|  |
| --- |
| Министерство науки и высшего образования Российской Федерации |
| федеральное государственное бюджетное образовательное учреждение высшего образования |
| "Российский экономический университет имени Г.В. Плеханова" |
| **МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ** |

**Лабораторная работа №8. Реализация связи many-to-many. Вывод списка тегов. Добавление тегов к посту.**

МДК 04.01 Внедрение и поддержка компьютерных систем

ПМ 04 Сопровождение и обслуживание программного обеспечения компьютерных систем

Тема: **Основы Python**

**Специальность: 09.02.07 «Информационные системы и программирование»**

Квалификация: программист

|  |  |
| --- | --- |
| Выполнил студент:  Группы П50-2-18  Суслин А.М.  «\_\_» \_\_\_\_\_\_\_\_\_\_\_2021 год  Подпись: \_\_\_\_\_\_\_\_\_ | Проверил преподаватель:  Морозов И. А.  «\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2021 года  Оценка: \_\_\_\_\_\_\_\_\_  Подпись: \_\_\_\_\_\_\_\_ |

Цель работы: Научиться работать с возможностями Orm в представлениях, выводить данных из БД на страницу, отображение зависимых объектов (через MTM) в шаблоне).

Создаём модель Tag cо связью с моделью Post.

class Post(models.Model):

    header = models.CharField(max\_length=200, verbose\_name='Заголовок статьи')

    body = models.TextField(max\_length=500, verbose\_name='Текст статьи')

    author = models.CharField(max\_length=50, verbose\_name='Автор статьи')

    time\_post = models.DateTimeField(auto\_now=True, verbose\_name='Дата публикации статьи')

    courses = models.ManyToManyField('Tag', blank=True, related\_name='posts')

    def \_\_str\_\_(self):

        return '{}'.format(self.header)

    class Meta:

        verbose\_name = 'Статья'

        verbose\_name\_plural = "Статьи"

class Tag(models.Model):

    nameTag = models.TextField(max\_length=500, verbose\_name='Текст тега')

    def \_\_str\_\_(self):

        return self.nameTag

Добавим модель Tag в панель админа.

from django.contrib import admin

from .models import \*

from itertools import chain

class BooksAdmin(admin.ModelAdmin):

    def tags(self, obj):

        a = obj.courses.values\_list('nameTag')

        return list(chain.from\_iterable(a))

    list\_display = ('header', 'body', 'author', 'time\_post', 'tags')

admin.site.register(Post, BooksAdmin)

admin.site.register(Comment)

admin.site.register(Tag)

Результат работы:

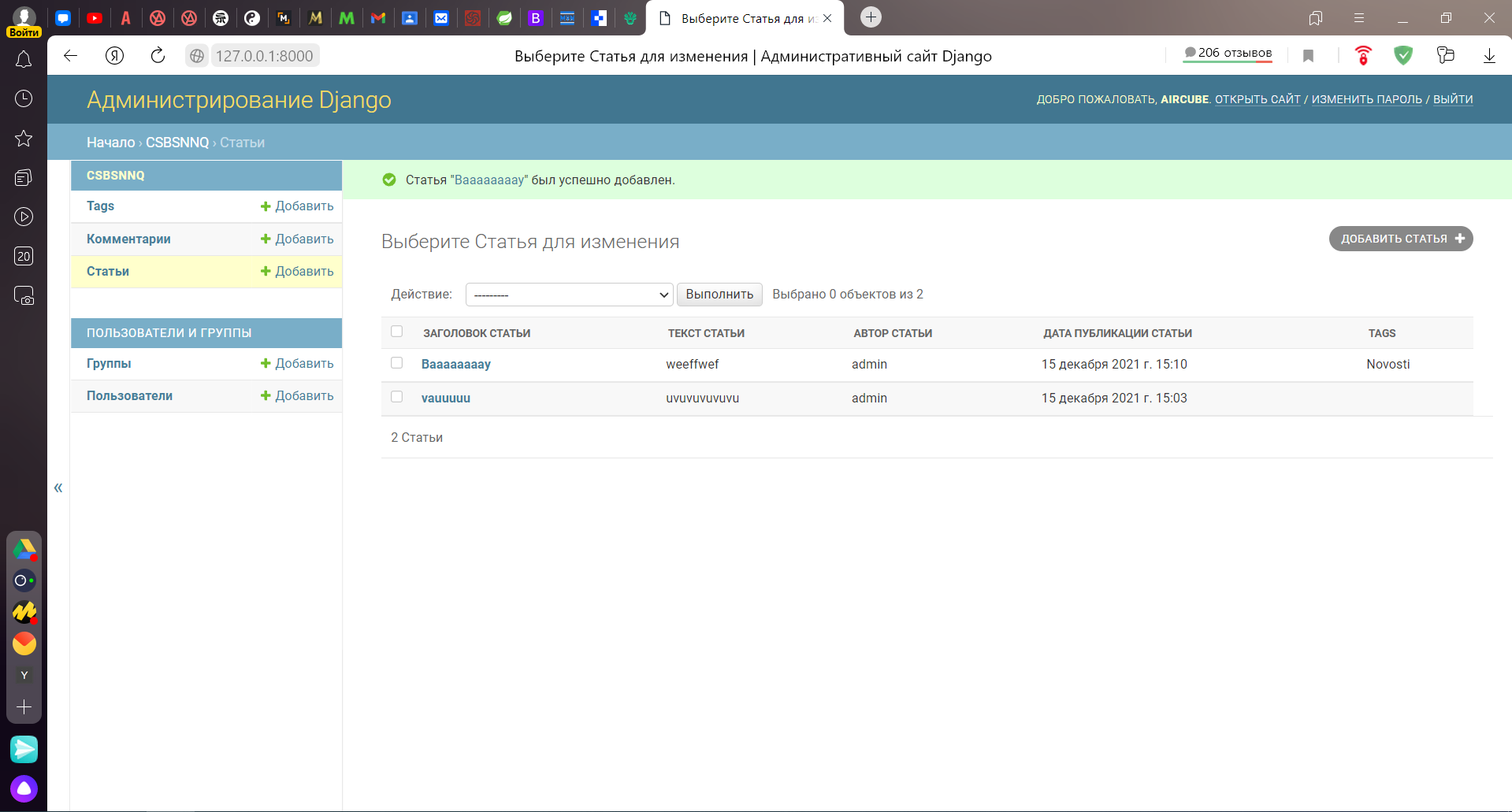


Рисунок - Выбрал тег

Вывод: в ходе данной практической работы мной были изучены возможности Orm в представлениях, выводить данные из БД на страницу, отображать зависимые объекты (через MTM) в шаблоне).