

**Московский государственный технический
университет им. Н.Э. Баумана.**

Факультет «Системы обработки информации и управления»

**Отчет по лабораторной работе №5 по курсу
«Разработка Интернет-Приложений»**

Тема: «Работа с СУБД.»

Выполнил:

студент группы ИУ5-53

Кириллов П. М.

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю.Е.

Подпись и дата:

Москва, 2018 г.

Задание

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 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

Порядок выполнения

1. Обращение к БД из Python.

Исходный код:

db1.py

```
import pymysql
pymysql.install_as_MySQLdb()

db=pymysql.connect(
    host='localhost',
    user='dbuser',
    passwd='123',
    db='anime'
)

c=db.cursor()

c.execute('INSERT INTO anime (name, description, author) VALUES (%s, %s, %s);', ("Унесенные призраками", "Первое, что вспомнила", "Хайао Миядзаки"))

db.commit()

c.execute('SELECT * FROM anime;')

entries=c.fetchall()

for e in entries:
    print(e)

c.close()
db.close()
```

Результат работы:

```
db1 x
C:\Users\kotsi\Desktop\lab5_bd\venv\Scripts\python.exe C:/Users/kotsi/Desktop/lab5_bd/anime/db1.py
(1, 'Rice-Tyan', 'about Coiny', 'Murad')
(2, 'Унесенные призраками', 'Первое, что вспомнила', 'Хайао Миядзаки')
Process finished with exit code 0
```

2. Написание классов предметной области с соединением с БД

Исходный код:

connection.py

```
import pymysql
pymysql.install_as_MySQLdb()

class Connection:
    def __init__(self, user, password, db, host='localhost'):
        # Сохраняем параметры соединения
        self.user = user
        self.password = password
        self.db = db
        self.host = host
        self._connection = None

    @property
    def connection(self):
        return self._connection

    def __enter__(self):
        self.connect()

    def __exit__(self, exc_type, exc_val, exc_tb):
        self.disconnect()

    def connect(self):
        """
        Открытие соединения
        """
        if not self._connection:
            self._connection = pymysql.connect(
                host=self.host,
                user=self.user,
                passwd=self.password,
                db=self.db
            )

    def disconnect(self):
        """
        Закрытие соединения
        """
        if self._connection:
            self._connection.close()
```

db2.py

```
from anime.connection import Connection

class Anime:
    def __init__(self, db_connection, name, description, author):
        self.db_connection = db_connection.connection
        self.name = name
        self.description = description
        self.author = author
```

```

def save(self):
    c = self.db_connection.cursor()
    c.execute("INSERT INTO anime (name, description, author) VALUES (%s, %s, %s);", (self.name, self.description, self.author))
    self.db_connection.commit()
    c.close()

con = Connection('dbuser', '123', 'anime')

with con:
    anime = Anime(con, 'Атака титанов', 'Второе, что вспомнила', 'Хадзимэ Исаёма')
    anime.save()

```

Результат работы:

```
mysql> SELECT * FROM anime;
```

id	name	description	author
1	Rice-Tyan	about Coiny	Murad
2	Унесенные призраками	Первое, что вспомнила	Хайао Миядзаки
3	Атака титанов	Второе, что вспомнила	Хадзимэ Исаёма

```
3 rows in set (0.00 sec)
```

```
mysql>
```

3. Django ORM

Исходный код:

Изменения в settings.py

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'anime',
        'USER': 'dbuser',
        'PASSWORD': '123',
        'HOST': 'localhost',
        'PORT': 3306,
        'OPTIONS': {'charset': 'utf8'},
        'TEST_CHARSET': 'utf8',
    }
}

```

models.py

```

from django.db import models

class AnimeModel(models.Model):
    class Meta:
        db_table='anime_Table'
    name = models.CharField(max_length=30)

```

```

description = models.CharField(max_length=255)
author = models.CharField(max_length=50)

def __unicode__(self):
    dict = {}
    dict['name'] = self.name
    dict['description'] = self.description
    dict['author'] = self.author
    return dict

```

migrations осле применения миграции:

```

# Generated by Django 2.1.3 on 2018-11-18 10:00

from django.db import migrations, models

class Migration(migrations.Migration):

    initial = True

    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name='AnimeModel',
            fields=[
                ('id', models.AutoField(auto_created=True, primary_key=True,
serialize=False, verbose_name='ID')),
                ('name', models.CharField(max_length=30)),
                ('description', models.CharField(max_length=255)),
                ('author', models.CharField(max_length=50)),
            ],
        ),
    ]

```

views.py

```

from django.shortcuts import render

# Create your views here.
from django.views import View

from anime.models import AnimeModel

class AnimeView(View):
    def get(self, request):
        #anime = AnimeModel(name="Ведьмочка Рурumo", description="Лехино
аниме", author="Ватару Ватанабэ")
        #anime.save()
        animes = AnimeModel.objects.all()
        data = {
            'animes': animes
        }
        return render(request, 'animeinfo.html', data)

```

