**Министерство транспорта Российской Федерации**

**Федеральное государственное автономное образовательное  
учреждение высшего образования «РУТ (МИИТ)»**

**Институт транспортной техники и систем управления**

**Кафедра «Управление и защита информации»**

ОТЧЁТ

по дисциплине

**«Основы построения защищенных баз данных»**

на тему

**«Архив»**

Выполнил: ст. гр. ТКИ-541  
Ситников Д.Ю.  
Проверил: Васильева М.А.

**Цель работы:**

Научится использовать структуру ОРМ на языке программирования.

**Постановка задачи:**

Написать на языке Python с использованием фреймворка Django базу данных для информационного обслуживания сотрудников архива уголовного розыска. В архиве лежат Дела за разные годы.

**Выполнение работы:**

**models.py**

from django.db import models  
  
class Suplier(models.Model):  
 company\_name = models.CharField(max\_length=255, verbose\_name=**"Архив"**, unique=True)  
  
 def \_\_str\_\_(self):  
 return self.company\_name  
  
class Ingredients(models.Model):  
 name = models.CharField(max\_length=255, unique=True)  
 count = models.IntegerField()  
 delivery = models.ForeignKey(Suplier, on\_delete=models.CASCADE)  
  
 def \_\_str\_\_(self):  
 return self.name  
  
class Store(models.Model):  
 store\_name = models.CharField(max\_length=255, verbose\_name=**"ФИО"**)  
 address = models.CharField(max\_length=255, verbose\_name=**"Дело "**)  
  
 def \_\_str\_\_(self):  
 return self.store\_name  
  
class Arhiv(models.Model):  
 name = models.CharField(max\_length=255, unique=True)  
 ingredients = models.ManyToManyField(Ingredients, verbose\_name=**"Поиск дела"**)  
 transfer\_to = models.ManyToManyField(Store)  
 cost = models.IntegerField(verbose\_name=**"Статьи"**)  
  
 def \_\_str\_\_(self):  
 return self.name  
  
class Produced(models.Model):  
 arhiv = models.ForeignKey(Arhiv, on\_delete=models.CASCADE)  
 count = models.IntegerField(verbose\_name=**"Следователь"**)  
 date = models.DateField(verbose\_name=**"Дата открытия дела"**)  
  
 def \_\_str\_\_(self):  
 return self.arhiv.name  
  
class Purchase(models.Model):  
 Sell = models.ForeignKey(Arhiv, to\_field=**"name"**, verbose\_name=**"Осуждённый"**, on\_delete=models.CASCADE)  
 count = models.IntegerField()  
 date = models.DateField()  
  
 def \_\_str\_\_(self):  
 return **f"**{self.Sell} {self.count} {self.date}**"**

**Описание классов:**

**Таблица 1 – Схема отношения** **Сущность\_1** (**Entity\_1**)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | ID | Целое число | первичный ключ |
| Следователь | ID\_Investigator | Целое число | обязательное поле |
| Осужденный | ID\_ Convicted | Целое число | обязательное поле |
| Номер дела | Number | nvarchar(6) | уникальное, обязательное поле |
| Название | Name | nvarchar(10) | уникальное, обязательное поле |
| Содержание | Content | ntext | уникальное, обязательное поле |
| Год открытия дела | Yea case open | nvarchar(10) | обязательное поле |
| Год окончания дела | Yes case of close | nvarchar(10) | обязательное поле |
| Статья | Article | nvarchar(10) | обязательное поле |

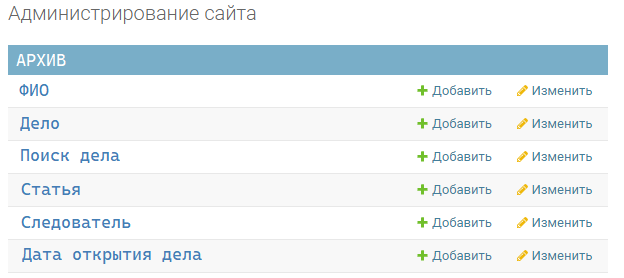
Таблица 2 – Схема отношения Сущность\_2 (Entity\_2)

|  |  |  |  |
| --- | --- | --- | --- |
| *Содержание поля* | *Имя поля* | *Тип, длина* | *Примечания* |
| Первичный ключ | ID | Целое число | первичный ключ |
| Фамилия | Surname | nvarchar(10) | обязательное поле |
| Имя | Name | nvarchar(10) | обязательное поле |

**Скриншоты работы**

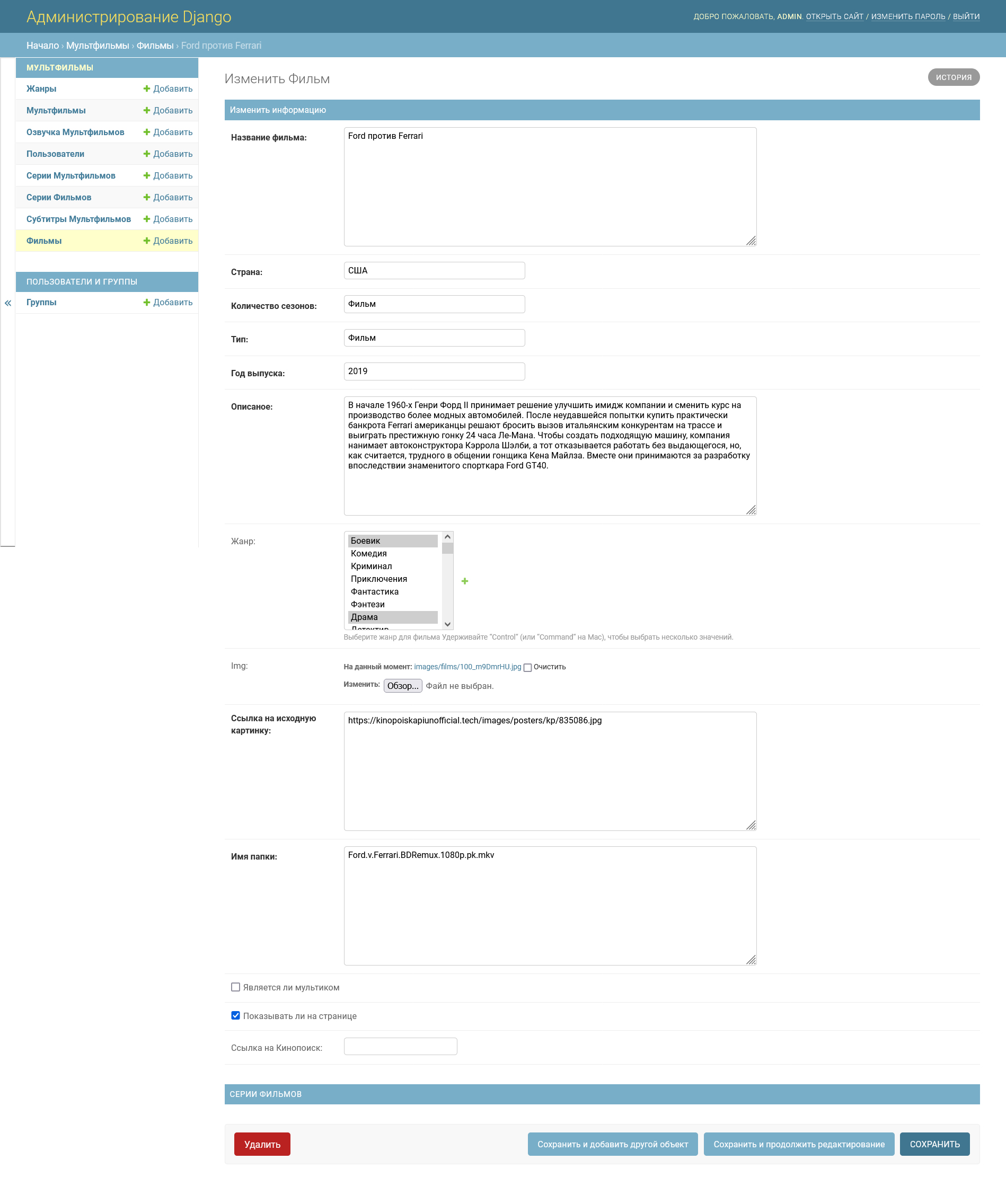
После создания классов в администрировании Django стало доступно изменение и добавления элементов в базу данных.

**Изменение данных**

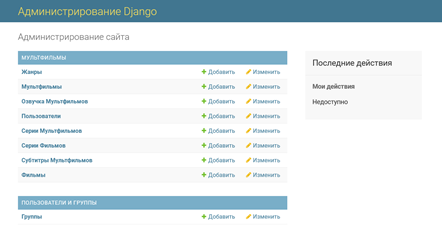
****

*Рисунок 1 Страница администратора*

**Изменение вводимых параметров в деле**

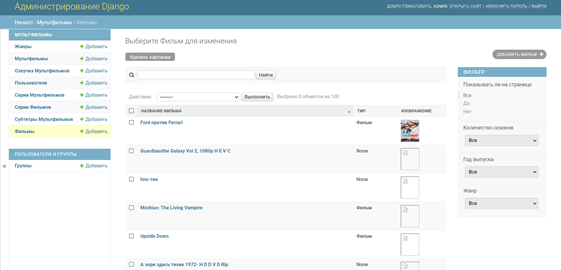


*Рисунок 2 Страница работы с делами в архиве*

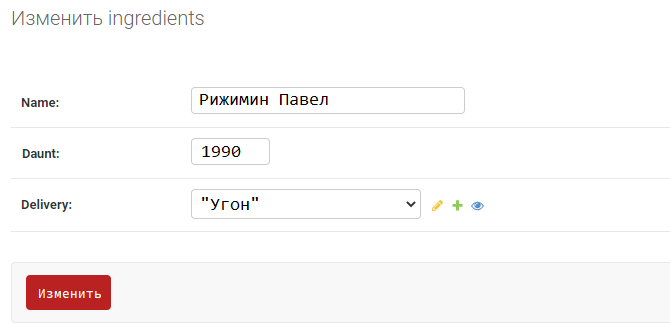
**

*Рисунок 3 Страница изменения меню*

**Изменение данных следователя**

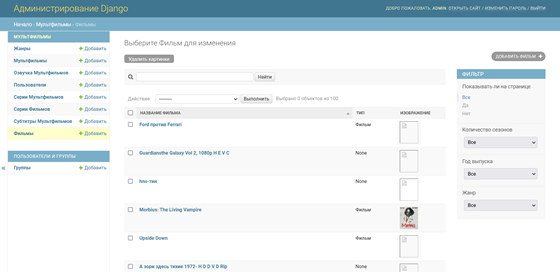
****

*Рисунок 4 Страница изменения ингредиентов*

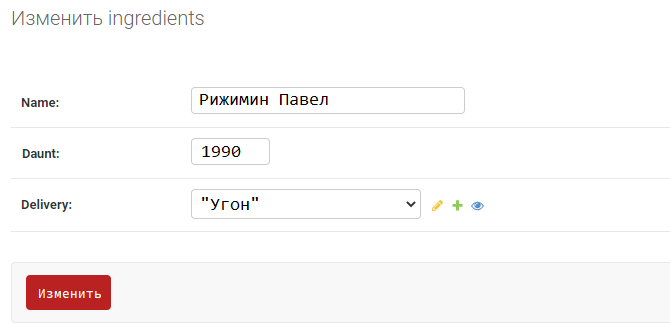
**

*Рисунок 5 Страница изменения сведений об следователе*

**Изменение данных осуждённых**

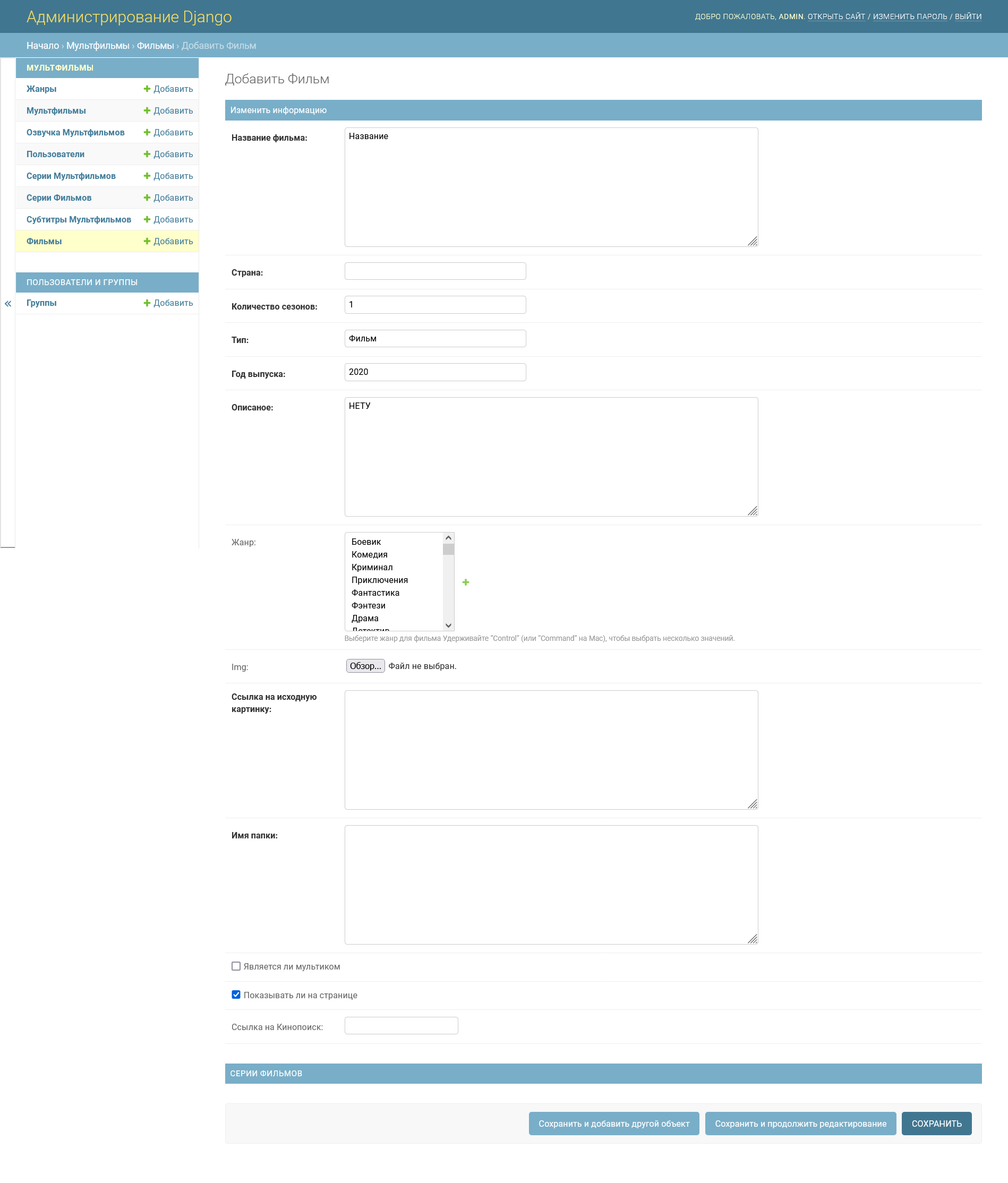
****

*Рисунок 6 Страница изменения производств*

**

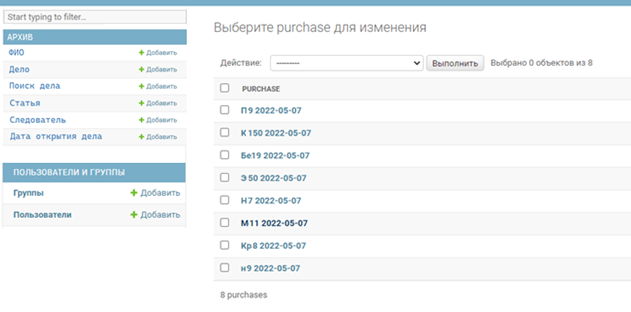
*Рисунок 7 Страница изменения производств*

**Изменение содержимого в деле**

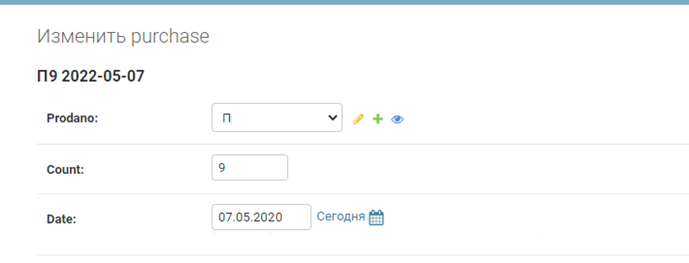


*Рисунок 8 Страница изменения данных об осуждённом*

**Изменение статей в деле**

****

*Рисунок 9 Страница изменения статей дела*

**

*Рисунок 10 Страница изменения даты работы с делом*

**Вывод:**

Была произведена работа по созданию базы данных с помощью технологии ОРМ Python Django