ГУАП

КАФЕДРА № 42

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | Т.В. Семененко |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1 |
| СОЗДАНИЕ БАЗЫ ДАННЫХ И ОБЪЕКТОВ ТИПА “ТАБЛИЦА” В СРЕДЕ MS ACCESS. ОПРЕДЕЛЕНИЕ СВЯЗЕЙ МЕЖДУ ТАБЛИЦАМИ, ЗАДАНИЕ СХЕМЫ БАЗЫ ДАННЫХ |
| по курсу: управление данными |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4329 |  |  |  | Д.С. Шаповалова |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2025

Содержание

[1. Цель работы: 3](#_Toc190831132)

[2. Задание: 3](#_Toc190831133)

[3. Описание предметной области 3](#_Toc190831134)

[4. Обоснование принятых решений по таблицам 4](#_Toc190831135)

[5. Схема базы данных: 9](#_Toc190831136)

[6. Вывод 10](#_Toc190831137)

# 1. Цель работы:

Изучение процесса создания базы данных и объектов типа “Таблица” с помощью Конструктора и с помощью Мастера, освоение методов задания свойств полей таблицы и манипуляций с ними, заполнение таблицы данными и редактирование; изучение способов экспорта/импорта данных из других БД, из БД иной архитектуры, из текстовых файлов; освоение способов определения связей между таблицами.

# 2. Задание:

Создать базу данных в MS Access (определить, сколько и какие таблицы должна содержать БД). Индивидуальные задания находятся в разделе «Варианты заданий для лабораторных работ». При создании таблиц использовать метод построения таблиц с помощью режимов Таблица и Конструктор. При создании таблиц учесть все ограничения, которые заданы в постановке задачи. Предусмотреть необходимые связи между таблицами, определить все поля данных (атрибуты или столбцы) таблицы. Проверить работу всех введенных ограничений, для чего ввести данные во все созданные таблицы в количестве 2 – 3х строк. Задать схему БД. Установить требуемые связи и создать требуемую схему БД. Применить необходимые методы поддержки целостности при описании связей.

# 3. Описание предметной области

Вы разрабатываете базу данных для домашней библиотеки компьютерных дисков и книг. В вашей библиотеке присутствуют как компьютерные диски различных жанров типов и форматов (CD и DVD, mp3 и mp4, диски с программами, энциклопедии и т. д.), так и книги (учебная и художественная литература). У вас есть знакомые, у которых вы можете взять какие-нибудь диски или книги взаймы, также вы сами можете дать им в долг часть своей библиотеки. Разрабатываемая информационная система должна отслеживать перемещение вашей собственности, а также отслеживать ваши долги. Хранится информация не только о названии диска или книги, но и более подробная информация (например, если это фильм, то в каком формате, на каком диске, какой жанр фильма, длительность фильма, какие известные актеры там снимались…).

# 4. Обоснование принятых решений по таблицам

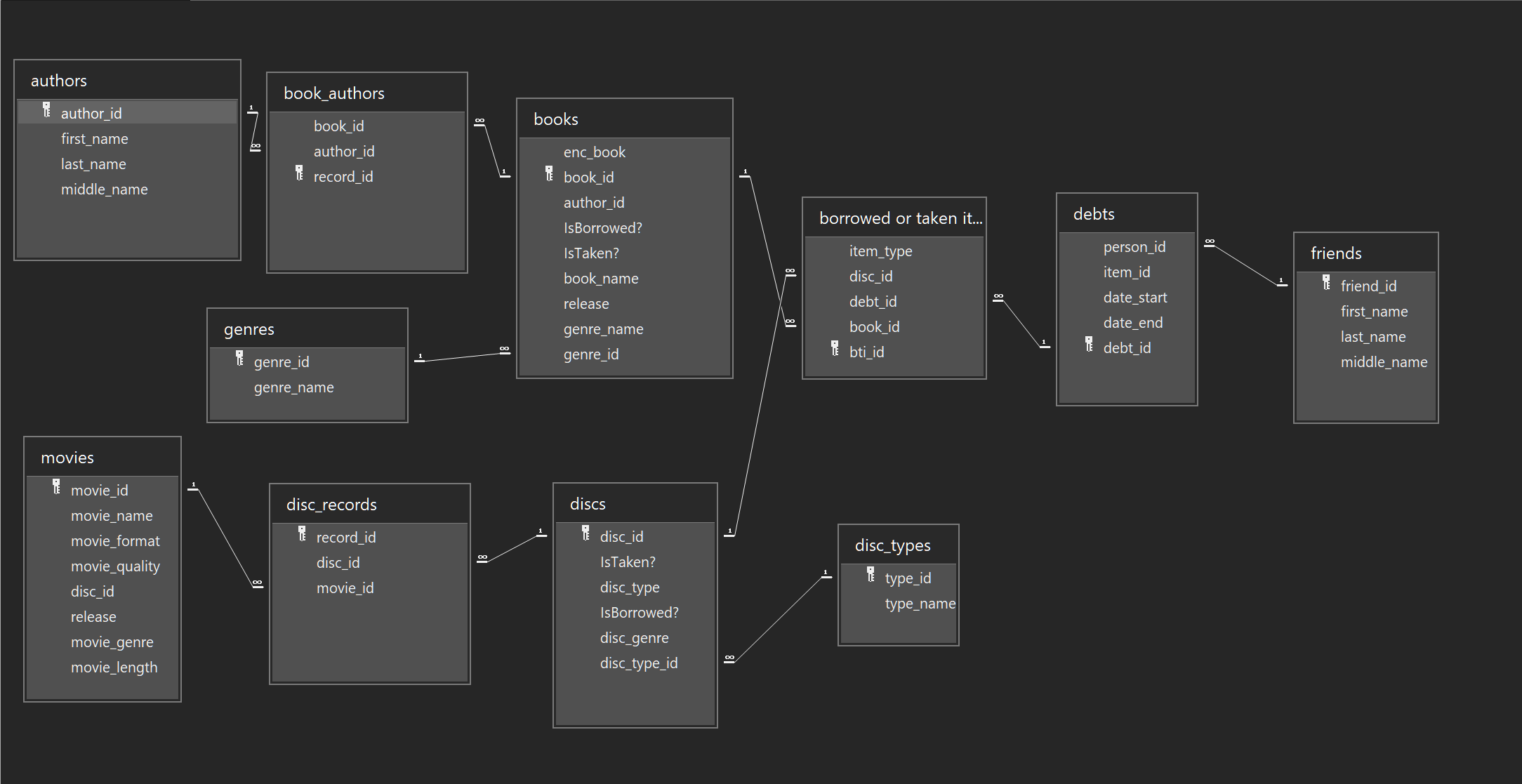


Рисунок 1 – Схема результирующей БД.

Для работы с данными о домашней библиотеке книг и дисков была разработана БД. Для удобного представления данных они были расставлены по таблицам с характеризующими эти данными названиями и полями.

Помимо этого, были выстроены связи между таблицами по ключевым полям(как правило id тех или иных элементов).

Далее будут представлены примеры настроек таблиц из БД, которые помогают либо унифицировать способ представления данных(все даты в таблицах вводятся при помощи одинаковых маск ввода), либо упрощают восприятие данных для пользователя БД(item\_type в borrowed or taken items или IsTaken?/IsBorrowed? в discs и books).

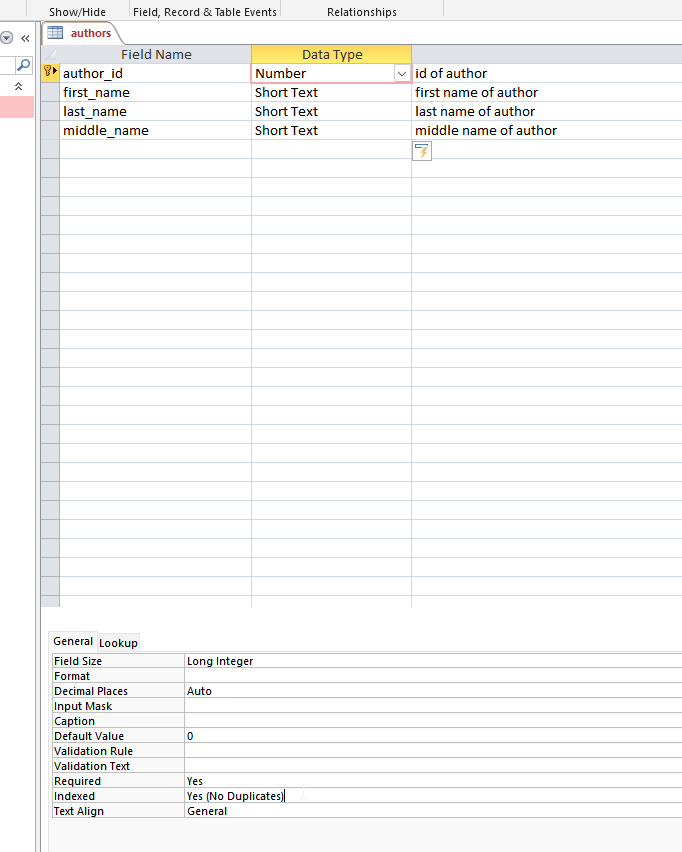


Рисунок 2 – Конструктор таблицы authors.

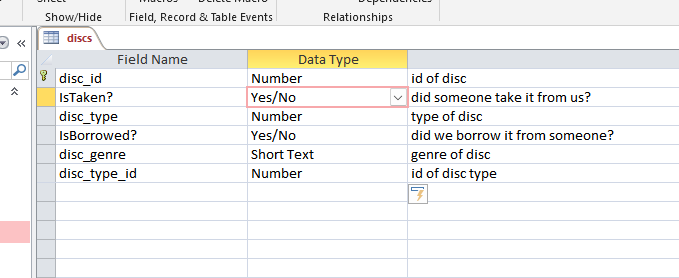


Рисунок 3 – Конструктор discs.

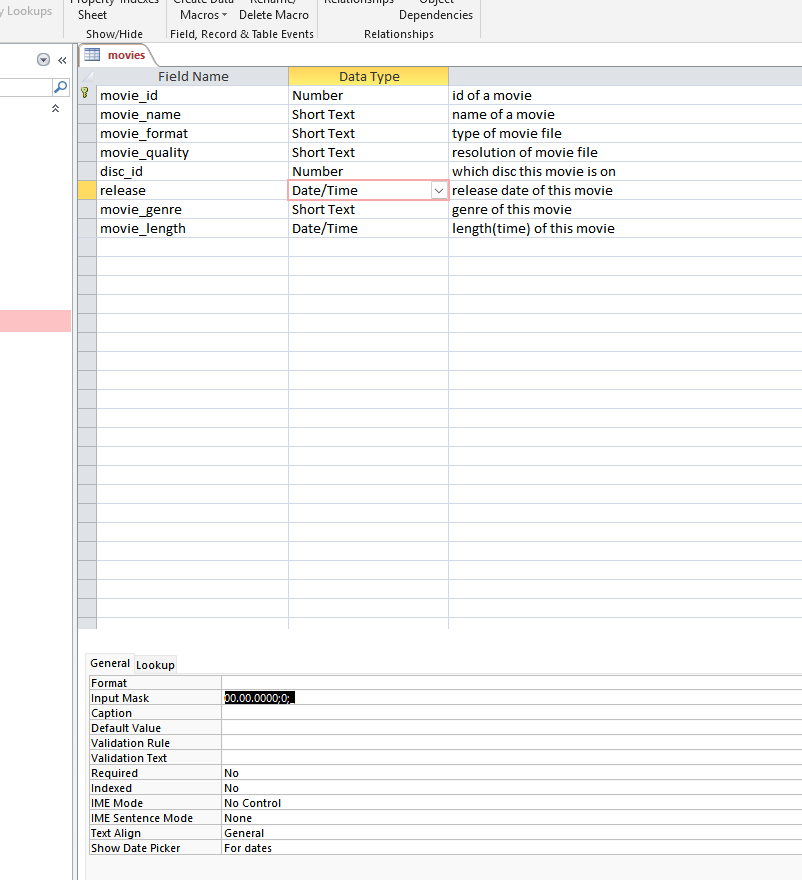


Рисунок 4 – Конструктор movies

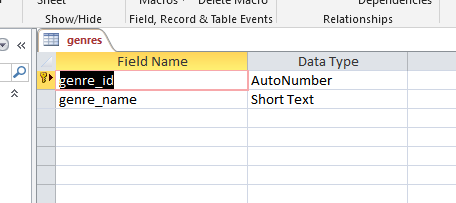


Рисунок 5 – Конструктор genres

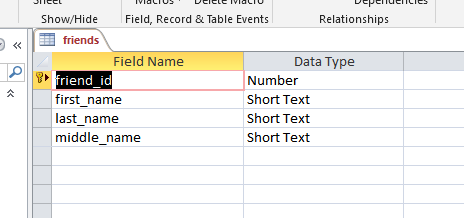


Рисунок 6 – Конструктор friends

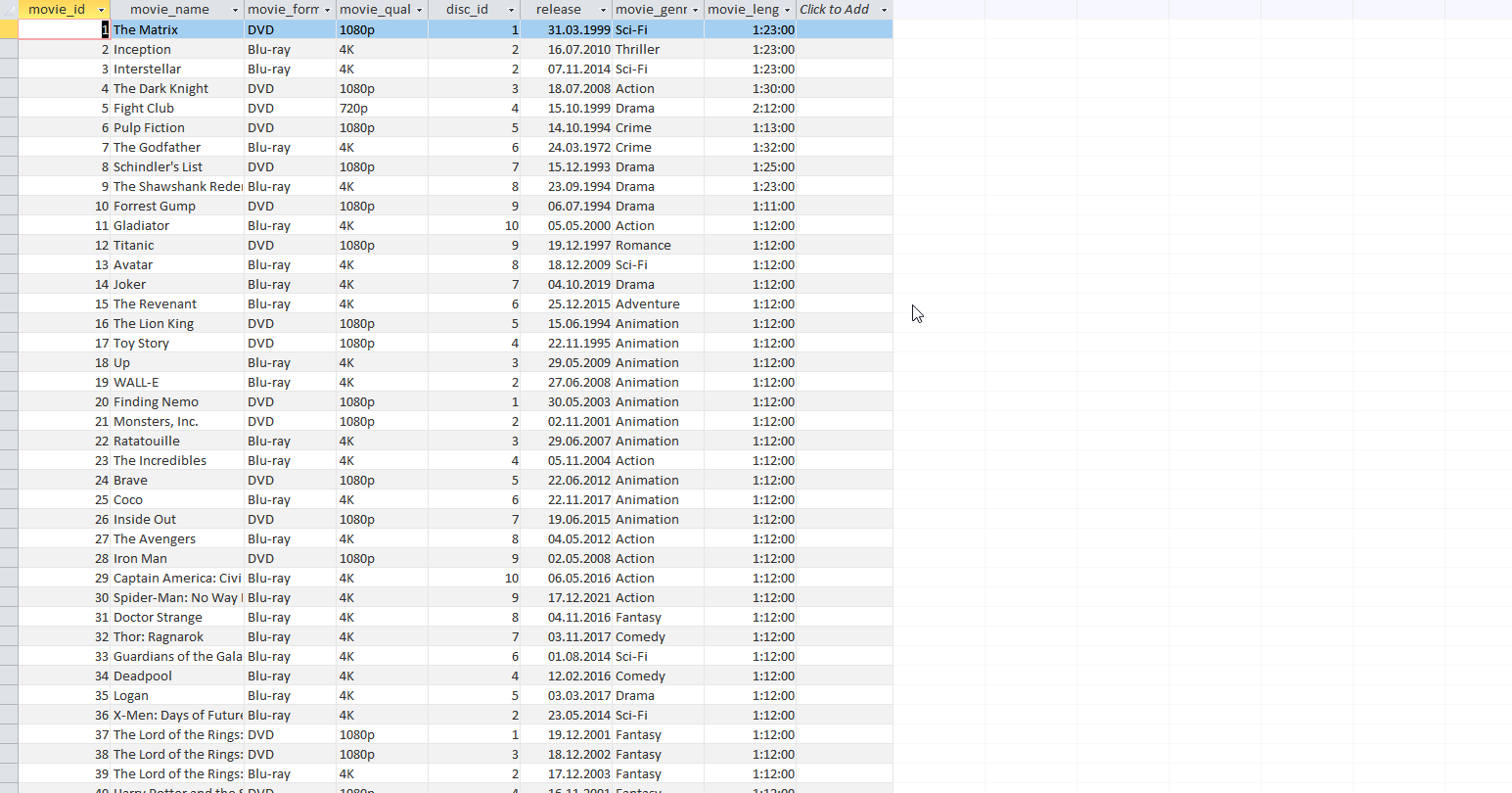


Рисунок 7 – Таблица movies.

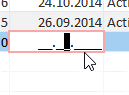


Рисунок 8 – Пример работы маски ввода.

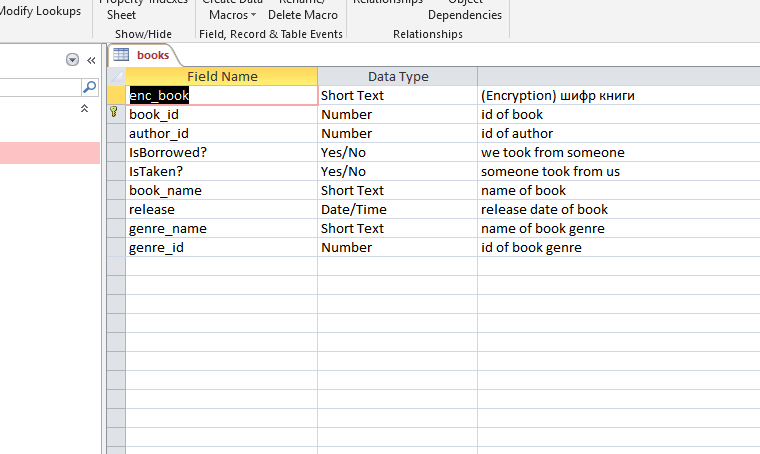


Рисунок 9 – Конструктор books.

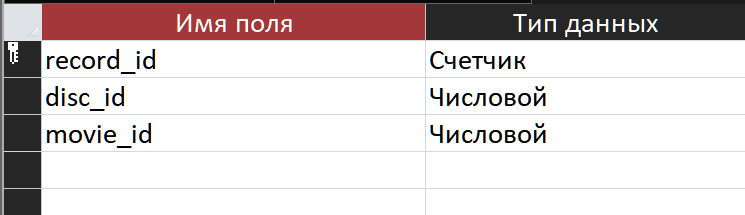


Рисунок 10 – Конструктор disc\_records

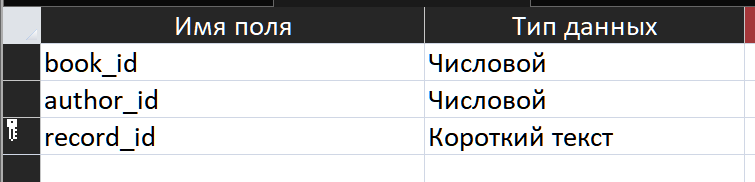


Рисунок 11 – Конструктор book\_authors.

На данных рисунках можно было увидеть отдельные таблицы БД “Домашняя библиотека книг и дисков”. Для разных полей таблиц были использованы соответствующие типы данных. К примеру – на рисунке 2 можно увидеть конструктор authors, в котором используются поля first name, last name, middle name, author\_id (key). В полях name выставлен тип данных – short text для записи имени, фамилии, отчества или второго имени соответственно. Author\_id является индексируемым числом для списка авторов книг, также является ключевым полем для соотношения связи с таблицей books.

В основе многих полей использован short text для ношения информации об предмете данной таблицы, будь то фильм (movie – Рисунок 4), жанр (genre – Рисунок 5) или друг (friends – Рисунок 6).

На рисунках 10-11 представлены промежуточные таблицы, связывающие таблицы книг и авторов, а также фильмов и дисков, так мы демонстрируем связь многие ко многим – 1 книга может иметь несколько авторов, а 1 автор может иметь несколько книг. Для фильмов аналогично – на 1 диске может быть несколько фильмов, а 1 фильм может быть на нескольких дисках.

Для удобства и унификации ввода данных о дате выхода фильмов, книг, а также дате долгов и еще длине фильмов были введены “маски ввода”, которые фиксируют определенный способ записи даты и продолжительности фильмов, книг и т.д. (Рисунок 8).

Для более точного понимания приспособления полей в таблицах были добавлены пометки, описание к столбцам, что позволяет точнее понимать для чего тот или иной столбец необходим.

Рассмотрим описание столбцов на примере books:

enc\_books – Encryption – шифр книги

book\_id – id of a book

author\_id – id of author

IsBorrowed? – we took from someone

IsTaken? – someone took from us

book\_name – name of book

Release - release date of book

genre\_name - name of book genre

genre\_id - id of book genre

В данном примере мы можем видеть название поля слева, а справа, после дефиса, описание данного поля, и хоть очевидно, что условный author\_id это, действительно, id автора, но есть и не сразу очевидные поля: IsBorrowed? и IsTaken?. Данные флажки описывают разницу между взятыми у кого-то и взятыми у нас предметами. Благодаря описанию, можно данную разницу увидеть.

Таким образом, можно сказать, что данная БД разделена на таблицы с определенными полями, имеющими свойства, типы значений, маски ввода, ограничения на ввод, а также описание столбцов для упрощения понимания их смысла – данные черты БД позволяют ей быть структурной и связанной, а также простой в реализации.

# 5. Схема базы данных:

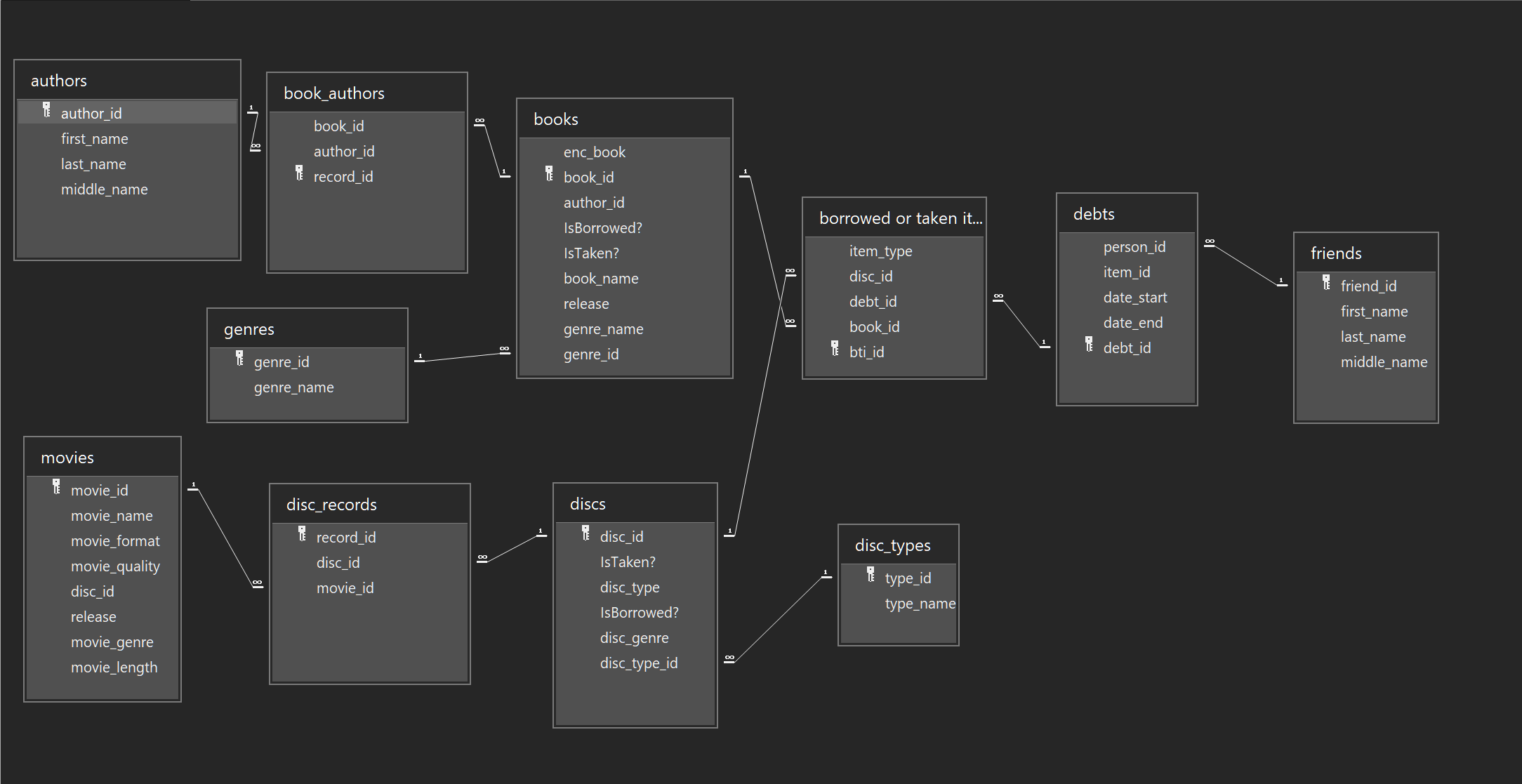


Рисунок 10 – Схема БД.

# 6. Вывод

В ходе выполнения данной лабораторной работы мы разработали базу данных в MS Access, спроектировали её структуру, настроили связи между таблицами и определили ограничения для полей.

Мы выбрали подходящие типы данных для каждого столбца:

1. Ключевые поля (ID) были заданы как Autonumber или Integer, чтобы обеспечить уникальность записей.
2. Текстовые поля, такие как название книги или имя автора, были заданы как Short Text с ограничением по количеству символов.
3. Логические флаги, такие как IsBorrowed? и isTaken?, были определены как Yes/No для удобного представления данных.
4. Дата выпуска книги была задана в формате Date/Time, что позволяет выполнять сортировку и фильтрацию по этому полю.

Мы реализовали связи между несколькими таблицами, что позволило структурировать данные:

* Один ко многим между таблицей книг и таблицей авторов (один автор может написать несколько книг).
* Один ко многим между таблицей жанров и книгами (один жанр может включать множество книг).

Эти связи обеспечили целостность данных и исключили избыточность информации.

*Ограничения и индексация*

Для первичных ключей (book\_id, author\_id, genre\_id) были заданы ограничения уникальности, чтобы исключить дублирование записей.

*Организация удобного доступа к данным*

Использование связанных таблиц позволило упростить ввод данных и избежать дублирования информации. Созданные ограничения и связи помогают поддерживать целостность базы данных при обновлении данных.

Разработанная база данных является структурированной, логически целостной и удобной для работы. Она позволяет эффективно управлять библиотечным фондом, отслеживать выдачу книг, работать с авторами и жанрами. Реализованные связи и ограничения обеспечивают правильность и непротиворечивость данных, а индексация – быструю обработку запросов.