## Задача 4

Используя метод get() стандартного менеджера модели Article, получите запись со значением поля title равным Django 5 для начинающих.

P.S. На экран ничего не нужно выводить, код модели Article:

class Article(models.Model):

title = models.CharField(max\_length=200)

date = models.DateTimeField()

content = models.TextField()

status = models.IntegerField()

## Задача 5

Используя метод стандартного менеджера модели Group, получите QuerySet содержащий все записи из таблицы данной модели.

P.S. На экран ничего не нужно выводить, код модели Group:

class Group(models.Model):

title = models.CharField(max\_length=200, verbose\_name='Категория')

slug = models.SlugField(max\_length=255, unique=True, verbose\_name='URL имя')

description = models.TextField(verbose\_name='Описание')

## Задача 6

Используя метод filter() стандартного менеджера модели Category, получите QuerySet содержащий все записи из таблицы данной модели, у которых значение поля name равно строке Django 4.

P.S. На экран ничего не нужно выводить, код модели Category:

class Category(models.Model):

name = models.CharField(max\_length=50)

description = models.TextField(max\_length=255, blank=True, default='')

## Задача 7

Используя метод стандартного менеджера модели Person, удалите запись из таблицы данной модели, у которой значение поля id равно 7.

P.S. На экран ничего не нужно выводить, код модели Person:

class Person(models.Model):

first\_name = models.CharField(max\_length=30)

last\_name = models.CharField(max\_length=30)

## Задача 8

Используя метод update() стандартного менеджера модели Article, измените значение поля status на равное 1 для записи, в таблице данной модели, у которой значение поля id равно 21.

P.S. На экран ничего не нужно выводить, код модели Article:

class Article(models.Model):

title = models.CharField(max\_length=200)

date = models.DateTimeField()

content = models.TextField()

status = models.IntegerField()