

## HINTS

### MANEJO DE RELACIONES EN EL ORM DE DJANGO

En Django no existe la relación uno a muchos (**OneToMany**), sólo la Muchos a Uno (**ManyToOne**) usando **ForeignKey**.

1. El campo **ForeignKey** admite dos argumentos: el primero es la tabla con la que se va a relacionar, y el segundo es el tipo de eliminación de registro.
2. Para crear una relación recursiva, un objeto que tiene una relación de muchos a uno consigo mismo, use: **models.ForeignKey('self', on\_delete=models.CASCADE)**
3. Si necesita crear una relación en un modelo que aún no se ha definido, puede usar el nombre del modelo, en lugar del objeto del modelo en sí:

```
1 from django.db import models
2
3 class Car(models.Model):
4     manufacturer = models.ForeignKey(
5         'Manufacturer',
6         on_delete=models.CASCADE,
7     )
8     # ...
9
10 class Manufacturer(models.Model):
11     # ...
12     pass
```

4. Los posibles valores de **on\_delete** se encuentran en **django.db.models**:
  - **CASCADE:** eliminaciones en cascada. Django emula el comportamiento de la restricción SQL **ON DELETE CASCADE**, y también el objeto que contiene **ForeignKey**.
  - **PROTECT:** evita la eliminación del objeto al que se hace referencia generando **ProtectedError**, una subclase de **django.db.IntegrityError**.
  - **SET\_NULL:** establece el **ForeignKey** nulo; esto solo es posible si **null** es **True**.
  - **SET\_DEFAULT:** establece el **ForeignKey** en su valor predeterminado; se debe establecer un valor predeterminado para el **ForeignKey**.

- **SET():** establece el valor del **ForeignKey** con el pasado con **SET()**, o si se pasa un invocable, el resultado de llamarlo. En la mayoría de los casos, será necesario pasar un invocable para evitar ejecutar consultas en el momento en que se importe su **models.py**:

```

1 from django.conf import settings
2 from django.contrib.auth import get_user_model
3 from django.db import models
4
5 def get_sentinel_user():
6     return
7 get_user_model().objects.get_or_create(username='deleted')[0]
8
9 class MyModel(models.Model):
10     user = models.ForeignKey(
11         settings.AUTH_USER_MODEL,
12         on_delete=models.SET(get_sentinel_user),
13     )
    
```

- **DO NOTHING:** no tomar ninguna medida. Si el backend de su base de datos impone la integridad referencial, esto provocará un error **IntegrityError**, al menos que agregue manualmente una restricción SQL al campo de la base de datos **ON DELETE**.

En Django, para desarrollar la **relación muchos a muchos** (**ManyToMany**) se realiza lo siguiente:

```

1 from django.db import models
2 from App.retiro.models import retiro → (1)
3 class accesorio(models.Model):
4     tipo=models.CharField(max_lenght=50) → (3)
5 class articulo(models.Model):
6     tipor_art=models.CharField(max_lenght=50)
7     talla=models.CharField(max_lenght=3)
8     retiro=models.ForeignKey(retiro, null=True, blank=True,
9 on_delete=models.CASCADE) → (2)
10     accesorio=models.ManyToManyField(accesorio) → (4)
11
12 from django.db import models
13 class retiro(models.Model): → (1)
14     nombre=models.CharField(max_lenght=50)
15     apellido=models.CharField(max_lenght=50)
    
```

1. Importar desde la carpeta donde creamos las aplicaciones la clase a llamar.
2. Dentro de la clase, donde estamos modificando, incluir el nombre de la clase que llamaremos.
3. Crear una nueva clase a la que se le aplicará la función de enlace.

4. Incluir dentro de la clase donde nos encontramos la nueva clase creada, junto con la relación que deseamos que tenga.

Todo esto dentro del `Models.py` de la aplicación.

Luego, se realizan las migraciones con `manage.py makemigrations`, y consecuentemente `manage.py migrate`. Se importan y registran los modelos creados.

```
1 from django.contrib import admin
2 from app.apptienda.models import retiro
3 #Registrar el modelo
4 admin.site.register (retiro)
5
6 from django.contrib import admin
7 from app.apptienda.models import articulo, accesorio
8 #Registrar los modelo
9 admin.site.register (articulo)
10 admin.site.register (accesorio)
```

En Django, un ejemplo de modelo para realizar la **relación uno a uno (OneToOne)** es la de **Usuario** que tiene ya sus campos y atributos; en la práctica se puede necesitar autenticar un usuario en un blog, y crear un perfil personalizado para conectarlo al modelo **Usuario** de Django usando una relación uno a uno, como se mostrará a continuación:

1. Utilice el modelo de usuario predeterminado de Django.

```
1 from django.contrib.auth.forms import UserCreationForm
2 from django.contrib.auth.models import User
```

2. Crear una clase para heredar `UserCreationForm`. Dentro de esta clase, cree otra **metaclass** que tendrá dos variables: **model** y **fields**. La variable **model** contendrá su modelo de usuario, y la variable **fields** contendrá los campos del formulario que se crearán.

```
1 class createUserForm(UserCreationForm):
2     class meta:
3         model = User
4         fields=['username','password1','password2']
```

El código anterior creará un formulario con campos para **nombre de usuario**, **contraseña**, y **confirmación de contraseña**.

3. Crear un modelo de perfil de usuario personalizado. En su archivo `models.py` importe el modelo de usuario predeterminado.

```
1 from django.contrib.auth.models import User
```

A continuación, se debe crear su propio modelo de perfil, y también un campo usuario con una relación de uno a uno con el modelo de usuario predeterminado de Django.

```
1 class Profile(models.Model):
2     user=models.OneToOneField(User, on_delete=models.CASCADE)
3     name=models.CharField(max_length=200)
4     email=models.CharField(max_length=200)
5     def __str__(self):
6         return self.name
```

4. Crear una forma para su modelo de perfil. Abra el archivo `form.py`, e importe su modelo de perfil desde `models.py`.

```
1 from django import forms
2     from .models import Profile
```

Esto le permitirá crear un formulario de validación de usuario haciendo uso del modelo Usuario predeterminado de Django, estableciendo la relación uno a uno (**OneToOne**).