Московский государственный технический университет им. Н. Э. Баумана

Факультет «Информатика и системы управления»

Кафедра ИУ5. Курс «Разработка интернет-приложений	й»
Рубежный контроль №2	
Вариант А17.	

Проверил:

Студент группы ИУ5-51Б Гапанюк Ю. Е.

Ноздрова Валентина

Дата: 16.12.2021 Дата:

Подпись:

Задание варианта:

Вариант запросов: А

Вариант предметной области: 17

Описание классов предметной области:

- 1. Класс «Дирижер», содержащий поля:
 - ІD записи о дирижере
 - ФИО дирижера
 - Количество произведений, которые оркестр исполняет под руководством этого дирижера
 - ID записи об оркестре
- 2. Класс «Оркестр», содержащий поля:
 - ID записи об оркестре
 - Название оркестра

Задание:

- 1. Создайте проект Python Django с использованием стандартных средств Django.
- 2. Создайте модель Django ORM, содержащую две сущности, связанные отношением один-ко-многим в соответствии с Вашим вариантом из условий рубежного контроля №1.
- 3. С использованием стандартного механизма Django сгенерируйте по модели макет веб-приложения, позволяющий добавлять, редактировать и удалять данные.
- 4. Создайте представление и шаблон, формирующий отчет, который содержит соединение данных из двух таблиц.

Листинг программы:

```
from django.db import models

class Dir(models.Model):
"""

Сущность "Дирижер"
"""

fio = models.CharField(max_length=30, verbose_name='Дирижер')
pr = models.IntegerField(blank=True, null=True, verbose_name='Произведения')
orc = models.ForeignKey('Orc', models.Do_NOTHING, db_column='orc', blank=True,
null=True, verbose_name='Opxecrp')

class Meta:
    managed = False
    db_table = 'dir'

class Orc(models.Model):
"""

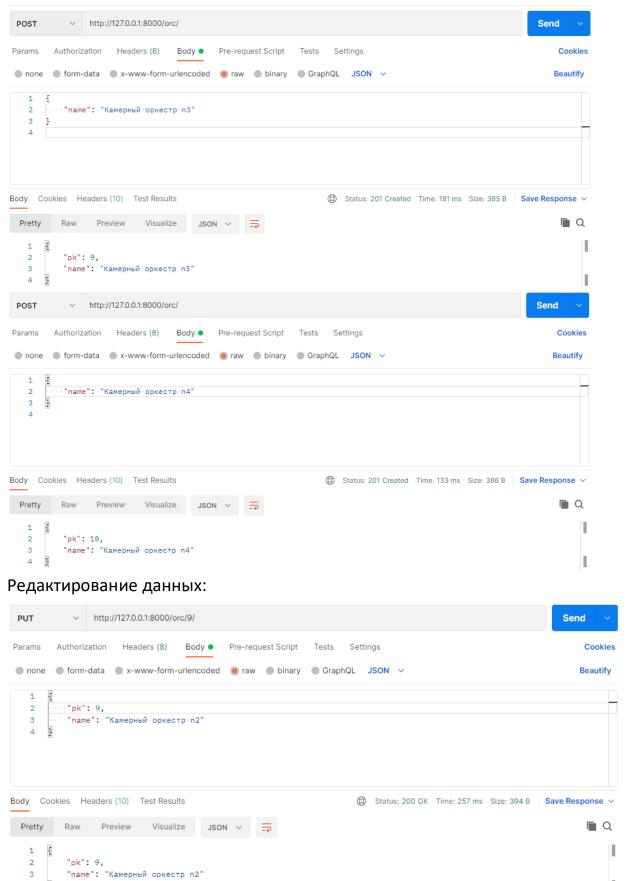
Сущность "Оркестр"
```

```
name = models.CharField(max length=50, blank=True, null=True,
verbose name='HasBahue opkecTpa')
   class Meta:
       managed = False
       db table = 'orc'
-----serializers.py------
from rk2.models import Orc
from rk2.models import Dir
from rest framework import serializers
class OrcSerializer(serializers.ModelSerializer):
Сериализатор для класса "Оркестр"
11 11 11
   class Meta:
       model = Orc
       fields = ["pk", "name"]
class DirSerializer(serializers.ModelSerializer):
Сериализатор для класса "Дирижер"
   class Meta:
       model = Dir
       fields = ["pk", "fio", "pr", "orc"]
-----views.py------
from django.shortcuts import render
from rest framework import viewsets
from rk2.serializers import OrcSerializer
from rk2.serializers import DirSerializer
from rk2.models import Orc
from rk2.models import Dir
class OrcViewSet(viewsets.ModelViewSet):
View для класса "Оркестр"
   queryset = Orc.objects.all().order by('pk')
   serializer class = OrcSerializer
class DirViewSet(viewsets.ModelViewSet):
View для класса "Дирижер"
11 11 11
   queryset = Dir.objects.all().order by('pk')
   serializer class = DirSerializer
def main(request):
View для общей таблицы
   return render(request, 'index.html', {'data': Dir.objects.all()});
```

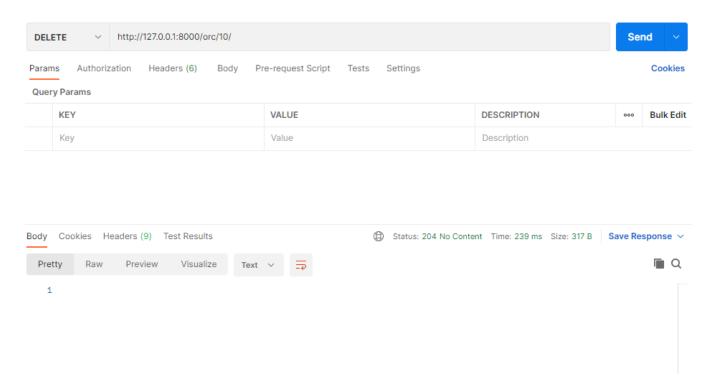
```
from django.contrib import admin
from rk2 import views as rk2 views
from django.urls import include, path
from rest framework import routers
router = routers.DefaultRouter()
router.register(r'orc', rk2_views.OrcViewSet)
router.register(r'dir', rk2_views.DirViewSet)
urlpatterns = [
   path('', include(router.urls)),
   path('api-auth/', include('rest_framework.urls', namespace='rest_framework')),
   path('admin/', admin.site.urls),
   path('main/', rk2_views.main)
1
<!DOCTYPE html>
<html lang="ru">
<head>
   <meta charset="UTF-8">
   <title>PK2 PMT</title>
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
</head>
<body>
<div class="container">
   >
          >Название оркестра
          <th>ФИО Дирижера</th>
          <th>Произведения</th>
      {% for i in data %}
      >
          { i.orc.name } } 
          { i.fio } } 
          {{ i.pr }}
      {% endfor %}
   </div>
</body>
</html>
```

Результаты выполнения:

Добавление данных с помощью Postman:



Удаление данных:



Отчет, который содержит соединение данных из двух таблиц:

