**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: |  | Проверил: |
| студент группы ИУ5-52Б |  | преподаватель каф. ИУ5 |
| Васильченко Дарья |  | 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.](https://github.com/iu5team/iu5web-fall-2021/blob/main/tutorials/lab5/lab5_tutorial.md)
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, подключение к базе данных, добавление записей в таблицу и выборка данных:**

import sqlite3  
  
try:  
 sqlite\_connection = sqlite3.connect('db.sqlite3')  
 cursor = sqlite\_connection.cursor()  
 print("Подключен к SQLite")  
  
 cursor.execute("""CREATE TABLE IF NOT EXISTS IT\_Company(  
 ID\_company INT PRIMARY KEY,  
 Name\_company TEXT,  
 Foundation\_year INT,  
 Specialization TEXT);  
 """)  
 sqlite\_connection.commit()  
  
 cursor.execute("""INSERT INTO IT\_Company(ID\_company, Name\_company, Foundation\_year, Specialization)  
 VALUES(1, 'Dell Technologies', 1984, 'Production of computers');""")  
 sqlite\_connection.commit()  
  
 cursor.execute("""INSERT INTO IT\_Company(ID\_company, Name\_company, Foundation\_year, Specialization)  
 VALUES(2, 'IBM', 1911, 'Software');""")  
 sqlite\_connection.commit()  
  
 cursor.execute("""INSERT INTO IT\_Company(ID\_company, Name\_company, Foundation\_year, Specialization)  
 VALUES(3, 'Cisco Systems', 1984, 'Network equipment');""")  
 sqlite\_connection.commit()  
  
 cursor.execute("""CREATE TABLE IF NOT EXISTS Founder(  
 ID\_founder INT PRIMARY KEY,  
 Founder\_name TEXT,  
 Career INT,  
 ID\_company INT);  
 """)  
 sqlite\_connection.commit()  
  
 cursor.execute("""INSERT INTO Founder(ID\_founder, Founder\_name, Career, ID\_company)  
 VALUES(1, 'Michael Dell', 'Businessman', 1);""")  
 sqlite\_connection.commit()  
  
 cursor.execute("""INSERT INTO Founder(ID\_founder, Founder\_name, Career, ID\_company)  
 VALUES(2, 'Charles Flint', 'Businessman', 2);""")  
 sqlite\_connection.commit()  
  
 cursor.execute("""INSERT INTO Founder(ID\_founder, Founder\_name, Career, ID\_company)  
 VALUES(3, 'Sandra Lerner', 'Businesswoman', 3);""")  
 sqlite\_connection.commit()  
  
 cursor.execute("""INSERT INTO Founder(ID\_founder, Founder\_name, Career, ID\_company)  
 VALUES(4, 'Leonard Bosak', 'Businessman', 3);""")  
 sqlite\_connection.commit()  
  
 cursor.execute("SELECT \* FROM Founder WHERE Founder.id\_company = 1;")  
 one\_result = cursor.fetchone()  
 print(one\_result)  
  
 cursor.close()  
  
except sqlite3.Error as error:  
 print("Ошибка при работе с SQLite", error)  
finally:  
 if sqlite\_connection:  
 sqlite\_connection.close()  
 print("Соединение с SQLite закрыто")

**Файл main\_app/models.py:**

from django.db import models  
  
  
class ITCompany(models.Model):  
 id\_company = models.AutoField(db\_column='ID\_company', primary\_key=True)  
 name\_company = models.CharField(db\_column='Name\_company', max\_length=100, blank=True, null=False)  
 foundation\_year = models.IntegerField(db\_column='Foundation\_year', blank=True, null=True)  
 specialization = models.CharField(db\_column='Specialization', max\_length=1000, blank=True, null=True)  
  
 class Meta:  
 db\_table = 'IT\_Company'  
  
  
class Founder(models.Model):  
 id\_founder = models.AutoField(db\_column='ID\_founder', primary\_key=True)  
 founder\_name = models.CharField(db\_column='Founder\_name', max\_length=100, blank=True, null=False)  
 career = models.CharField(db\_column='Career', max\_length=1000, blank=True, null=True)  
 id\_company = models.ForeignKey('ITCompany', models.DO\_NOTHING, db\_column='ID\_company', blank=True, null=True)  
  
 class Meta:  
 db\_table = 'Founder'

**Файл Lab5/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)

**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>

**Файл main\_app/index.py:**

{% 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>Cущность <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:**





