Разработка программного обеспечения на языке Python

Лекция 4. Основы баз данных, СУБД

Посмотрите видеоуроки и ответьте на контрольные вопросы после лекции

Манипуляция с данными в Django

00:00 / 03:52

В данной теме рассмотрим возможные операции с моделями и соответствующими данными, создание, получение, обновление и удаление данных.

Аббревиатура CRUD обозначает четыре основные операции, которые используются при работе с базами данных.

Этот термин представляет собой сочетание первых букв английских слов: создание (create), чтение (Read), модификация (Update) и удаление (Delete). Это стандартная классификация функций для манипуляций с данными.

В Django при создании моделей данных они наследуют свое поведение от класса Model, который предоставляет базовые операции с данными. Рассмотрим список доступных операций на примере модели **Student**, содержащей поля имя и возраст.

models.py

```
1  from django.db import models
2
3
4  class Student(models.Model):
5      name = models.CharField(max_length=20)
6      age = models.IntegerField()
```

В Django для **добавления данных** в БД можно использовать два метода:

- create () (создать)
- save ( ) (сохранить).

Для добавления данных методом **create()** для нашей модели можно использовать следующий код: в метод create передать значения полей создаваемой записи.

Однако в своей сути метод create() использует другой метод - **save()**, который мы также можем использовать отдельно для добавления объекта.

Пример добавления данных.

Варианты создание объекта

•[Название модели].objects.create(имя_поля=значение)

•[имя переменной] = [Название модели] (имя_поля=значение)
[имя переменной].save()

Пример

```
student1 = Student.objects.create(name="Kate", age=20)

student2 = Student(name="Tom", age=23)
student2.save()
```

В Django **получить значение данных из БД** можно несколькими методами:

- get() и get _ or _ create - для одного объекта;
- all () - для всех объектов;
- filter () - для группы объектов по фильтру;
- exclude () - для группы объектов с исключением некоторых;
- in_bulk () - для группы объектов в виде словаря.

Рассмотрим подробнее все указанные методы.

Для **получение из бд одного объекта** из таблицы может быть использован метод **get()** . Он возвращает один объект - т. е. одну запись из БД по определенному условию, которое передается в качестве параметра. Этот метод предназначен для выборки объектов, которые имеются в единичном числе в базе данных. Если в таблице не окажется подобного объекта, или будет несколько объектов, которые соответствуют условию, то в качестве результата будет сгенерирована ошибка. Поэтому следует применять данный метод с осторожностью. В примере мы получаем объект класса Студент, у которого поле name="Tom".

get
[Название модели].objects.get([условия поиска])

```
tom = Student.objects.get(name="Tom")
print(tom.age)  # 20
```

get_or_create
[Название модели].objects.get_or_create([условия поиска])

```
student, created = Student.objects.get_or_create(
name="Tom", age=30)
```

Метод get_or_create() возвращает объект, а если его нет в бд, то добавляет в бд новый объект. Метод возвращает добавленный объект (в нашем примере переменная student) и булевое значение (created), которое хранит True, если добавление прошло успешно.

Если требуется получить все имеющиеся в таблице объекты, то применяется метод **all()**.

Если нужно получить все объекты, которые соответствуют определенному условию, то применяется метод **filter()**, который в качестве параметра принимает критерии выборки. Например, получим все объекты таблицы студент, для которых age=20.

all
[Название модели].objects.all()

```
students = Student.objects.all()
```

filter
[Название модели].objects.filter(условие поиска)

```
students = Student.objects.filter(age=20)
```

Метод **exclude()** позволяют исключить из выборки записи, которые соответвуют переданному в качестве параметра критерию. Например, исключаем записи, у которых age=20. Можно комбинировать два выше рассмотренных метода.

exclude
[Название модели].objects.exclude(условие исключения из поиска)

```
s1 = Student.objects.exclude(age=20)

s2 = Student.objects.filter(name="Tom").exclude(age=20)
```

Метод **in_bulk()** является эффективным способом для чтения большого количества записей. Он возвращает словарь, где ключи представляют id объектов, а значения по этим ключам - сами эти объекты, то есть в данном случае объекты класса Студент. В качестве параметра методу можно передать список id записей, которые требуется получить.

[Название модели].objects.in_bulk()

```
students = Student.objects.in_bulk([1,2])
print(students)
for id in students:
    print(students[id].name+" "+str(students[id].age))
```

```
{1: <Student: Student object (1)>, 2: <Student: Student object (2)>}
Kate 20
Tom 23
```

Обновление данных в БД. UPDATE

Для обновления объекта применяется метод **save()**. Здесь мы сначала в переменную student прочитали из БД все данные, информация о котором хранится в БД в записи с id=2. Затем во второй строке изменили содержимое поля age. И в третьей строке записали всю информацию в БД. Другой способ обновления объектов представляет метод **update()** в сочетании с методом **filter**.

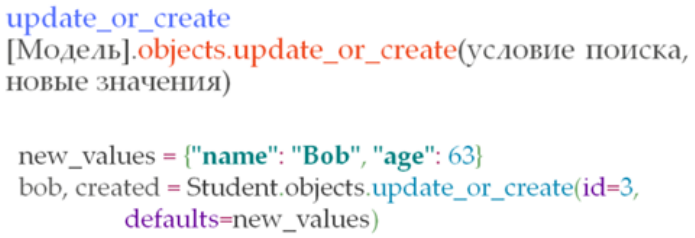
get, save

```
student = Student.objects.get(id=2)
student.age = 18
student.save()
```

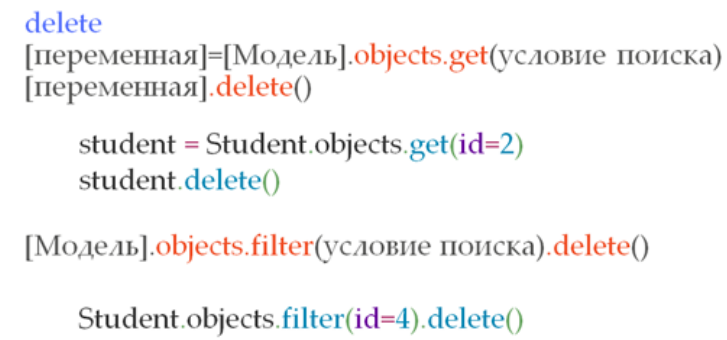
update
[Модель].objects.filter(условие поиска).update(новые значения)

```
Student.objects.filter(id=2).update(name="Mike")
```

Метод **update_or_create** обновляет запись, а если ее нет, то добавляет ее в таблицу. Он принимает два параметра. Первый представляет критерий выборки объектов, которые будут обновляться. Второй параметр представляет словарь с новыми значениями. Если критерию не соответствует никаких записей, то в таблицу добавляется новый объект.



Для удаления мы можем вызвать метод **delete()** у удаляемого объекта. Объект находим по заданному условию, например, значению id.



Подведем итоги, в данной теме мы рассмотрели набор методов, которые позволят работать с данными таблиц базы данных без использования SQL, обращаясь к каждой строке таблицы как к объекту класса (через который можем получить значения его свойств).

Операции с моделями

ПРЕДЫДУЩИЙ ЭЛЕМЕНТ КУРСА

◄ Задание 5. Добавление шаблонов страниц веб-приложения

Перейти на...

СЛЕДУЮЩИЙ ЭЛЕМЕНТ КУРСА

Задание 6. Создание базы данных ►

Контакты +7(391) 206-27-05
info-ms@sfu-kras.ru

Скачать мобильное приложение

Инструкции по работе в системе