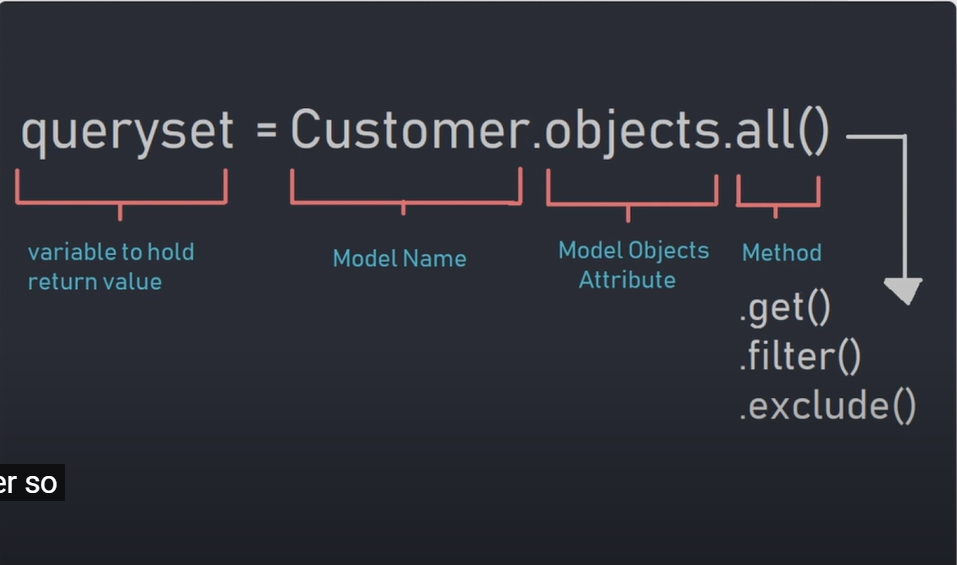
Para hacer una consulta a la base de datos, se utiliza este modelo



**CONSULTAS REALIZADAS**

*#\*\*\*(1)Returns all customers from customer table*

from .models import \*

customers = costumer.objects.all()

*#(2)Returns first customer in table*

firstCustomer = costumer.objects.first()

*#(3)Returns last customer in table*

lastCustomer = costumer.objects.last()

*#(4)Returns single customer by name*

customerByName = costumer.objects.get(*name*='Peter Piper')

*#\*\*\*(5)Returns single customer by name*

customerById = costumer.objects.get(*id*=4)

*#(7)\*\*\*Returns orders customer name: (Query parent model values)*

order = order.objects.first()

parentName = order.customer.name

*#(8)\*\*\*Returns products from products table with value of "Out Door" in category attribute*

products = product.objects.filter(*category*="Out Door")

*#(9)\*\*\*Order/Sort Objects by id*

leastToGreatest = product.objects.all().order\_by('id')

greatestToLeast = product.objects.all().order\_by('-id')

*# (10) Devuelve todos los productos con la etiqueta "Deportes": (Consultar muchos a muchos campos)*

productsFiltered = product.objects.filter(*tags\_\_name*="Sports")

'''

(11) Bono

P: Si el cliente tiene más de 1 bola, ¿cómo lo reflejaría en la base de datos?

R: Debido a que hay muchos productos diferentes y este valor cambia constantemente, lo

Probablemente no desee almacenar el valor en la base de datos, sino más bien hacer que esta sea una función que podamos ejecutar

cada vez que cargamos el perfil del cliente

'''

*# Devuelve el recuento total de la cantidad de veces que el primer cliente ordenó una "Bola"*

ballOrders = firstCustomer.order\_set.filter(*product\_\_name*="Ball").count()

*#Returns total count for each product orderd*

allOrders = {}

for order in firstCustomer.order\_set.all():

    if order.product.name in allOrders:

        allOrders[order.product.name] += 1

    else:

        allOrders[order.product.name] = 1

*#Returns: allOrders: {'Ball': 2, 'BBQ Grill': 1}*

*#RELATED SET EXAMPLE*

*class* ParentModel(*models*.*Model*):

    name = models.CharField(*max\_length*=200, *null*=True)

*class* ChildModel(*models*.*Model*):

    parent = models.ForeignKey(Customer)

    name = models.CharField(*max\_length*=200, *null*=True)

parent = ParentModel.objects.first()

*#Returns all child models related to parent*

parent.childmodel\_set.all()

Ahora bien, por ejemplo queremos reflejar esto en una pagina web, lo que hacemos es cargar una consulta y mostrar, en este caso lo hago con la tabla product:

*def* productos(*request*):

    context = product.objects.all()🡪consulta

    return render(request,'productos.html',{'context':context})

Luego, cargo para visualizarla en product.html

{% for i in products  %}

            <tr>

                <td>{{i.name}}</td>

                <td>{{i.category}}</td>

                <td>{{i.price}}</td>

            </tr>

            {% endfor %}