



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ _____

КАФЕДРА _____ СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5) _____

О Т Ч Е Т

по рубежному контролю №2

по дисциплине: Разработка Интернет-приложений _____

на тему: Django ORM _____

Студент ИУ5-54Б
(Группа)

(Подпись, дата)

(И.О.Фамилия)

Руководитель

(Подпись, дата)

(И.О.Фамилия)

2019 г.

Задание рубежного контроля

Рубежный контроль представляет собой разработку веб-приложения, которое должно выполнять следующие функции.

1. Создайте проект Python Django с использованием стандартных средств Django.
2. Создайте модель Django ORM, содержащую две сущности, связанные соотношением один-ко-многим в соответствии с вариантом из таблицы. По согласованию с преподавателем предметная область может быть изменена.
3. С использованием стандартного механизма Django сгенерируйте по модели макет веб-приложения, позволяющий добавлять, редактировать и удалять данные.
4. Создайте представление и шаблон, формирующий отчет, который содержит соединение данных из двух таблиц.

Листинги исходных модулей

Вариант: язык программирования – средство разработки

1. Создание пользователя:

```
CREATE USER 'dbuser'@'localhost'
```

2. Создание и использование базы данных:

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

3. Выдача привилегий пользователю:

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

4. Создание таблиц:

```
CREATE TABLE  
  `langs` (  
    `name` VARCHAR(30) NOT NULL,  
    `id` INT(11) NOT NULL AUTO_INCREMENT,  
    `dif_level` SMALLINT(6) NOT NULL,  
    `code_level` VARCHAR(30) NOT NULL,  
    PRIMARY KEY(`id`)  
  );
```

```
CREATE TABLE  
  `ide` (  
    `id` INT(11) NOT NULL AUTO_INCREMENT,  
    `name` VARCHAR(30) NOT NULL,  
    `developer` VARCHAR(30) NOT NULL,  
    `type` SMALLINT(6) NOT NULL,  
    PRIMARY KEY(`id`)  
  );
```

Работа с базой данных из Python:

1. Подключение к базе данных:

```
import pymysql.cursors  
  
# Connect to the database  
connection = pymysql.connect(host='localhost',  
                             user='dbuser',
```

```

        password='123',
        db='IDE_DB',
        charset='utf8',
        cursorclass=pymysql.cursors.DictCursor)

try:
    with connection.cursor() as cursor:

        # Вставка записи
        sql_request = "INSERT INTO langs (name, dif_level, code_level) VALUES (%s, %s, %s)"
        cursor.execute(sql_request, ('c++', '6', 'high'))

        # 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()

```

Остальные листинги:

- 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'

```

- Паттерн из urls.py:

```

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

```