|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

|  |  |
| --- | --- |
| ФАКУЛЬТЕТ | ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ |

|  |  |
| --- | --- |
| КАФЕДРА | СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ |

Отчет по РК2

Студент Мартынова Полина Владимировна.

*фамилия, имя, отчество*

Группа ИУ5-51Б.

Студент 15.12.2021 **Мартынова П.В.**

*подпись, дата фамилия, и.о.*

Преподаватель 15.12.2021 **Гапанюк Ю.Е.** *подпись, дата фамилия, и.о.*

2021г.

**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.

**Код:**

*models.py:*

from django.db import models  
  
  
class CD\_lib(models.Model):  
 *"""библиотека cd-дисков"""* objects = models.Manager()  
 name = models.CharField(max\_length=50, verbose\_name="Название библиотеки")  
  
  
class CD(models.Model):  
 *"""cd-диск"""* objects = models.Manager()  
 name = models.CharField(max\_length=50, verbose\_name="Название диска")  
 capacity = models.CharField(max\_length=50, verbose\_name="Ёмкость диска")  
 cdLib = models.ForeignKey(CD\_lib, on\_delete=models.CASCADE, null=True)

*urls.py:*

from django.urls import path  
from cd import views  
  
urlpatterns = [  
 path('', views.index),  
 path('create/', views.create),  
 path('create\_disk/', views.create\_disk),  
 path('edit/<int:id>/', views.edit),  
 path('edit\_lib/<int:id>/', views.edit\_lib),  
 path('delete/<int:id>/', views.delete),  
 path('delete\_lib/<int:id>/', views.delete\_lib),  
]

*views.py:*

from django.shortcuts import render  
from django.http import HttpResponseRedirect  
from django.http import HttpResponseNotFound  
from cd.models import CD  
from cd.models import CD\_lib  
  
  
def index(request):  
 libs = CD\_lib.objects.all()  
 disks = CD.objects.all()  
 return render(request, "index.html", {"libs": libs, "disks": disks})  
  
  
# сохранение данных в бд  
def create(request):  
 if request.method == "POST":  
 dl = CD\_lib()  
 dl.name = request.POST.get("name\_lib")  
 dl.save()  
 return HttpResponseRedirect("/")  
  
  
def create\_disk(request):  
 if request.method == "POST":  
 d = CD()  
 d.name = request.POST.get("name")  
 d.capacity = request.POST.get("capacity")  
 d.cdLib = CD\_lib.objects.get(name=request.POST.get("cdLib"))  
 d.save()  
 return HttpResponseRedirect("/")  
  
  
# изменение данных в бд  
def edit(request, id):  
 try:  
 disk = CD.objects.get(id=id)  
  
 if request.method == "POST":  
 disk.name = request.POST.get("name")  
 disk.capacity = request.POST.get("capacity")  
 disk.cdLib = CD\_lib.objects.get(name=request.POST.get("cdLib"))  
 disk.save()  
 return HttpResponseRedirect("/")  
 else:  
 return render(request, "edit.html", {"disk": disk})  
 except CD.DoesNotExist:  
 return HttpResponseNotFound("<h2>Disk not found</h2>")  
  
  
def edit\_lib(request, id):  
 try:  
 lib = CD\_lib.objects.get(id=id)  
  
 if request.method == "POST":  
 lib.name = request.POST.get("name")  
 lib.save()  
 return HttpResponseRedirect("/")  
 else:  
 return render(request, "edit\_lib.html", {"lib": lib})  
 except CD\_lib.DoesNotExist:  
 return HttpResponseNotFound("<h2>Library not found</h2>")  
  
  
# удаление данных из бд  
def delete(request, id):  
 try:  
 disk = CD.objects.get(id=id)  
 disk.delete()  
 return HttpResponseRedirect("/")  
 except CD.DoesNotExist:  
 return HttpResponseNotFound("<h2>Disk not found</h2>")  
  
  
def delete\_lib(request, id):  
 try:  
 lib = CD\_lib.objects.get(id=id)  
 lib.delete()  
 return HttpResponseRedirect("/")  
 except CD.DoesNotExist:  
 return HttpResponseNotFound("<h2>Library not found</h2>")

*index.html:*

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8" />  
 <title>рк2</title>  
</head>  
<body class="container">  
 <form method="POST" action="create\_disk/">  
 {% csrf\_token %}  
  
 <h2>Добавление нового диска</h2>  
 <p>  
 <label>Введите название диска</label><br>  
 <input type="text" name="name" />  
 </p>  
 <p>  
 <label>Введите ёмкость</label><br>  
 <input type="text" name="capacity" />  
 </p>  
 <p>  
 <label>Введите название библиотеки</label><br>  
 <input type="text" name="cdLib"/>  
 </p>  
 <input type="submit" value="Сохранить" >  
 </form>  
  
 <form method="POST" action="create/">  
 {% csrf\_token %}  
  
 <h2>Добавление новой библиотеки</h2>  
 <p>  
 <label>Введите название библиотеки</label><br>  
 <input type="text" name="name\_lib" />  
 </p>  
 <input type="submit" value="Сохранить" >  
  
 <h2>Список библиотек</h2>  
 <table>  
 <tr>  
 <th>Id</th>  
 <th>Название</th>  
 </tr>  
 {% for lib in libs %}  
 <tr>  
 <td>{{ lib.id }}</td>  
 <td>{{ lib.name }}</td>  
 <td><a href="edit\_lib/{{lib.id}}">Изменить</a> | <a href="delete\_lib/{{lib.id}}">Удалить</a></td>  
 </tr>  
 {% endfor %}  
 </table>  
  
  
 {% if disks.count > 0 %}  
 <h2>Список дисков</h2>  
 <table>  
 <tr>  
 <th>Id</th>  
 <th>Название</th>  
 <th>Ёмкость</th>  
 <th>Библиотека</th>  
 </tr>  
 {% for disk in disks %}  
 <tr>  
 <td>{{ disk.id }}</td>  
 <td>{{ disk.name }}</td>  
 <td>{{ disk.capacity }}</td>  
 <td>{{ disk.cdLib.name }}</td>  
 <td><a href="edit/{{disk.id}}">Изменить</a> | <a href="delete/{{disk.id}}">Удалить</a></td>  
 </tr>  
 {% endfor %}  
 </table>  
 {% endif %}  
  
 {% for lib in libs %}  
 <h3>Диски в библиотеке {{ lib.name }}:</h3>  
 {% for disk in disks %}  
 {% if lib.name == disk.cdLib.name %}  
 <li>{{disk.name}}</li>  
 {% endif %}  
 {% endfor %}  
 {% endfor %}  
 </form>  
</body>  
</html>

*edit.html:*

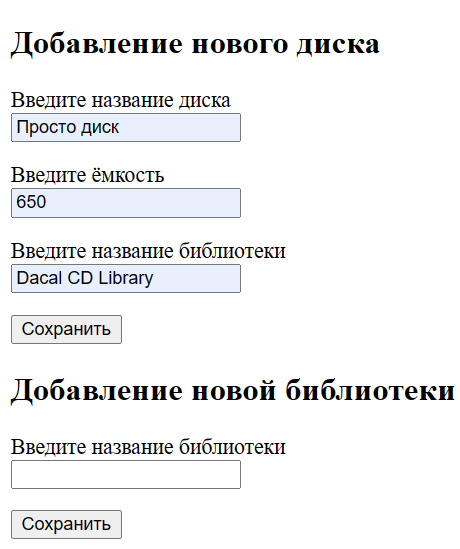
<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8" />  
 <title>edit disk</title>  
</head>  
<body class="container">  
 <form method="POST">  
 {% csrf\_token %}  
 <p>  
 <label>Введите название диска</label><br>  
 <input type="text" name="name" value="{{disk.name}}" />  
 </p>  
 <p>  
 <label>Введите ёмкость</label><br>  
 <input type="text" name="capacity" value="{{disk.capacity}}" />  
 </p>  
 <p>  
 <label>Введите название библиотеки</label><br>  
 <input type="text" name="cdLib" value="{{disk.cdLib.name}}" />  
 </p>  
 <input type="submit" value="Сохранить" >  
 </form>  
</body>  
</html>

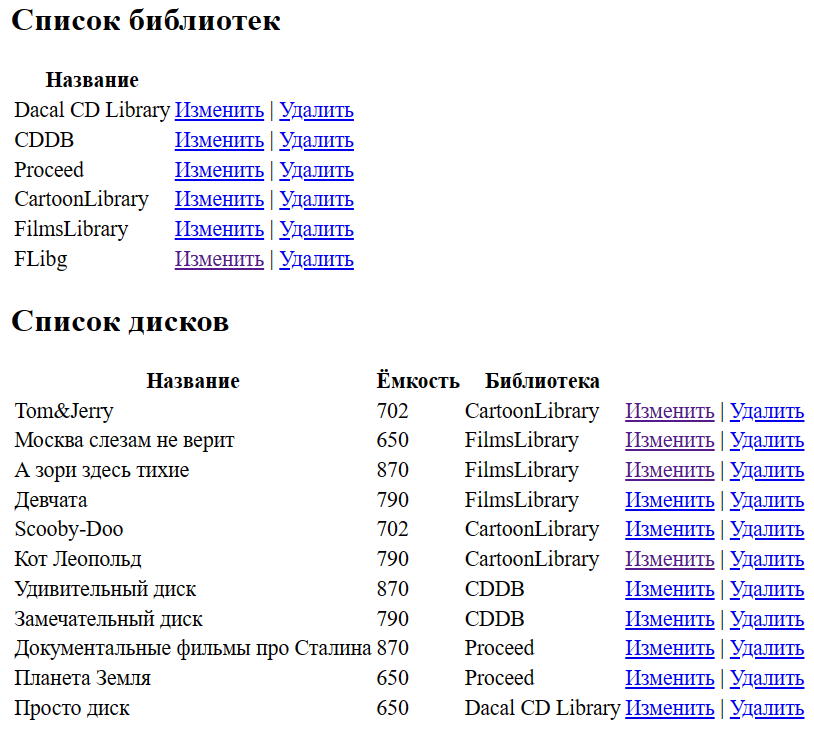
*edit\_lib.html:*

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8" />  
 <title>edit lib</title>  
</head>  
<body class="container">  
 <form method="POST">  
 {% csrf\_token %}  
 <p>  
 <label>Введите название библиотеки</label><br>  
 <input type="text" name="name" value="{{lib.name}}" />  
 </p>  
 <input type="submit" value="Сохранить" >  
 </form>  
</body>  
</html>

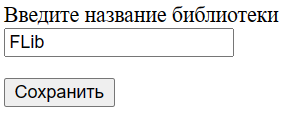
**Демонстрация работы:**

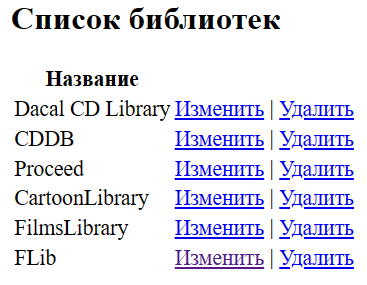
Добавим новый диск:



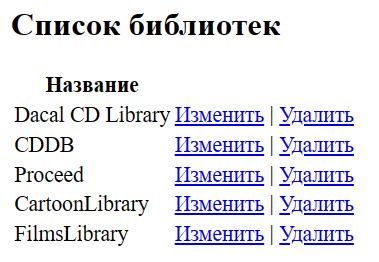


Изменим название библиотеки FLibg:



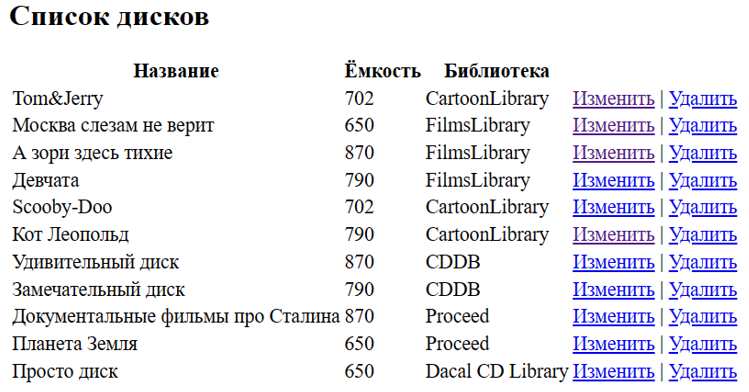


Удалим библиотеку FLib:

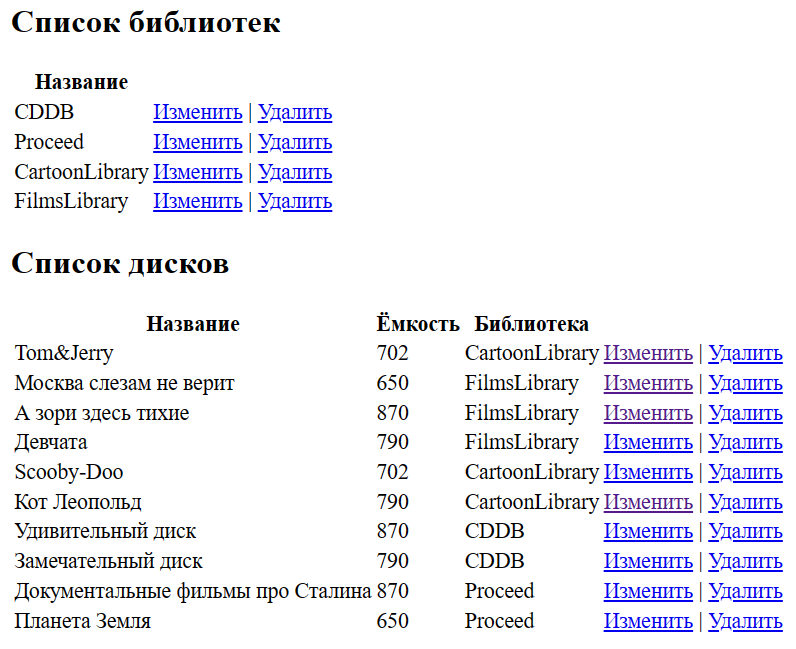


Удалим библиотеку Dacal CD Library:

Было:



Стало:



Видим, что вместе с библиотекой удалились и все диски, связанные с этой библиотекой.

Отчет, содержащий соединение данных из двух таблиц:

