МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ   
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА» (РУТ (МИИТ))

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

ОТЧЁТ   
ПО ПРАКТИЧЕСКОЙ РАБОТЕ

по дисциплине: «Основы построения защищённых баз данных»

на тему: «Библиотека»

Выполнила:

Ст. гр. ТКИ-541

Седых А.В.

Принял:

Доцент кафедры «УиЗИ», к.т.н.

Васильева М.А.

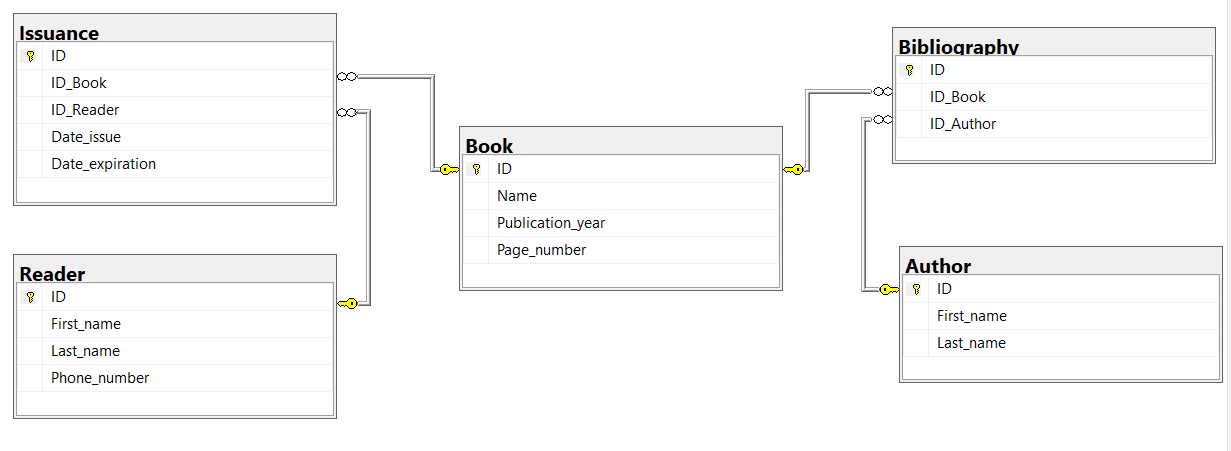
Москва 2022

Цель работы

Разработать и создать базу данных по предметной области «Библиотека». Приобрести навыки использования ORM на различных языках программирования.

Ход выполнения работы

Исходная схема созданной ранее базы данных в Microsoft SQL Server:



В данной работе реализованы таблицы «Читатель», «Выдача», «Книга» и «Автор». Между сущностями «Книга» и «Автор» реализована связь многие ко многим (исключена таблица «Библиография»).

Код программы (Django):

from django.db import models

from django.core.validators import MinValueValidator, MaxValueValidator

import datetime

from phonenumber\_field.modelfields import PhoneNumberField

class Author(models.Model):

    name = models.CharField(max\_length=255, verbose\_name="Имя")

    surname = models.CharField(max\_length=255, verbose\_name="Фамилия")

    class Meta:

        verbose\_name = 'Автор'

        verbose\_name\_plural = 'Авторы'

    def \_\_str\_\_(self):

        return f"{self.name} {self.surname}"

class Book(models.Model):

    author = models.ManyToManyField(Author, blank=True, verbose\_name="Автор")

    name = models.CharField(max\_length=255, verbose\_name="Название")

    publication\_year = models.PositiveSmallIntegerField(verbose\_name='Год публикации', validators=[

        MaxValueValidator(datetime.datetime.now().year), MinValueValidator(1)

    ])

    page\_number = models.PositiveSmallIntegerField(verbose\_name='Количество страниц', validators=[MinValueValidator(1)])

    class Meta:

        verbose\_name = 'Книга'

        verbose\_name\_plural = 'Книги'

    def \_\_str\_\_(self):

        return f"{self.name} {self.publication\_year} {self.page\_number}"

class Reader(models.Model):

    first\_name = models.CharField(max\_length=255, verbose\_name="Имя")

    last\_name = models.CharField(max\_length=255, verbose\_name="Фамилия")

    phone\_number = PhoneNumberField(verbose\_name="Номер телефона")

    class Meta:

        verbose\_name = 'Читатель'

        verbose\_name\_plural = 'Читатели'

    def \_\_str\_\_(self):

        return f"{self.first\_name} {self.last\_name} {self.phone\_number}"

class Issuance(models.Model):

    book = models.ForeignKey(Book, on\_delete=models.CASCADE, verbose\_name="Книга")

    reader = models.ForeignKey(Reader, on\_delete=models.CASCADE, verbose\_name="Читатель")

    date\_issue = models.DateField(verbose\_name="Дата выдачи")

    date\_expiration = models.DateField(verbose\_name="Дата возврата", null=True)

    class Meta:

        verbose\_name = 'Выданная книга'

        verbose\_name\_plural = 'Выданные книги'

    def \_\_str\_\_(self):

        return f"{self.book} {self.reader} {self.date\_issue} {self.date\_expiration}"

Описание сущностей:

|  |  |  |
| --- | --- | --- |
| Книга | | |
| Название поля | Тип данных | Обязательное |
| author | ManyToManyField(Author) | Да |
| name | CharField(255) | Да |
| publication\_year | DateField | Да |
| page\_number | IntegerField | Да |

|  |  |  |
| --- | --- | --- |
| Автор | | |
| Название поля | Тип данных | Обязательное |
| name | CharField(255) | Да |
| surname | CharField(255) | Да |

|  |  |  |
| --- | --- | --- |
| Читатель | | |
| Название поля | Тип данных | Обязательное |
| first\_name | CharField(255) | Да |
| last\_name | CharField(255) | Да |
| phone\_number | PhoneNumberField | Да |

|  |  |  |
| --- | --- | --- |
| Выданная книга (выдача) | | |
| Название поля | Тип данных | Обязательное |
| book | ForeignKey(Book) | Да |
| reader | ForeignKey(Reader) | Да |
| date\_issue | DateField | Да |
| date\_expiration | DateField | Да |

Администрирование Django (работа с полученной базой данных после создания классов).

|  |
| --- |
| Главная страница – изменение данных таблиц «Авторы», «Выданные книги», «Книги» и «Читатели». |
|  |

|  |
| --- |
| Таблица «Автор». (Имя и фамилия автора) |
|  |
|  |

|  |
| --- |
| Таблица «Книги». (Название книги, год выпуска и количество страниц) |
| Изображение выглядит как текст  Автоматически созданное описание |
| Изображение выглядит как текст  Автоматически созданное описание |

|  |
| --- |
| Таблица «Читатель». (Имя и фамилия читателя, номер телефона) |
| Изображение выглядит как текст  Автоматически созданное описание |
| Изображение выглядит как текст  Автоматически созданное описание |

|  |
| --- |
| Таблица «Выданная книга». (Книга, читатель, дата взятия книги и дата возвращения книги) |
| Изображение выглядит как текст  Автоматически созданное описание |
| Изображение выглядит как текст  Автоматически созданное описание |

Вывод

В результате выполнения практической работы была создана база данных с предметной областью «Библиотека» с помощью технологии ORM Python Django.