МГТУ им. Н.Э. Баумана

Факультет «Информатика и системы управления»

ДИСЦИПЛИНА:

«Разработка интернет-приложений»

Отчет по лабораторной работе №5

«Обработка данных с использованием Django ORM»

Выполнил:

Студент 3 курса

Факультет ИУ

Группа ИУ5-51Б

Фонканц Р.В.

Преподаватель:

Гапанюк Ю.Е.

Задание и порядок выполнения

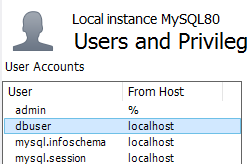
В этой лабораторной работе вы познакомитесь с популярной СУБД MySQL, создадите свою базу данных. Также вам нужно будет дополнить свои классы предметной области, связав их с созданной базой. После этого вы создадите свои модели с помощью Django ORM, отобразите объекты из БД с помощью этих моделей и ClassBasedViews.

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
5. **Выполнение задания и исходный код проекта**

**Работа с MySQL:**

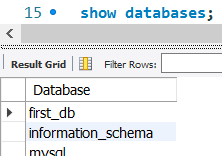
1. Создание пользователя

CREATE USER 'dbuser'@'localhost' IDENTIFIED BY '123'; 

1. Создание и использование базы данных

CREATE DATABASE first\_db CHARACTER SET utf8 COLLATE utf8\_general\_ci;

USE first\_db;



1. Привилегии пользователю:

GRANT ALL PRIVILEGES ON first\_db.\* TO 'dbuser'@'localhost';

1. Создание таблицы:

CREATE TABLE

`users` (

`id` INT(11) NOT NULL AUTO\_INCREMENT,

`name` CHAR(30) NOT NULL,

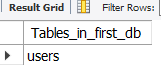
`age` SMALLINT(6) NOT NULL,

`email` VARCHAR(30) NOT NULL,

PRIMARY KEY(`id`)

);

show tables;



1. Вставка записи в таблицу:

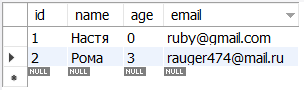
INSERT INTO users VALUES(1, 'Настя', 0, 'ruby@gmail.com');

SELECT \* from users;



**Подключение к базе данных из Python:**

import pymysql.cursors  
  
# Connect to the database  
connection = pymysql.connect(host='localhost',  
 user='dbuser',  
 password='123',  
 db='first\_db',  
 charset='utf8',  
 cursorclass=pymysql.cursors.DictCursor)  
  
try:  
 with connection.cursor() as cursor:  
  
 # Вставка записи  
 sql\_request = "INSERT INTO users (name, age, email) VALUES (%s, %s, %s)"  
 cursor.execute(sql\_request, ('Рома', '3', 'rauger474@mail.ru'))  
  
 # Зафиксировать изменения  
 connection.commit()  
  
 with connection.cursor() as cursor:  
 # Выполним выборку  
 sql\_request = "SELECT \* FROM users"  
 cursor.execute(sql\_request)  
 result = cursor.fetchall()  
  
 print(result)  
finally:  
 connection.close()



**Работа с Class Based Views:**

models.py

from django.db import models

class Post(models.Model):  
 post\_head = models.CharField(max\_length=70)  
 post\_text = models.CharField(max\_length=255)  
 publication\_date = models.DateField('Date published')  
  
 def \_\_str\_\_(self):  
 return self.post\_head

views.py

from django.views import generic

class PostView(generic.DeleteView):  
 model = Post  
 template\_name = 'post.html'

urls.py:

url(r'^(?P<pk>[0-9]+)/$', views.PostView.as\_view(), name='postDetail'