

# Практическое задание по Django ORM

В этом задании вы научитесь работать с ORM

Для выполнения задания необходимо использовать шаблон проекта

[https://github.com/alexopryshko/coursera\\_assignment\\_tmp](https://github.com/alexopryshko/coursera_assignment_tmp)

Файл, который необходимо отредактировать **db/query.py**. В файле представлены шаблоны функций, которые нужно дополнить. Решение должно быть файлом **query.py**.

Предметная область – блоги и топики:

```
1 class User(models.Model):
2     first_name = models.CharField(max_length=255)
3     last_name = models.CharField(max_length=255)
4
5 class Blog(models.Model):
6     title = models.CharField(max_length=255)
7     author = models.ForeignKey(User)
8     created = models.DateTimeField(default=timezone.now())
9
10    subscribers = models.ManyToManyField(User, related_name='subscriptions')
11
12 class Topic(models.Model):
13     title = models.CharField(max_length=255)
14     blog = models.ForeignKey(Blog)
15     author = models.ForeignKey(User)
16     created = models.DateTimeField(default=timezone.now())
17
18     likes = models.ManyToManyField(User, related_name='likes')
```

Необходимо написать запросы:

## 1. Создание (функция create):

- Создать пользователя first\_name = u1, last\_name = u1.
- Создать пользователя first\_name = u2, last\_name = u2.
- Создать пользователя first\_name = u3, last\_name = u3.
- Создать блог title = blog1, author = u1.
- Создать блог title = blog2, author = u1.
- Подписать пользователей u1 u2 на blog1, u2 на blog2.
- Создать топик title = topic1, blog = blog1, author = u1.
- Создать топик title = topic2\_content, blog = blog1, author = u3, created = 2017-01-01.

- Лайкнуть topic1 пользователями u1, u2, u3.

## 2. Редактирование:

- Поменять first\_name на uu1 у всех пользователей (функция **edit\_all**).
- Поменять first\_name на uu1 у пользователей, у которых first\_name u1 или u2 (функция **edit\_u1\_u2**).

## 3. Удаление:

- удалить пользователя с first\_name u1 (функция **delete\_u1**).
- отписать пользователя с first\_name u2 от блогов (функция **unsubscribe\_u2\_from\_blogs**).

## 4. Найти топики у которых дата создания больше 2018-01-01 (функция **get\_topic\_created\_grated**).

## 5. Найти топик у которого title заканчивается на content (функция **get\_topic\_title\_ended**).

## 6. Получить 2х первых пользователей (сортировка в обратном порядке по id) (функция **get\_user\_with\_limit**).

## 7. Получить количество топигов в каждом блоге, назвать поле topic\_count, отсортировать по topic\_count по возрастанию (функция **get\_topic\_count**).

## 8. Получить среднее количество топигов в блоге (функция **get\_avg\_topic\_count**).

## 9. Найти блоги, в которых топигов больше одного (функция **get\_blog\_that\_have\_more\_than\_one\_topic**).

## 10. Получить все топики автора с first\_name u1 (функция **get\_topic\_by\_u1**).

## 11. Найти пользователей, у которых нет блогов, отсортировать по возрастанию id (функция **get\_user\_that\_dont\_have\_blog**).

## 12. Найти топик, который лайкнули все пользователи (функция **get\_topic\_that\_like\_all\_users**).

## 13. Найти топики, у которых нет лайков (функция **get\_topic\_that\_dont\_have\_like**).