Московский государственный технический университет им. Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Автоматизированные системы обработки информации и управления»

Отчет

Лабораторная работа № 5

По курсу «Разработка интернет-приложений»

«Обработка данных с использованием Django ORM»

ИСПОЛНИТЕЛЬ:
Матиенко Андрей
Группа ИУ5-51
_____

"__"_____2019 г.

ПРЕПОДАВАТЕЛЬ:
Гапанюк Ю.Е.
_____

"__"_____2019 г.

Москва 2019

Задание и порядок выполнения

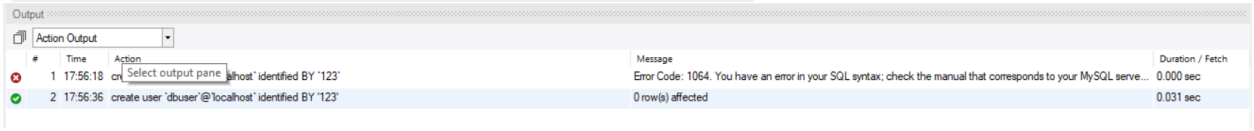
В этой лабораторной работе вы познакомитесь с популярной СУБД MySQL, создадите свою базу данных. Также вам нужно будет дополнить свои классы предметной области, связав их с созданной базой. После этого вы создадите свои модели с помощью Django ORM, отобразите объекты из БД с помощью этих моделей и ClassBasedViews.

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

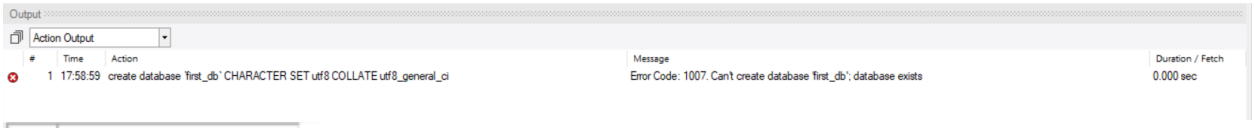
Для начала необходимо создать пользователя базы данных. У него будет доступ к этой БД.

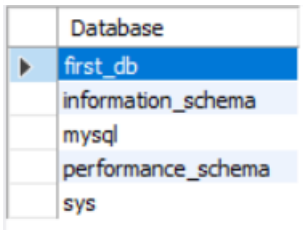
```
create user `dbuser`@`localhost` identified BY '123'
```



После этого нужно создать базу данных.

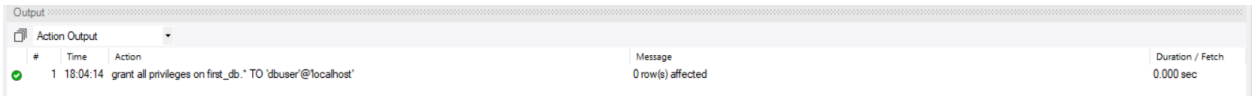
```
create database `first_db` CHARACTER SET utf8 COLLATE utf8_general_ci;
```





И выдать права новому пользователю на эту базу данных.

```
grant all privileges on first_db.* TO 'dbuser'@'localhost';
```



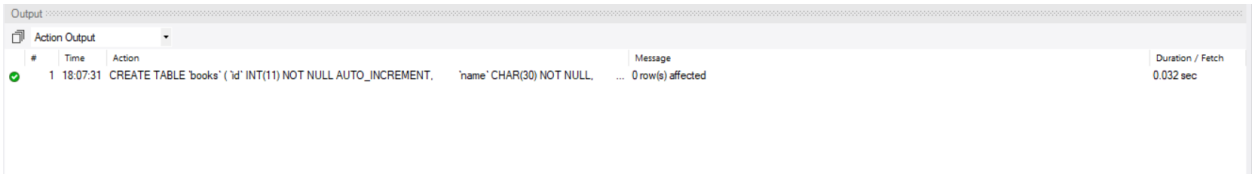
Теперь можно создавать таблицу в этой базе данных, но сначала нужно в нее перейти:

```
use first_db;
```


Создаем таблицу

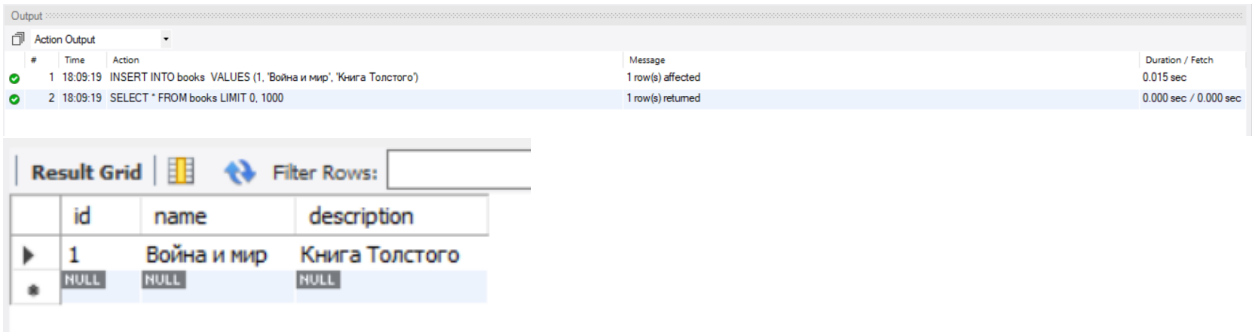
```
CREATE TABLE
    `books` (
        `id` INT(11) NOT NULL AUTO_INCREMENT,
        `name` CHAR(30) NOT NULL,
        `description` CHAR(255) NOT NULL,
        PRIMARY KEY(`id`)
    )

;
```



Добавим запись и получим ее из БД:

```
INSERT INTO books  VALUES (1, 'Война и мир', 'Книга Толстого');
SELECT * FROM books;
```



В этой части вашей задачей является написание простого скрипта, который подключается к базе данных, добавляет одну запись, затем получает и выводит на экран все записи таблицы books, а затем удаляет все записи.

```
import MySQLdb

db = MySQLdb.connect(
    host="localhost",
    user="dbuser",
    passwd="123",
    db="first_db"
)

c = db.cursor()

c.execute("INSERT INTO books (name, description) VALUES (%s, %s)", ('Book','Describe'))
db.commit()

c.execute("SELECT * FROM books;")

entries=c.fetchall()

for e in entries:
    print(e)

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

Принимает только английский

```
(1, '????? ? ???', '????? ????????')
(2, 'Book', 'Describe')
(3, 'Finansier', 'Theodore Dreiser')
```

Написание классов предметной области с соединением с БД

Класс для подключения к БД:

```
import MySQLdb

class Connection:
    def __init__(self, user, password, db, host="localhost"):...

    @property
    def connection(self):...

    def __enter__(self):...

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

    def connect(self):...

    def disconnect(self):...
```

создадим класс для книги и сделаем метод сохранения книги в БД.

```
class Book:
    def __init__(self, db_connection, name, description):
        #Сохраняем соединения и данные книги
        self.db_connection = db_connection.connection
        self.name = name
        self.description = description

    def save(self):
        # Записываем данные из объекта книги в запись БД
        c = self.db_connection.cursor()
        c.execute("INSERT INTO books (name, description) VALUES (%s, %s);",
                  (self.name, self.description))
        self.db_connection.commit()
        c.close()

con = Connection("dbuser", "123", "first_db")

with con:
    book = Book(con, 'The Fountainhead', 'Ayn Rand')
    book.save()
```

Книга добавилась:

	id	name	description
▶	1	Война и мир	Книга Толстого
	2	Book	Describe
	3	Finansier	Theodore Dreiser
	5	The Fountainhead	Ayn Rand
*	NULL	NULL	NULL

Django ORM

Для использования ORM требуется описать свои модели предметной области в виде классов, наследованных от django.db.models.Model. :

models.py:

```
from sec_file import *
from django.db import models

class BookModel(models.Model):
    name = models.CharField(max_length=30)
    description = models.CharField(max_length=255)
```

Views.py:

```
class PostView(generic.DeleteView):
    model = Post
    template_name = 'post.html'
```

Выводы:

Познакомился с БД MySQL, научился работе в ней и познакомился с Django ORM.