

## Отчет по лабораторной работе №6 «Работа с СУБД»

Выполнил:

Катков Александр, ИУ5-52

---

Преподаватель:

Гапанюк Ю.Е.

---

2017 г.

## 1) Задание лабораторной работы.

Вэтой 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 базой. После этого вы создадите свои модели с помощью Django ORM, отобразите объекты из БД с помощью этих моделей и ClassBasedViews. Для сдачи вы должны иметь:

1. Скрипт с подключением к БД и несколькими запросами.
2. Набор классов вашей предметной области с привязкой к СУБД (класс должен уметь хотя бы получать нужные записи из БД и преобразовывать их в объекты этого класса)
3. Модели вашей предметной области
4. View для отображения списка ваших сущностей

## 2) Листинг

### Base.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- The above 3 meta tags *must* come first in the head; any other
    head content must come *after* these tags -->
    <title>{% block title %}{% endblock %}</title>
    <link href="css/my_style.css" rel="stylesheet" type="text/css">
    <link rel="stylesheet" type="text/css" href="{% static
'css/my_style.css' %}">
    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="{% static
'css/bootstrap.min.css' %}">
    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and
    media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file://
    -> <!--[if lt IE 9]>
    <script
src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
    <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    > <![endif]>
  </head>
  <body>

    {% block body %}Переопределяется в наследниках{% endblock %}
    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></scr
ipt>
    <!-- Include all compiled plugins (below), or include individual files
    as needed -->
    <script src="static/js/bootstrap.min.js"></script>
  </body>
</html>
```

### Groups.html

```
{% extends 'base.html' %}
{% load static %}
{% block title %}Музыкальные группы{% endblock %}
{% block body %}
    <div>
        {% for order in orders %}
            <div class="alert alert-success"
role="alert"> <!---->
                <a href="#">
                    {{ order.name }} {{ order.date_create }}
                </a>
            </div>
        {% empty %}
            <li>пустой список</li>
        {% endfor %}
    </div>
{% endblock %}
```

### Views.py

```
from django.shortcuts import render from
django.views.generic import ListView from
dz_v1 import models # Create your views
here.
class
musical_group_view(ListView):
    model = models.MusicalGroup
    def get(self,
request):
        d = dict(orders=self.model.objects.values('name', 'date_create',
'description'))
        return render(request, "groups.html", d)
```

### Urls.py

```
"""untitled10 URL Configuration
The `urlpatterns` list routes URLs to views. For more information please see:
https://docs.djangoproject.com/en/1.11/topics/http/urls/
Examples:
Function views
1. Add an import: from my_app import views
2. Add a URL to urlpatterns: url(r'^$', views.home, name='home')
Class-based views
1. Add an import: from other_app.views import Home
2. Add a URL to urlpatterns: url(r'^$', Home.as_view(), name='home')
Including another URLconf
1. Import the include() function: from django.conf.urls import
url, include
2. Add a URL to urlpatterns: url(r'^blog/', include('blog.urls')) """
from django.conf.urls import url
from django.contrib import admin
from dz_v1 import views

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^groups/', views.musical_group_view.as_view()),
]
```

## 3) Результат:

MySQL Workbench interface showing the **Result Grid** and **Action Output** panels.

**Result Grid:**

| id | name   | date_create | description |
|----|--------|-------------|-------------|
| 1  | Битлз  | 1960        | Рок-группа. |
| 2  | Бит-2  | 1988        | Рок-группа. |
| 3  | Лен... | 2008        | Российск... |
| 4  | Рук... | 1993        | «Рукн Вб... |

**Action Output:**

| # | Time     | Action                                                                                                       | Message                        |
|---|----------|--------------------------------------------------------------------------------------------------------------|--------------------------------|
| 1 | 21:35:04 | SELECT * FROM first_db.django_migrations LIMIT 0, 1000                                                       | 0 row(s) returned              |
| 2 | 21:35:18 | SELECT * FROM first_db.dz_v1_musicalgroup LIMIT 0, 1000                                                      | 2 row(s) returned              |
| 3 | 21:41:09 | UPDATE first_db.dz_v1_musicalgroup SET 'name'='Битлз', 'date_create'='1960', 'description'='The Beatles ...  | 1406: Data too long for col... |
| 4 | 21:41:09 | UPDATE first_db.dz_v1_musicalgroup SET 'name'='Бит-2', 'date_create'='1988', 'description'='В 1991 году Ш... | 1406: Data too long for col... |
| 5 | 21:41:09 | INSERT INTO first_db.dz_v1_musicalgroup (id, 'name', 'date_create', 'description') VALUES (3, 'Ленингра...   | 1406: Data too long for col... |
| 6 | 21:41:09 | INSERT INTO first_db.dz_v1_musicalgroup (id, 'name', 'date_create', 'description') VALUES (4, 'Рукн вер...   | 1406: Data too long for col... |

