

**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 лабораторной работе №5
«Работа с СУБД. Обработка данных с использованием Django ORM.»

Выполнил:
студент группы ИУ5-52Б
Перова Анна

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.

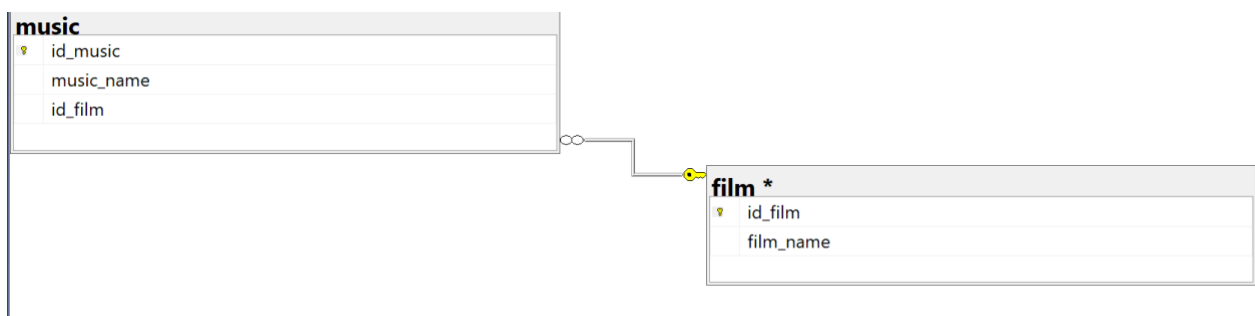
Москва, 2021 г.

Описание задания:

В этой лабораторной работе Вы познакомитесь с популярной СУБД MySQL, создадите свою базу данных. Также Вам нужно будет дополнить свои классы предметной области, связав их с созданной БД. После этого Вы создадите свои модели с помощью Django ORM, отобразите объекты из БД с помощью этих моделей.

1. Создайте сценарий с подключением к БД и несколькими запросами, примеры рассмотрены в методических указаниях.
2. Реализуйте модели Вашей предметной области из предыдущей ЛР (минимум две модели, т.е. две таблицы).
3. Создайте представления и шаблоны Django для отображения списка данных по каждой из сущностей.

Создание двух таблиц базы данных:



Скрипт подключения к базе данных, добавления записей в таблицу и выборки данных:

```
import pyodbc

connectionString = ("Driver={SQL Server Native Client 11.0};"
                    "localhost;"
                    "Database=MusicFilms;"
                    "Trusted_Connection=yes;")

request1 = "SELECT * FROM dbo.music"
```

```
request2 = "SELECT * FROM dbo.film"

connection = pyodbc.connect(connectionString, autocommit=True)
dbCursor = connection.cursor()
dbCursor.execute(request1)
dbCursor.execute(request2)
for row in dbCursor:
    print(f"{row.id_film} {row.film_name} ")
connection.commit()
dbCursor.close()
connection.close()
```

Файл main_app/models.py с созданием классов моделей таблиц на основе кода, сгенерированного с помощью команды «python manage.py inspectdb»:

```
from django.db import models

# Create your models here.
from django.db import models

class Film(models.Model):
    id_film = models.ForeignKey('self', models.DO_NOTHING, db_column='id_film',
primary_key=True)
    film_name = models.CharField(max_length=50)

    class Meta:
        managed = False
        db_table = 'film'

class Music(models.Model):
    id_music = models.IntegerField(primary_key=True)
    music_name = models.CharField(max_length=50)
    id_film = models.ForeignKey(Film, models.DO_NOTHING, db_column='id_film')

    class Meta:
        managed = False
        db_table = 'music'
```

Файл djangoProject3/urls.py:

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('main_app.urls')),
]
```

Файл main_app/urls.py:

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index),
    path('<str:model_name>/', views.list, name='list')
]
```

Файл main_app/views.py:

```
from django.shortcuts import render
from .models import *
from django.apps import apps

def index(request):
    models = apps.get_app_config('main_app').get_models()
    models_names = [model._meta.db_table for model in models]
    models_dict = {model_id: models_name for model_id, models_name in
zip(range(len(models_names)), models_names)}
    params = {'models_dict': models_dict}
    print(params)
    return render(request, 'index.html', params)

def list(request, model_name):
    models = apps.get_app_config('main_app').get_models()
    model = ''
    for elem in models:
        if elem._meta.db_table == model_name:
            model = elem
    params = { 'model_name': model._meta.db_table,
               'objects': model.objects.values() }
    return render(request, 'list.html', params)
from django.shortcuts import render
```

Файл main_app/base.html:

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8">
    <title>{% block title %}{% endblock %}</title>
</head>
<body>
    <a href="/">Главная</a>
    {% block content %}{% endblock %}
</body>
</html>
{% extends 'base.html' %}

{% block title %}
Главная
{% endblock %}
```

```
{% block content %}
<h2>Подключенная база данных содержит следующие сущности:</h2>
<ol>
    {% for model_id, model_name in models_dict.items %}
    <li><a href="{% url 'list' model_name %}">{{ model_name }}</a></li>
    <br>
    {% endfor %}
</ol>
{% endblock %}
```

Файл main_app/list.py:

```
{% extends 'base.html' %}

{% block title %}
{{ model_name }}
{% endblock %}

{% block content %}
<h2>Сущность <i>{{ model_name }}</i></h2>
<ul>
    {% for object in objects %}
    <li>
        {% for key, value in object.items %}
        <i>{{key}}</i>: {{value}}
        <br>
        {% endfor %}
    </li>
    <br>
    {% endfor %}
</ul>
{% endblock %}
```

Прототип веб-приложения:

← ⓘ ↻ 🌐 127.0.0.1:8000

Главная

[Главная](#)

Подключенная база данных содержит следующие сущности:

1. [film](#)
2. [music](#)

[Главная](#)

Сущность *film*

- *id_film_id*: 1
film_name: Titanik
- *id_film_id*: 2
film_name: Beauty
- *id_film_id*: 3
film_name: Love
- *id_film_id*: 4
film_name: love2