

Московский государственный технический университет 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
Ерохин И.А.
Преподаватель:
Гапанюк Ю.Е.

Москва, 2019 г.

1. 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.

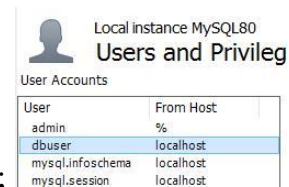
Для сдачи вы должны иметь:

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

2. Выполнение задания и исходный код проекта

Работа с MySQL:

- 1) Создание пользователя



Local instance MySQL80
Users and Privileg

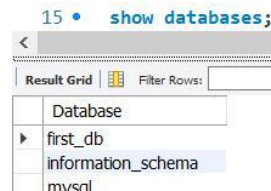
User Accounts

User	From Host
admin	%
dbuser	localhost
mysql.infoschema	localhost
mysql.session	localhost

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

- 2) Создание и использование базы данных

```
CREATE DATABASE first_db CHARACTER SET utf8 COLLATE utf8_general_ci;  
USE first_db;
```



```
15 • show databases;
```

Database
first_db
information_schema
mysql

- 3) Привилегии пользователю:

```
GRANT ALL PRIVILEGES ON first_db.* TO 'dbuser'@'localhost';
```

- 4) 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;
```



Tables_in_first_db
users

- 5) Вставка записи в таблицу:

```
INSERT INTO users VALUES(1, 'Иван', 0, '182@gmail.com');  
SELECT * from users;
```

	id	name	age	email
	1	Иван	0	182.@gmail.com

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()
```

	id	name	age	email
	1	Иван	0	182.@gmail.com
▶	2	Рома	3	rauger474@mail.ru
*	HULL	HULL	HULL	HULL

Работает с Class Based Views:

models.py

```
from django.db import models

class Post(models.Model):
    post_head = models.CharField(max_length=50)
    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'
```

url-паттерн:

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