

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка Интернет-Приложений»

Отчет по Рубежному контролю №2

Вариант №17

Выполнил:

студент группы ИУ5-53
Миронова Александра

Подпись и дата:

22.12.21

Проверил:

Юрий Евгеньевич Гапанюк

Подпись и дата:

Москва, 2021 г.

Задание:

Разработать веб-приложение с использованием фреймворка Django. Веб-приложение должно выполнять следующие функции:

1. Создать проект Python Django с использованием стандартных средств Django.
2. Создать 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 сгенерировать по модели макет веб-приложения, позволяющий добавлять, редактировать и удалять данные.
4. Создать представление и шаблон, формирующий отчет, который содержит соединение данных из двух таблиц.

Вариант В.

Сущности «Оркестр» и «Дирижер» связаны соотношением один-ко-многим.

Описание предметной области:

Класс данных OrchesraStock содержит информацию о каждом оркестре
Поля класса OrchesraStock:

- Название оркестра
- Количество участников оркестра
- Дата формирования оркестра

Класс данных BatonistStock содержит информацию о каждом дирижере
Поля класса BatonistStock:

- Имя дирижера
- Фамилия дирижера
- Возраст дирижера
- Стаж дирижера
- Id оркестра, где работает дирижер (связь один ко многим)

Выполнение задания с экранными формы с примерами выполнения программы:

1)Создание моделей классов

Класс OrchesraStock:

```
class OrchesraStock(models.Model):  
    nameOrch = models.CharField(max_length=50, verbose_name="Название оркестра")  
    membersNum = models.DecimalField(max_digits=8, decimal_places=0, verbose_name="Количество участников")  
    dateCreation = models.DateTimeField(verbose_name="Дата формирования")
```

Класс BatonistStock:

```
class BatonistStock(models.Model):
    nameBat = models.CharField(max_length=30, verbose_name="Имя дирижера")
    secondName = models.CharField(max_length=200, verbose_name="Фамилия дирижера")
    age = models.DecimalField(max_digits=3, decimal_places=0, verbose_name="Возраст дирижера")
    experience = models.DecimalField(max_digits=3, decimal_places=0, verbose_name="Стаж дирижера")
    idOrch = models.ForeignKey('OrchesraStock', models.DO_NOTHING, db_column='idOrch')
```

Затем по данным моделям с помощью миграций были созданы таблицы.

2) Заполнение таблиц осуществлялось с использованием программы Insomnia, при помощи Django REST Framework. С помощью него реализуется добавление, удаление и изменение данных в таблицах.

Для преобразования моделей приложения из базы данных в JSON были написаны 2 сериализатора

Для модели OrchesraStock

```
class OrchesraStockSerializer(serializers.ModelSerializer):
    class Meta:
        # Модель, которую мы сериализуем
        model = OrchesraStock
        # Поля, которые мы сериализуем
        fields = ["pk", "nameOrch", "membersNum", "dateCreation"]
```

Для модели BatonistStock

```
class BatonistStockSerializer(serializers.ModelSerializer):
    class Meta:
        # Модель, которую мы сериализуем
        model = BatonistStock
        # Поля, которые мы сериализуем
        fields = ["pk", "nameBat", "secondName", "age", "experience", "idOrch"]
```

Создание запросов к таблицам

```
class OrchesraStockViewSet(viewsets.ModelViewSet):
    queryset = OrchesraStock.objects.all().order_by('pk')
    serializer_class = OrchesraStockSerializer

class BatonistStockViewSet(viewsets.ModelViewSet):
    queryset = BatonistStock.objects.all().order_by('pk')
    serializer_class = BatonistStockSerializer
```

Добавление роутера в URL'ы приложения

```
router = routers.DefaultRouter()
router.register('Orchestra', stock_views.OrchesraStockViewSet)
router.register('Batonist', stock_views.BatonistStockViewSet)
```

Заполненная таблица OrchesraStock

```
1 ▾ [  
2 ▾ {  
3   "pk": 1,  
4   "nameBat": "Anna",  
5   "secondName": "Yanova",  
6   "age": "43",  
7   "experience": "10",  
8   "idOrch": 1  
9 },  
10 ▾ {  
11   "pk": 2,  
12   "nameBat": "Dmitriy",  
13   "secondName": "Padilov",  
14   "age": "33",  
15   "experience": "8",  
16   "idOrch": 1  
17 },  
18 ▾ {  
19   "pk": 3,  
20   "nameBat": "Gagic",  
21   "secondName": "Macichan",  
22   "age": "31",  
23   "experience": "5",  
24   "idOrch": 2  
25 },  
26 ▾ {  
27   "pk": 4,  
28   "nameBat": "Ivan",  
29   "secondName": "Sokolov",  
30   "age": "34",  
31   "experience": "9",  
32   "idOrch": 3  
33 },  
34 {  
35   "pk": 5,  
36   "nameBat": "Nikolay",  
37   "secondName": "Kochetkov",  
38   "age": "40",  
39   "experience": "12",  
40   "idOrch": 3  
41 },  
42 {  
43   "pk": 6,  
44   "nameBat": "Vladimir",  
45   "secondName": "Volkov",  
46   "age": "41",  
47   "experience": "10",  
48   "idOrch": 4  
49 },  
50 {  
51   "pk": 7,  
52   "nameBat": "Kamila",  
53   "secondName": "Gilyajeva",  
54   "age": "30",  
55   "experience": "8",  
56   "idOrch": 5  
57 },  
58 {  
59   "pk": 8,  
60   "nameBat": "Michail",  
61   "secondName": "Antropov",  
62   "age": "27",  
63   "experience": "3",  
64   "idOrch": 6  
65 },  
66 {  
67   "pk": 9,  
68   "nameBat": "Aleksandr",  
69   "secondName": "Abramov",  
70   "age": "48",  
71   "experience": "18",  
72   "idOrch": 6  
73 }  
74 ]
```

Заполненная таблица BatonistStock

```
[  
  {  
    "pk": 1,  
    "nameOrch": "Simple music ensemble",  
    "membersNum": "20",  
    "dateCreation": "2008-11-22T00:00:00Z"  
  },  
  {  
    "pk": 2,  
    "nameOrch": "Neorchestra",  
    "membersNum": "13",  
    "dateCreation": "2007-03-12T00:00:00Z"  
  },  
  {  
    "pk": 3,  
    "nameOrch": "Imperialis Orchestra",  
    "membersNum": "28",  
    "dateCreation": "2010-08-05T00:00:00Z"  
  },  
  {  
    "pk": 4,  
    "nameOrch": "TCHAIKOVSKY ORCHESTRA",  
    "membersNum": "18",  
    "dateCreation": "1989-04-19T00:00:00Z"  
  },  
  {  
    "pk": 5,  
    "nameOrch": "Acapella-Sakartvelo",  
    "membersNum": "12",  
    "dateCreation": "2003-09-16T00:00:00Z"  
  },  
  {  
    "pk": 6,  
    "nameOrch": "Russian Folk Orchestra Moscow",  
    "membersNum": "80",  
    "dateCreation": "1997-01-20T00:00:00Z"  
  }  
]
```

3) Вывод отчета с введенными данными

Для этого создан файл orchestras.html, в который выводится информация из обеих таблиц по запрошенным данным.

Для составления таблицы, объединяющей данные двух таблиц со связью один-ко-многим, был написан метод GetBatonist, который объединяет таблицу дирижеров и таблицу оркестров по полю idOrch.

```
def GetBatonist(request):  
    context={  
        'data':BatonistStock.objects.select_related('idOrch')  
    }  
    return render(request, 'orchestras.html', context)
```

Настройка URLов

```
urlpatterns = [  
    path('', include(router.urls)),  
    path('api-auth/', include('rest_framework.urls', namespace='rest_framework')),  
  
    path('admin/', admin.site.urls),  
    path('data/', GetBatonist)  
]
```

Для создания orchestras.html был написан шаблон в файле base.html

```
<!doctype html>  
<html lang="en">  
<head>  
    <meta charset="utf-8">  
    <title>{% block title %}{% endblock %}</title>  
</head>  
<body>  
    {% block content %}{% endblock %}  
</body>  
</html>
```

Отчет о введенных данных:

Дирижеры оркестров

Имя дирижера: Anna
Фамилия дирижера: Yanova
Возраст дирижера: 43
Стаж дирижера: 10
Оркестр: Simple music ensemble
Количество участников оркестра: 20
Дата образования оркестра: Nov. 22, 2008, midnight

Имя дирижера: Dmitriy
Фамилия дирижера: Padilov
Возраст дирижера: 33
Стаж дирижера: 8
Оркестр: Simple music ensemble
Количество участников оркестра: 20
Дата образования оркестра: Nov. 22, 2008, midnight

Имя дирижера: Gagic
Фамилия дирижера: Macichan
Возраст дирижера: 31
Стаж дирижера: 5
Оркестр: Neorchestra
Количество участников оркестра: 13
Дата образования оркестра: March 12, 2007, midnight

Имя дирижера: Ivan
Фамилия дирижера: Sokolov
Возраст дирижера: 34
Стаж дирижера: 9
Оркестр: Imperialis Orchestra
Количество участников оркестра: 28
Дата образования оркестра: Aug. 5, 2010, midnight

Имя дирижера: **Nikolay**
Фамилия дирижера: **Kochetkov**
Возраст дирижера: **40**
Стаж дирижера: **12**
Оркестр: **Imperialis Orchestra**
Количество участников оркестра: **28**
Дата образования оркестра: **Aug. 5, 2010, midnight**

Имя дирижера: **Vladimir**
Фамилия дирижера: **Volkov**
Возраст дирижера: **41**
Стаж дирижера: **10**
Оркестр: **TCHAIKOVSKY ORCHESTRA**
Количество участников оркестра: **18**
Дата образования оркестра: **April 19, 1989, midnight**

Имя дирижера: **Kamila**
Фамилия дирижера: **Gilyajeva**
Возраст дирижера: **30**
Стаж дирижера: **8**
Оркестр: **Acapella-Sakartvelo**
Количество участников оркестра: **12**
Дата образования оркестра: **Sept. 16, 2003, midnight**

Имя дирижера: **Michail**
Фамилия дирижера: **Antropov**
Возраст дирижера: **27**
Стаж дирижера: **3**
Оркестр: **Russian Folk Orchestra Moscow**
Количество участников оркестра: **80**
Дата образования оркестра: **Jan. 20, 1997, midnight**

Имя дирижера: **Aleksandr**
Фамилия дирижера: **Abramov**
Возраст дирижера: **48**
Стаж дирижера: **18**
Оркестр: **Russian Folk Orchestra Moscow**
Количество участников оркестра: **80**
Дата образования оркестра: **Jan. 20, 1997, midnight**