1)Creamos un modelo dentro de models.py con la siguiente información

from django.db import models

*class* costumer(*models*.*Model*):

    name = models.CharField(*max\_length*=100,*null*=True)

    phone = models.CharField(*max\_length*=100, *null*=True)

    email = models.EmailField(*max\_length*=100, *null*=True)

    date\_created = models.DateField(*auto\_now\_add*=True)

*class* Meta:

        db\_table = 'costumer'

luego, para que nos aparece en el panel de administrador, lo que tenemos que hacer es, en admin.py, registrar el modelo de la siguiente forma:

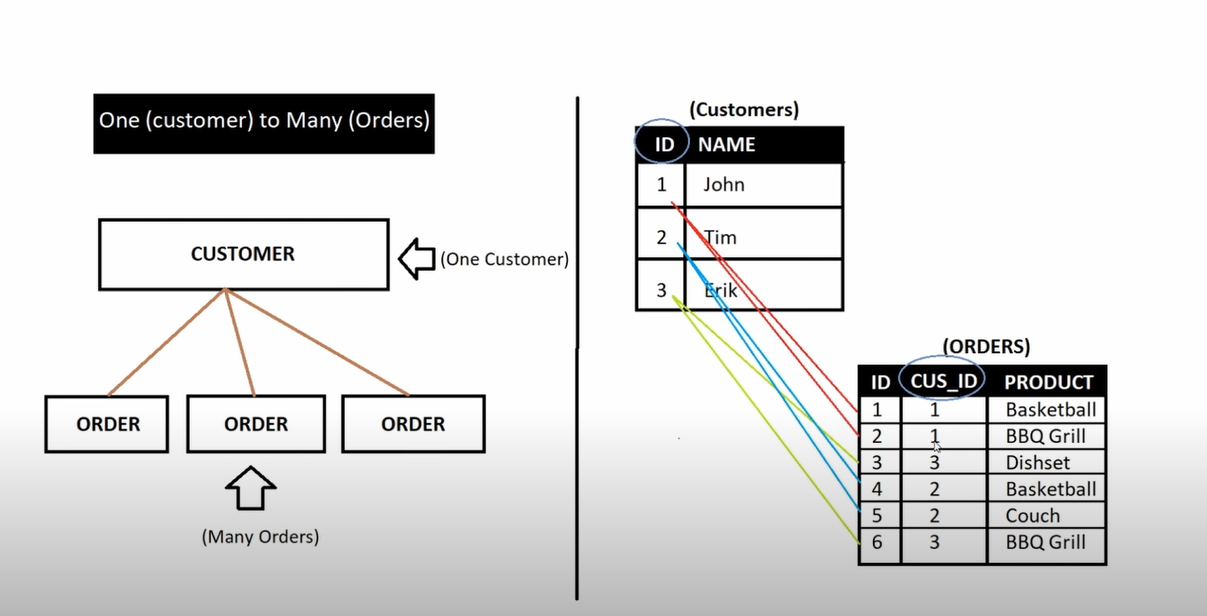
from .models import \*

admin.site.register(costumer)

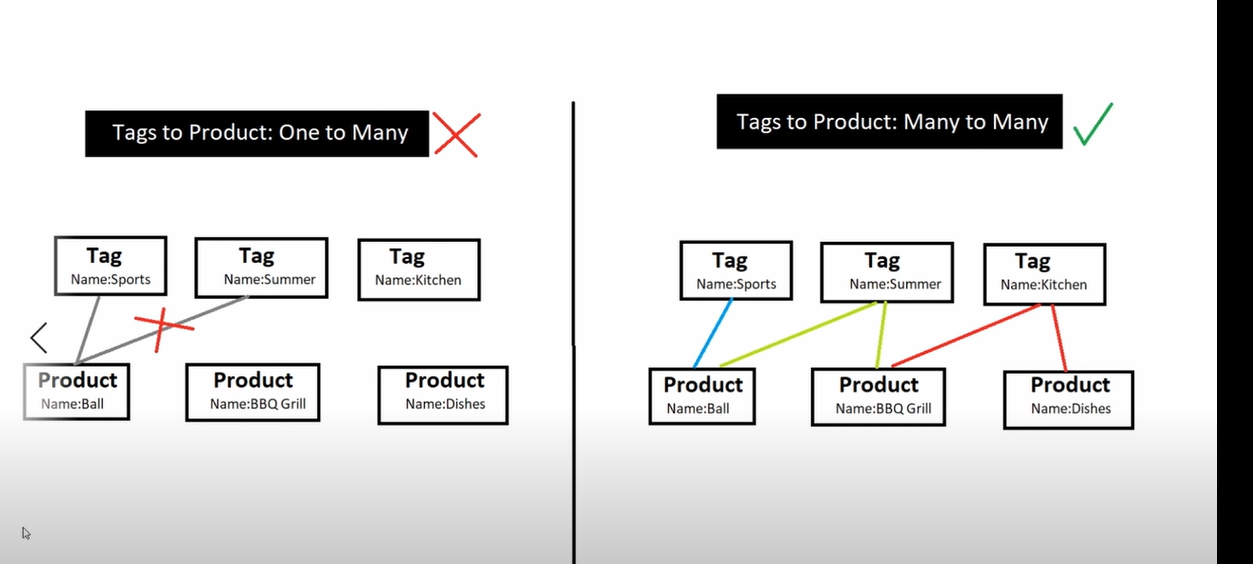
Agrego dos modelos mas y los registro en admin

Creación de una tabla one-to-many y many-to-many en Django:

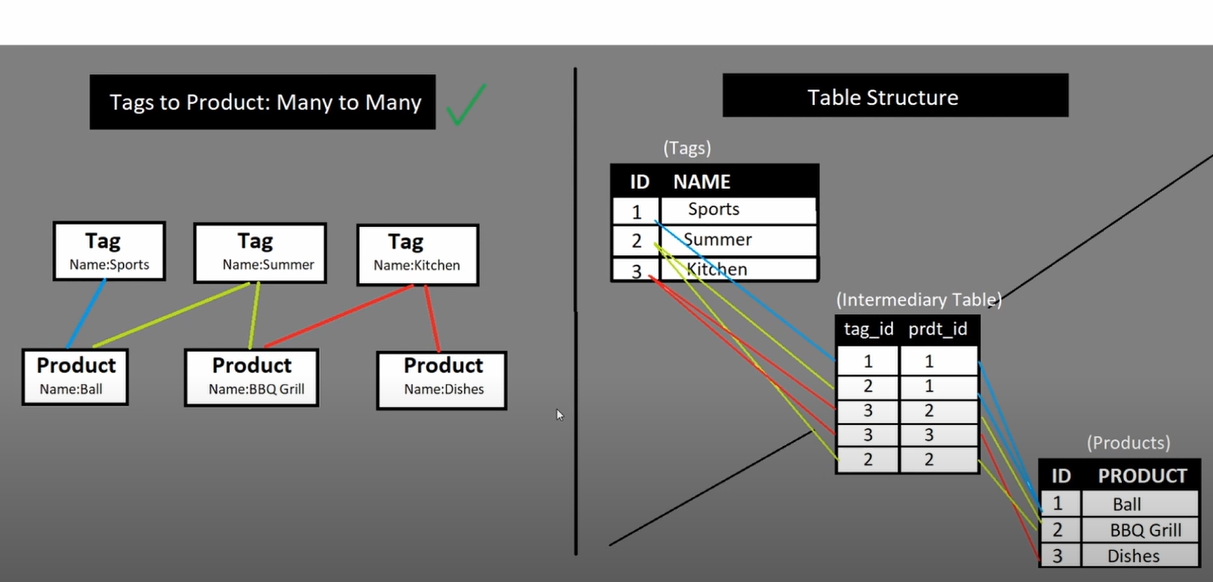
1) Explicación lógica de como representar una tabla one-to-many(uno a muchos)



Comparación de tabla uno a muchos, con una de muchos a muchos



Representacion lógica de una tabla many-to-many(muchos a muchos)



Ahora editamos nuestro modelo de order, agregando una clave foránea sobre la clave principal de costumer y clave foránea con productos

*class* order(*models*.*Model*):

    STATUS= (('Pending', 'Pending'),

            ('Out for Delivery', 'Out for Delivery'),

            ('Delivered', 'Delivered'))

    costumer= models.ForeignKey(costumer, *null*=True, *on\_delete*=models.SET\_NULL)

    product = models.ForeignKey(product, *null*=True, *on\_delete*=models.SET\_NULL)

    date\_created = models.DateField(*auto\_now\_add*=True)

    status=models.CharField(*max\_length*=100, *null*=True, *choices*=STATUS)

*class* Meta:

        db\_table = 'order'

Ahora mismo, creamos un modelo TAG. Esta tabla nos servirá para usar la representación many to many

Desde la tabla order

*class* order(*models*.*Model*):

    STATUS= (('Pending', 'Pending'),

            ('Out for Delivery', 'Out for Delivery'),

            ('Delivered', 'Delivered'))

    costumer= models.ForeignKey(costumer, *null*=True, *on\_delete*=models.SET\_NULL)

    product = models.ForeignKey(product, *null*=True, *on\_delete*=models.SET\_NULL)

    date\_created = models.DateField(*auto\_now\_add*=True)

    status=models.CharField(*max\_length*=100, *null*=True, *choices*=STATUS)

    tags=models.ManyToManyField(tag)

*class* Meta:

        db\_table = 'order'

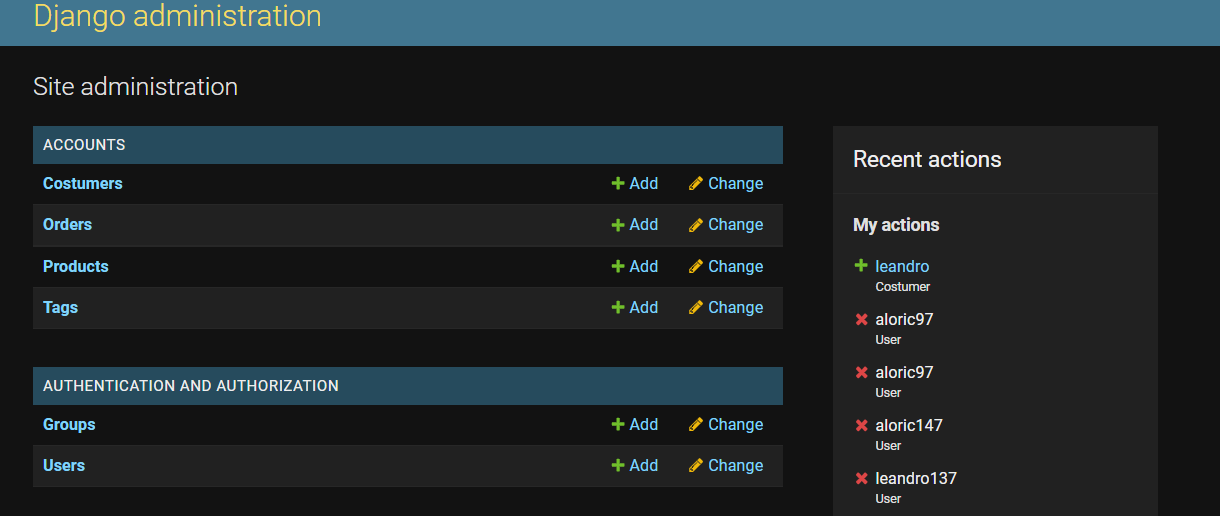
y registramos desde la admin.py

admin.site.register(tag)

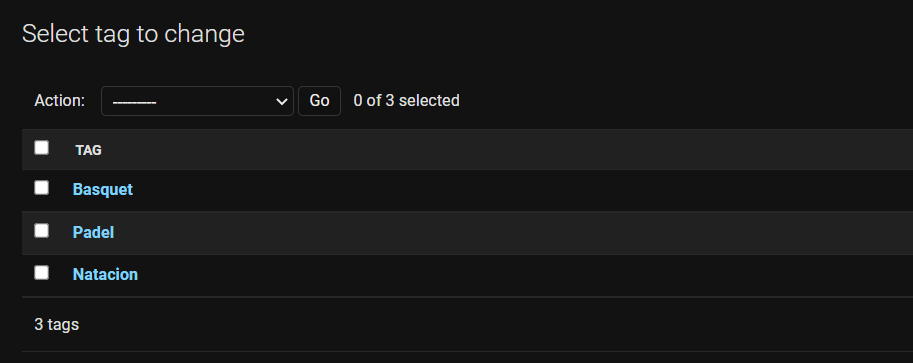
ahora mostramos las representación que incrustamos en nuestra bdd desde el panel admin.

Por ejemplo:

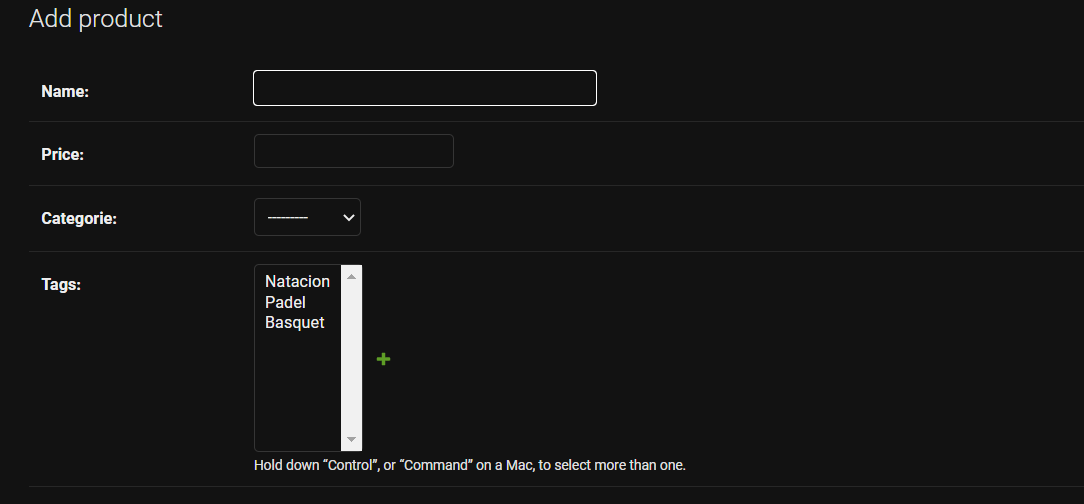
Nos vamos al panel



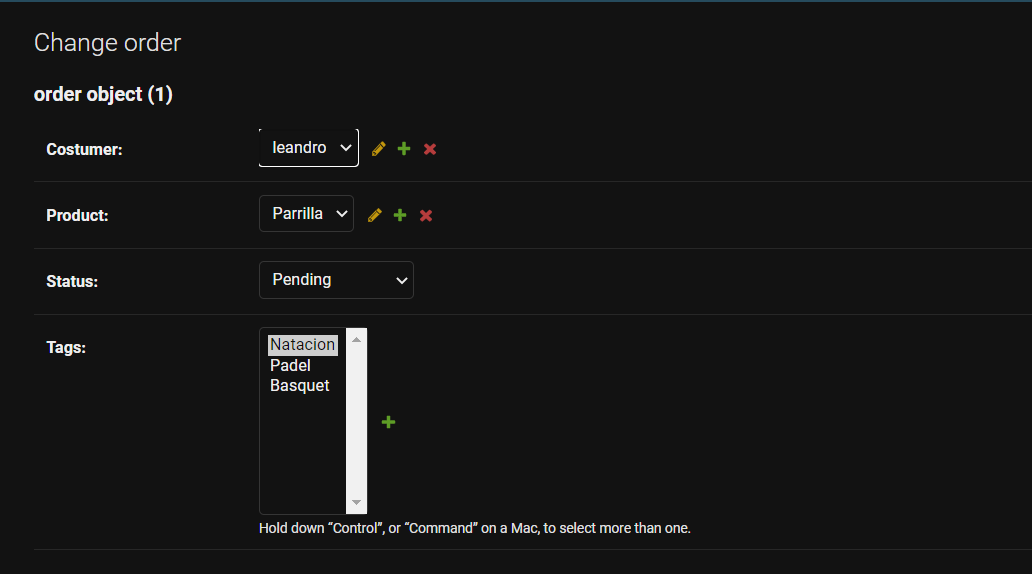
Desde ahí, nos dirigimos hacia tags y creamos objetos aleatorios, en este caso, cree 3:



Y cuando nos vamos a productos, nos mostrara los tags creados, debido a la representación many to many definida anteriormente



En la clase de order, como la definimos con dos claves foráneas, nos muestra la informacion de la siguiente manera:



Las claves foráneas representan tanto en costumer como en products