

**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  
Бабин Артём

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:

```
USE Applications;

CREATE TABLE dbo.App
(
    ID_app INT IDENTITY(1, 1) CONSTRAINT pk_app PRIMARY KEY,
    Application_name NVARCHAR(100),
    Developer NVARCHAR(100),
    ID_language INT
);

CREATE TABLE dbo.Prog_language
(
    ID_language INT IDENTITY(1, 1) CONSTRAINT pk_language PRIMARY KEY,
    Language_name NVARCHAR(100),
    Descript NVARCHAR(1000),
    Creation_year INT
);

ALTER TABLE dbo.App
ADD CONSTRAINT fk_app_lang
FOREIGN KEY (ID_language) REFERENCES Prog_language(ID_language)
```

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

```
import pyodbc

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

request1 = """ INSERT INTO dbo.Prog_language
                SELECT 'C++', 'C++ is the best programming language!', 19
83
                UNION ALL
                SELECT 'Python', 'Python is often used for machine learni
ng.', 1991 """

request2 = "SELECT * FROM dbo.Prog_language"

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

## Результат:

```
(venv) C:\Users\Mi\Documents\RIP\Lab5\Lab5>python script.py
15 C++ C++ is the best programming language! 1983
16 Python Python is often used for machine learning. 1991
```

**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

class ProgLanguage(models.Model):
    id_language = models.AutoField(db_column='ID_language', primary_key=True) # Field name made lowercase.
    language_name = models.CharField(db_column='Language_name', max_length=100, blank=True, null=False) # Field name made lowercase.
    descript = models.CharField(db_column='Descript', max_length=1000, blank=True, null=True) # Field name made lowercase.
    creation_year = models.IntegerField(db_column='Creation_year', blank=True, null=True) # Field name made lowercase.

    class Meta:
        db_table = 'Prog_language'

class App(models.Model):
    id_app = models.AutoField(db_column='ID_app', primary_key=True) # Field name made lowercase.
    application_name = models.CharField(db_column='Application_name', max_length=100, blank=True, null=False) # Field name made lowercase.
    developer = models.CharField(db_column='Developer', max_length=100, blank=True, null=True) # Field name made lowercase.
    id_language = models.ForeignKey('ProgLanguage', models.DO_NOTHING, db_column='ID_language', blank=True, null=True) # Field name made lowercase.

    class Meta:
        db_table = 'App'
```

**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`:

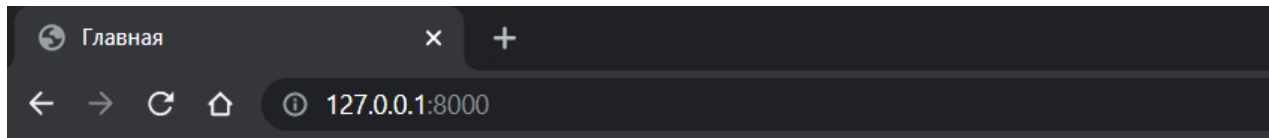
```

{% extends 'base.html' %}
{% block title %}
{{ model_name }}
{% endblock %}

{% block content %}
<h2>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 %}

```

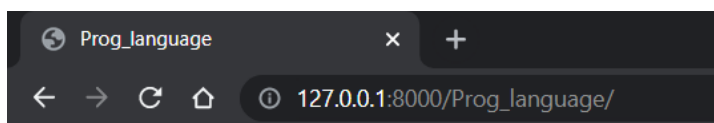
## Прототип веб-приложения:



[Главная](#)

## Подключенная база данных содержит следующие сущности:

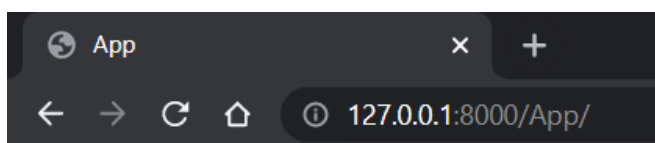
1. [Prog\\_language](#)
2. [App](#)



[Главная](#)

### Сущность *Prog\_language*

- *id\_language*: 1  
*language\_name*: C++  
*descript*: C++ is the best programming language!  
*creation\_year*: 1983
- *id\_language*: 2  
*language\_name*: Python  
*descript*: Python is often used for machine learning.  
*creation\_year*: 1991



[Главная](#)

### Сущность *App*

- *id\_app*: 1  
*application\_name*: Adobe Photoshop  
*developer*: Adobe  
*id\_language\_id*: 1
- *id\_app*: 2  
*application\_name*: Instagram  
*developer*: Meta  
*id\_language\_id*: 2