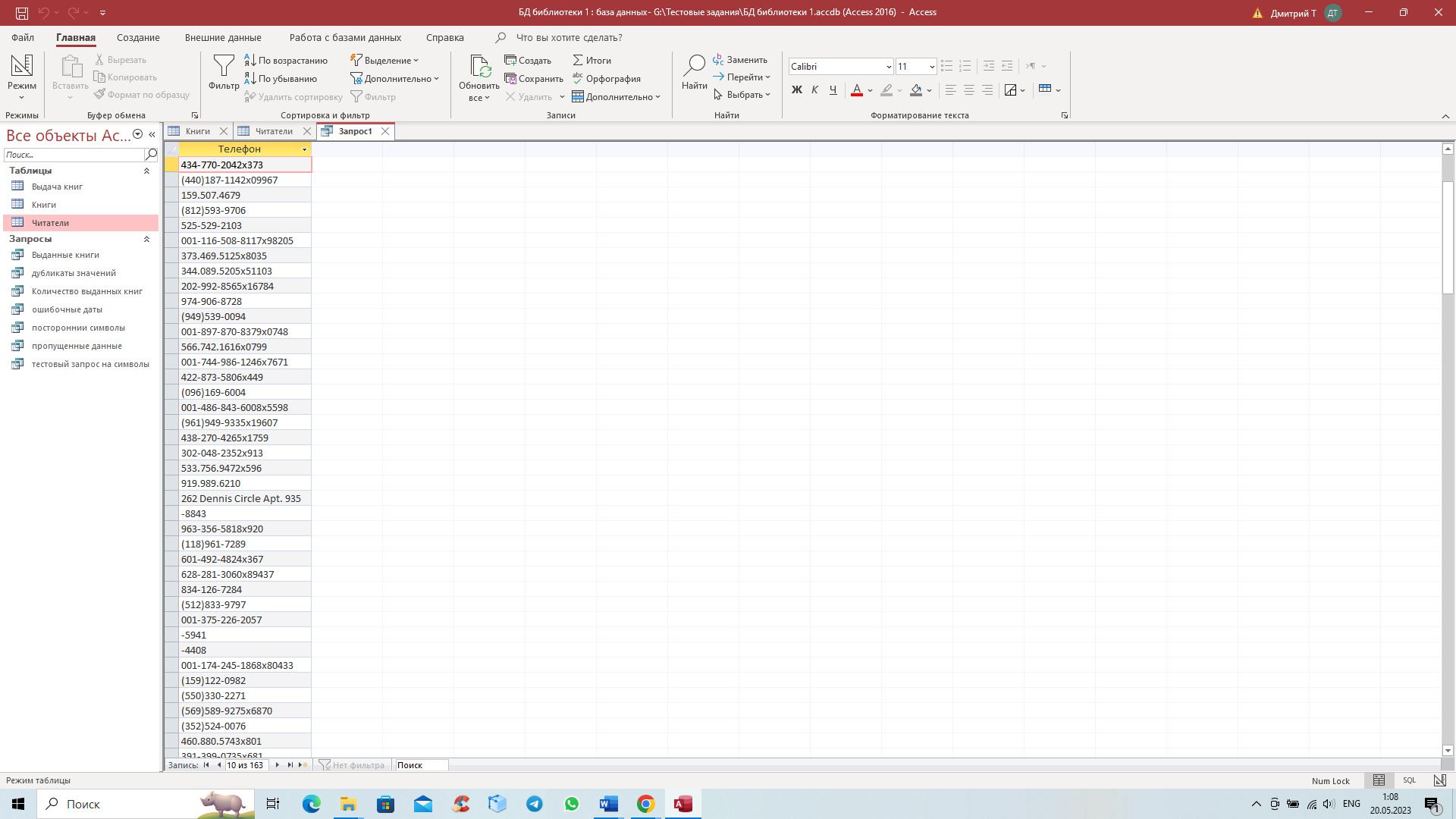
WHERE [Город издания] Like "\*[.||,||'||""||\||/||!||@|#||$||%||^||&||\*||(||)||?||>||<||;||:||+||-||=]\*";

В качестве ошибки так же можно выделить отсутствие единого шаблона для номера телефона читателей в следствии чего в данной графе появляются некорректные данные.

SELECT Телефон

FROM Читатели

WHERE ((Телефон Like "\*[a-Z||-||,|| ||.]\*") or (Телефон Like "\*-\*"));

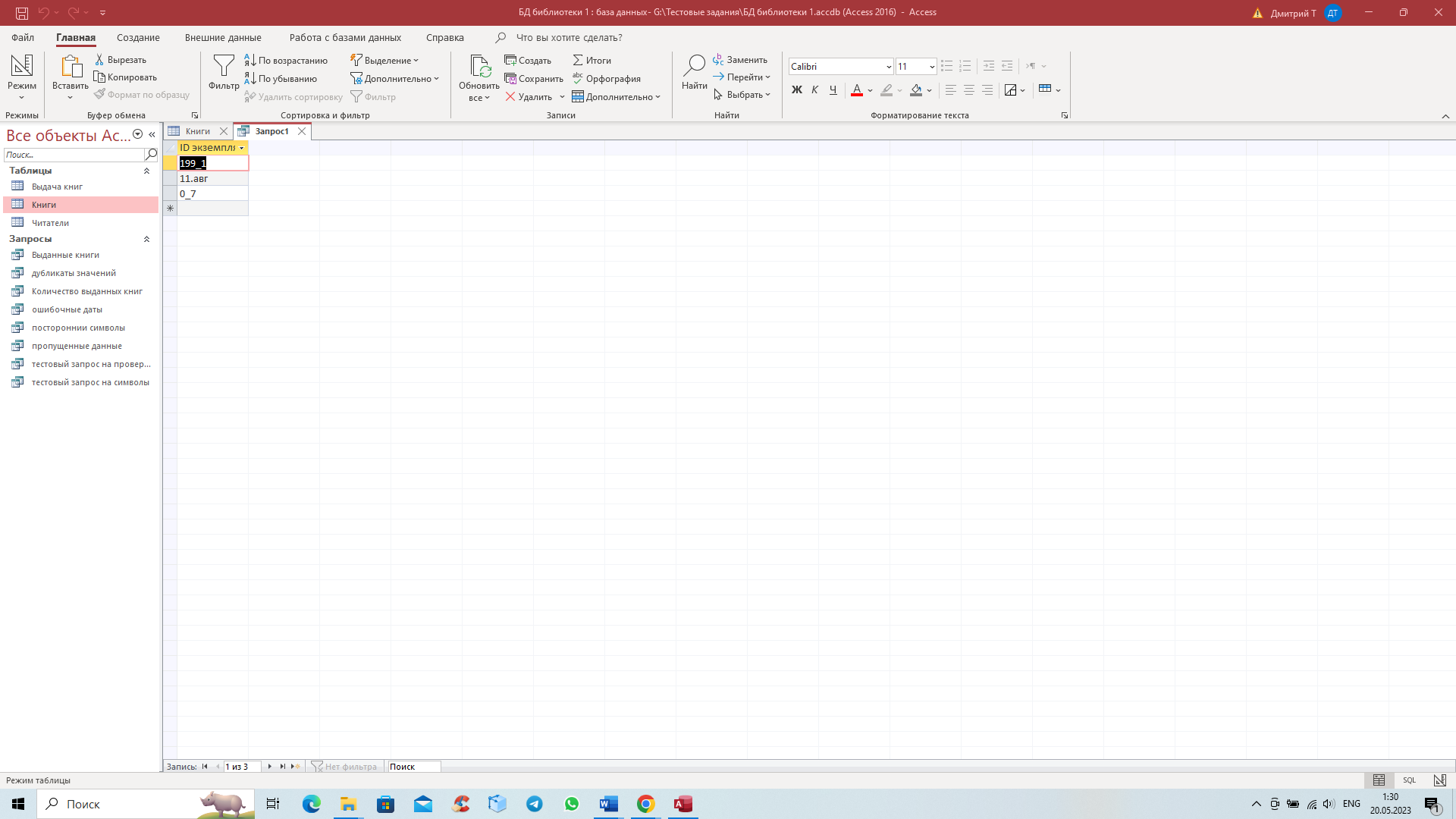


Также в таблице «Книги» в колонке «ID экземпляров» не указан фильтр вводимых данных, что приводит к образованию ошибочных наименований.

SELECT [ID экземпляра]

FROM Книги

WHERE ([ID экземпляра] Like "\*[а-Я||a-Z||\_||]\*");



В таблице «Выдачи книг» данные дублируются, это видно при помощи запроса

**SELECT** \*

**FROM** Выдачи\_книг

**WHERE** (Дата\_выдачи, Дата\_возврата, Номер\_читательского\_билета) **IN** (

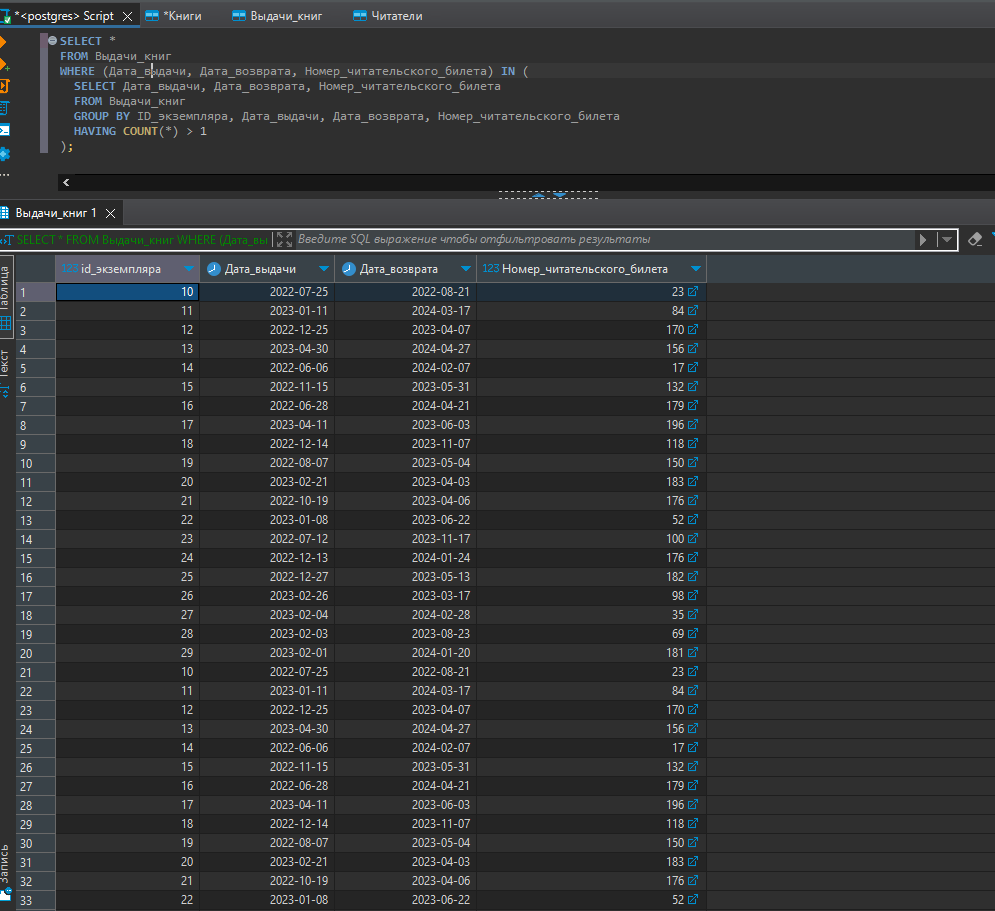
**SELECT** Дата\_выдачи, Дата\_возврата, Номер\_читательского\_билета

**FROM** Выдачи\_книг

**GROUP** **BY** ID\_экземпляра, Дата\_выдачи, Дата\_возврата, Номер\_читательского\_билета

**HAVING** **COUNT**(\*) > 1

);



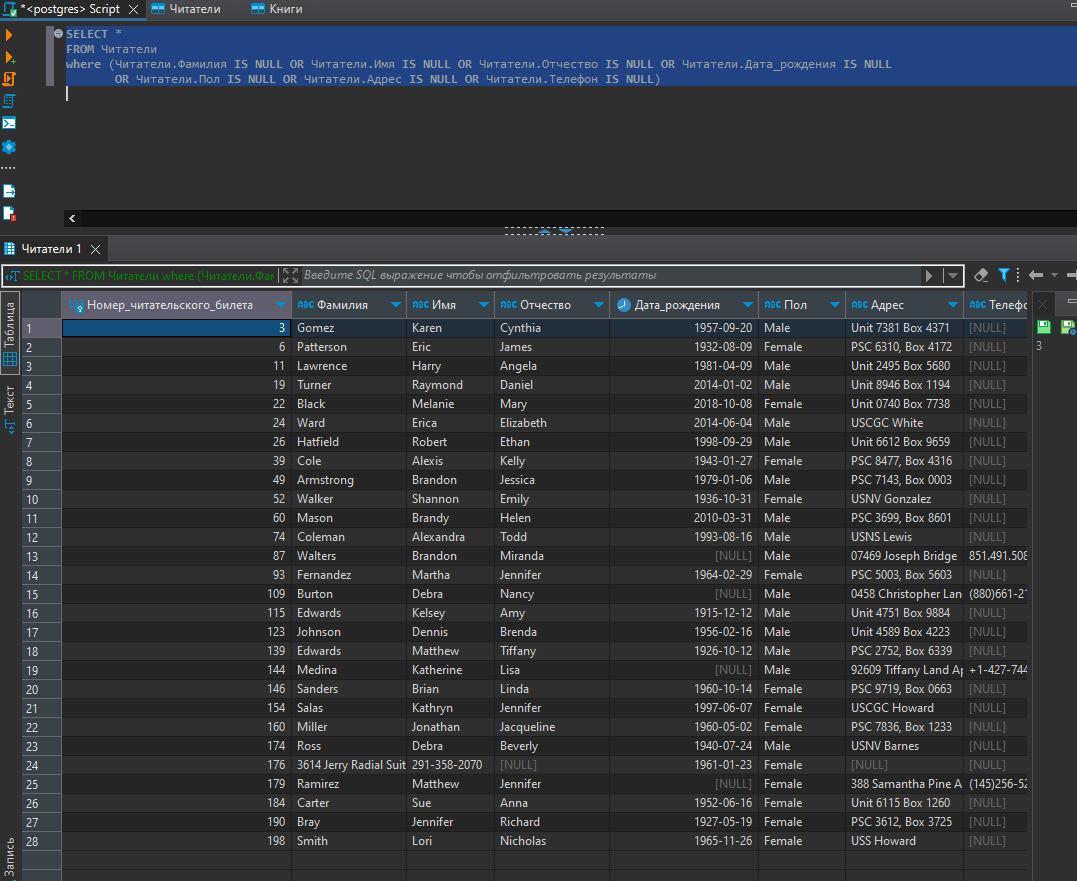
В таблице Читатели есть множество пропущенных значений, их можно увидеть при помощи кода

**SELECT** \*

**FROM** Читатели

**where** (Читатели.Фамилия **IS** **NULL** **OR** Читатели.Имя **IS** **NULL** **OR** Читатели.Отчество **IS** **NULL** **OR** Читатели.Дата\_рождения **IS** **NULL**

**OR** Читатели.Пол **IS** **NULL** **OR** Читатели.Адрес **IS** **NULL** **OR** Читатели.Телефон **IS** **NULL**)



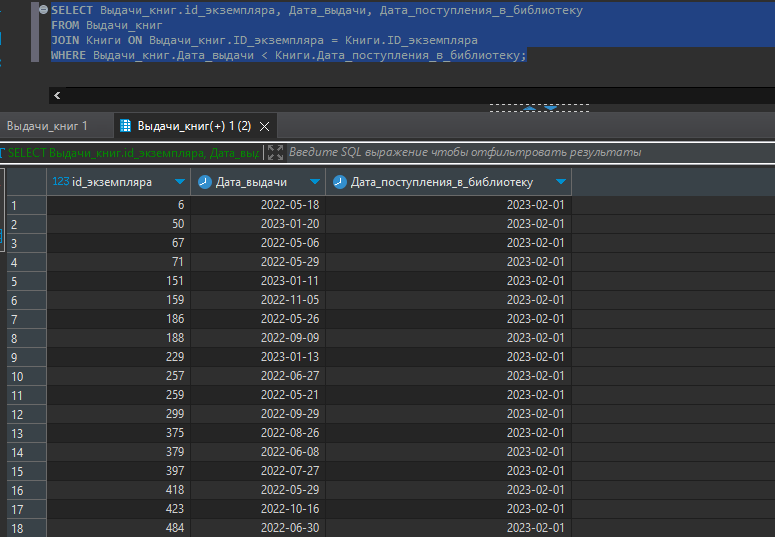
В таблицах присутствуют ситуации, когда дата выдачи раньше даты поступления в библиотеку.

**SELECT** Выдачи\_книг.id\_экземпляра, Дата\_выдачи, Дата\_поступления\_в\_библиотеку

**FROM** Выдачи\_книг

**JOIN** Книги **ON** Выдачи\_книг.ID\_экземпляра = Книги.ID\_экземпляра

**WHERE** Выдачи\_книг.Дата\_выдачи < Книги.Дата\_поступления\_в\_библиотеку;

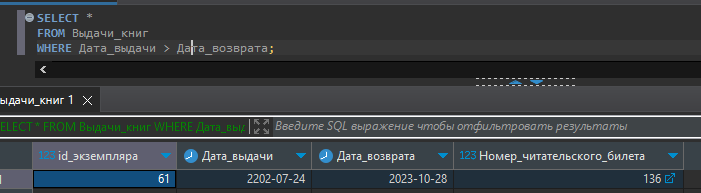


И когда дата выдачи раньше даты возврата

**SELECT** \*

**FROM** Выдачи\_книг

**WHERE** Дата\_выдачи > Дата\_возврата;



**Частые ошибки:**

1. Отсутствие примеров-доказательств проделанной работы. Встречались ответы, которые как будто состояли из самого задания (синим – текст из задания, красным – дописанный ответ)
2. *В предоставленных сэмплах были найдены* пропущенные значения; дубликаты; противоречия; шум; некорректные форматы и представления данных; фиктивные значения; нарушения структуры.
3. Проверка вручную, не функциями и не sql. Как только объем данных станет больше, план будет провален.
4. Поиск ошибок, которых нет ("город издания в таблице может повторяться, значит, это будут дубли")

**Задание** **3.4**

Напишите sql-запросы (приложите текст запроса).

**3.4.1.** Найти города, в которых в 2016 году было издано больше всего книг;

**Хороший ответ:**

SELECT "Город издания", COUNT(\*) as "Количество книг"

FROM «Книги»

WHERE EXTRACT(YEAR FROM "Дата поступления в библиотеку") - 5 = 2016

GROUP BY "Город издания"

ORDER BY 2 DESC;

**3.4.2.** Вывести количество экземпляров книг «Война и мир» Л.Н.Толстого, которые находятся в библиотеке;

**Хороший ответ:**

**SELECT** **COUNT**(\*) **AS** Количество\_экземпляров

**FROM** Книги

**JOIN** Выдачи\_книг **ON** Книги.ID\_экземпляра = Выдачи\_книг.ID\_экземпляра

**WHERE** Книги.Название = 'Война и мир' **AND** Книги.Автор = 'Л.Н.Толстой' **AND** Выдачи\_книг.Дата\_выдачи < '17-05-2023' **AND** Выдачи\_книг.Дата\_возврата < '17-05-2023';

Код выводит количество книг Л.Н. Толстого «Война и мир», которые были возвращены, небыли взяты снова (т.е. находятся сейчас в библиотеке).

**3.4.3.** Найти читателей, которые за последний месяц брали больше всего книг в библиотеке. При выводе выполнить сортировку читателей по возрасту (от молодых к старшим).

**Хороший ответ:**

**SELECT** Читатели.Номер\_читательского\_билета, Читатели.Фамилия, Читатели.Имя, Читатели.Отчество, Читатели.Дата\_рождения, **COUNT**(\*) **AS** Количество\_книг

**FROM** Читатели

**JOIN** Выдачи\_книг **ON** Читатели.Номер\_читательского\_билета = Выдачи\_книг.Номер\_читательского\_билета

**WHERE** Выдачи\_книг.Дата\_выдачи >= '17-04-2023'

**GROUP** **BY** Читатели.Номер\_читательского\_билета, Читатели.Фамилия, Читатели.Имя, Читатели.Отчество, Читатели.Дата\_рождения

**HAVING** **COUNT**(\*) = (

**SELECT** **MAX**(Максимальное\_Количество\_Книг)

**FROM** (

**SELECT** **COUNT**(\*) **AS** Максимальное\_Количество\_Книг

**FROM** Выдачи\_книг

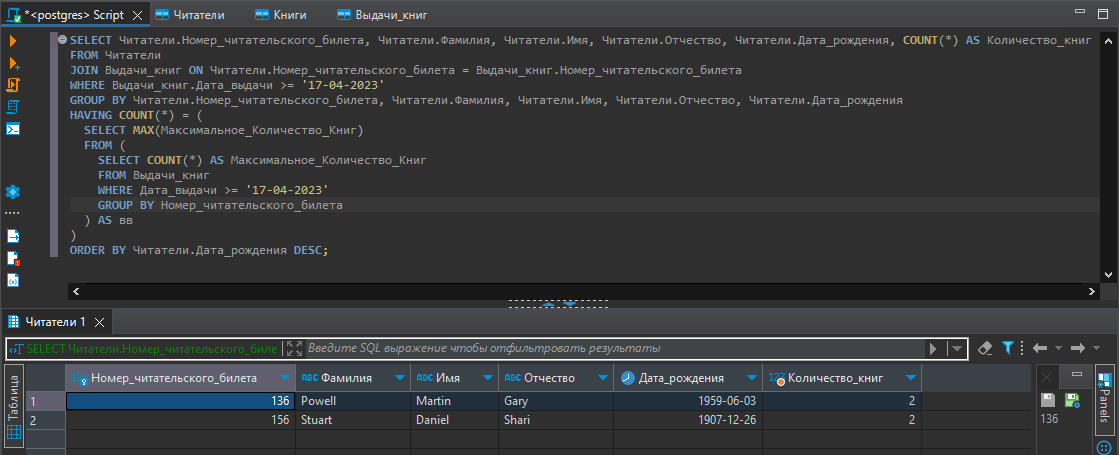
**WHERE** Дата\_выдачи >= '17-04-2023'

**GROUP** **BY** Номер\_читательского\_билета

) **AS** вв

)

**ORDER** **BY** Читатели.Дата\_рождения **DESC**;



**Частые ошибки:**

1. Ограничение мышления по сэмплам. Сэмпл — это вырезка, а не вся таблица целиком. Другие значения в таблице могут отличаться от сэмпла. При решении задачи нужно немного абстрагироваться и подумать, какие еще значения могут принимать атрибуты. При поиске книги «Война и мир» очень часто отвечали, что такой книги нет. При поиске книг за 2016 год – что поле «год издания» не заполнено.
2. Недочитанное задание. Когда в задании было указано «найти наибольшее значение», многие останавливались на подсчете количества. В задании про читателей просто считали количество книг и сортировали по дате рождения, не обращая внимания на то, сколько книг получилось и что надо взять наибольшее значение.
3. Некорректный запрос, который выдаст ошибку при запуске (например, неполный список полей в group by, опечатки и т.д.)
4. Для 3.4.2. не учтены книги, которые никогда не брали (запрос сделан только из таблицы «Выдачи книг») или, наоборот, не учтены выданные книги (запрос только из таблицы «Книги»).
5. Ограничение «наибольших значений» цифрой на свой вкус – топ 10/топ100 и т.д. (limit/top). Не сработает в том случае, если у 11 строк будет одинаковое значение. Если захотелось отсечь 10 результатов, лучше в таком случае использовать ранжирующую функцию, хотя данные задачки решались и без нее.
6. Невнимательность при выполнении задания: выборка из столбца, которого нет, подсчет суммы вместо количества, субъективная интерпретация столбцов («Поле "ID экземпляра" решил понимать, как "Количество экземпляров данной книги в библиотеке"»).

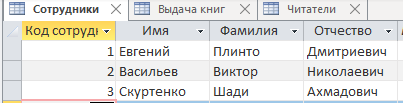
**Задание** **3.5**

Доработать логическую модель данных (ЛМД) таким образом, чтобы можно было хранить данные о сотрудниках библиотеки, которые выдали книгу читателю. Нарисовать ER - диаграмму получившейся ЛМД.

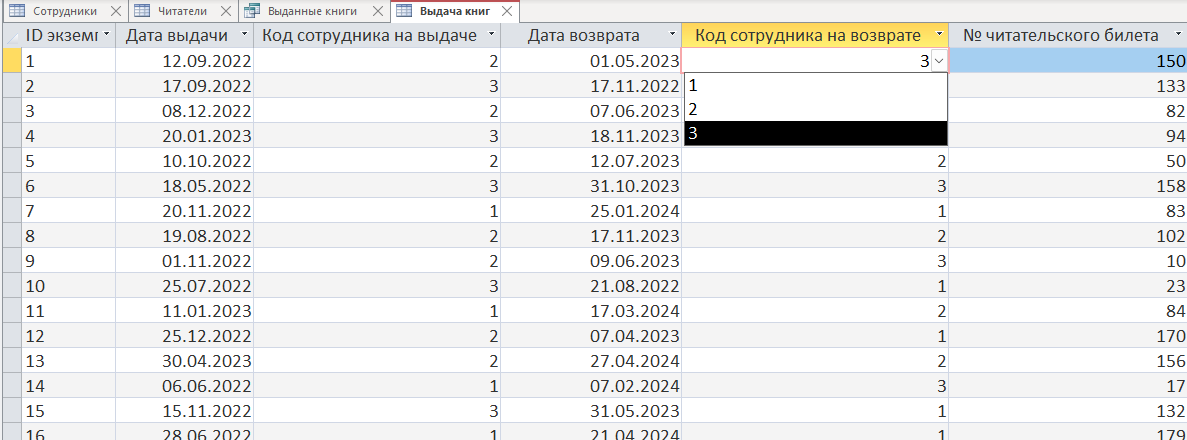
**Хороший ответ:**

Для того что бы выполнить данное задание, необходимо добавить в базу данных еще одну таблицу: «Сотрудники». Также в таблицу «Выдача книг» нужно добавить данные о сотруднике который осуществлял выдачу книги и возврат книги.

Добавленная таблица «Сотрудники»:

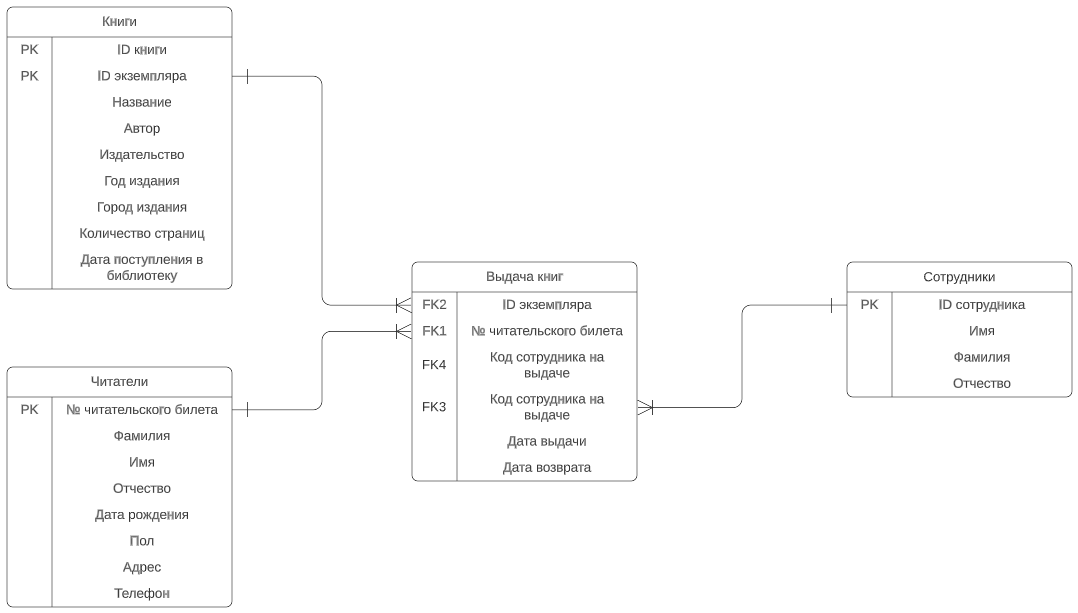


Измененная таблица «Выдача книг»:



В измененной таблице «Выдача книг» код сотрудника, который осуществлял выдачу книги, и который принимал ее обратно, производится выбором из списка, значения для которого берутся из таблицы «Сотрудники». При додавблении новых данных в таблицу «Сотрудники», обновленные ID появятся так же и в предложенных вариантах выбора.

Так же я разработал ER-диаграмму представленной модели данных:



**Общие советы**

Хорошей практикой будет являться:

1. Дублирование постановки задачи перед решением, красивое и читаемое оформление кода и работы в целом. Это облегчает проверку и показывает ваше отношение к заданию.
2. При отрисовке блок-схемы (коей является ER-диаграмма) применение какого-либо устоявшегося стандарта (нотации).
3. Подробное описание логики рассуждений, даже для sql-запроса.
4. Обдумывание и решение разных вариантов (например: непонятно как в библиотеке отмечается дата возврата по книгам, которые еще не вернули: null или поле заполняется, как «плановая дата возврата». Можно проработать оба варианта.
5. Если вам что-то непонятно и кажется неправильным, то лучше поразмышлять и описать это в своих рассуждениях, а не игнорировать. Например, для таблицы «Книги» непонятно, почему в таблице есть «id книги» и «id экземпляра». Какие выводы можно сделать из этого? Все книги хранятся в 1 экземпляре или в таблице есть повторяющиеся книги с разными id экземпляра?
6. Внимание к деталям. Судя по сэмплам таблицы «Книги» и «Читатели», это может быть иностранная библиотека, поэтому стоит поискать «Войну и мир» и автора также на английском языке.