

[ ----- ]

Let's Understand

related\_name

In Django Models



# Introduction to related\_name

related\_name is an attribute used with OnetoOneField, ForeignKey and ManyToManyField fields. It is used to define reverse relationship with its parent models.

for eg., let's say we have a Address model and here we have foreign key relation with User model

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

class Address(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    address = models.TextField(max_length=500)
```

Whenever, you made any relationship with the parent model, django automatically create related\_name with lowercase <model\_name>\_set

```
user = models.ForeignKey(
    User,
    on_delete=models.CASCADE,
    related_name="address_set")
```

# How to use related\_name

Let's say, i have to get all the address of  
an logged in user

```
from django.shortcuts import render

def address(request):
    user_address = request.user.address_set.all()
```

request.user get the  
user object

we can simply use this related  
name, to fetch all the address  
of an particular user

# What if, we don't use related\_name

Let's suppose, if we don't use related\_name, then obviously, you just need to write an extra query

```
from django.shortcuts import render
from .models import Address

def address(request):
    user_address = Address.objects.select_related().filter(user=request.user)
```

Here, you can see we are using another query with Address model to fetch all the address of an user

# How to override the related\_name

As, we have discussed already django provide the default related\_name followed by lowercase <model\_name>\_set. However you can easily override the default related\_name

```
user = models.ForeignKey(  
    User,  
    on_delete=models.CASCADE,  
    related_name="user_address")
```



Simply we can override the related\_name like this

Now our query would be like this

```
def address(request):  
    user_address = request.user.user_address.all()
```

custom related\_name

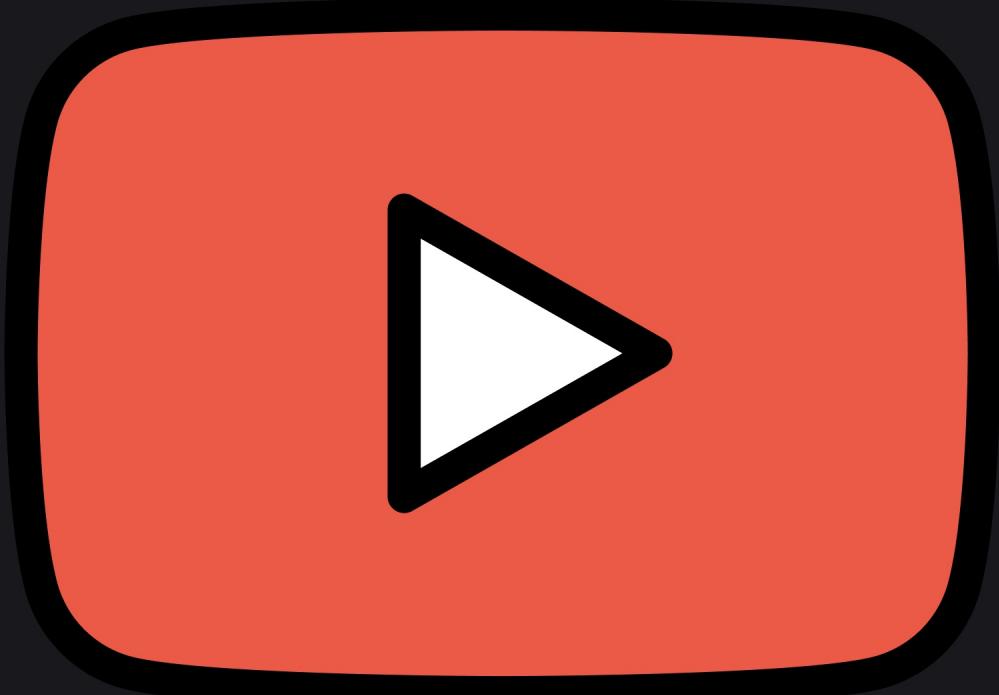
# Conclusion

related\_name attribute provide flexibility and readability when working with related objects and performing database queries in Django.

For more django  
related content

Subscribe

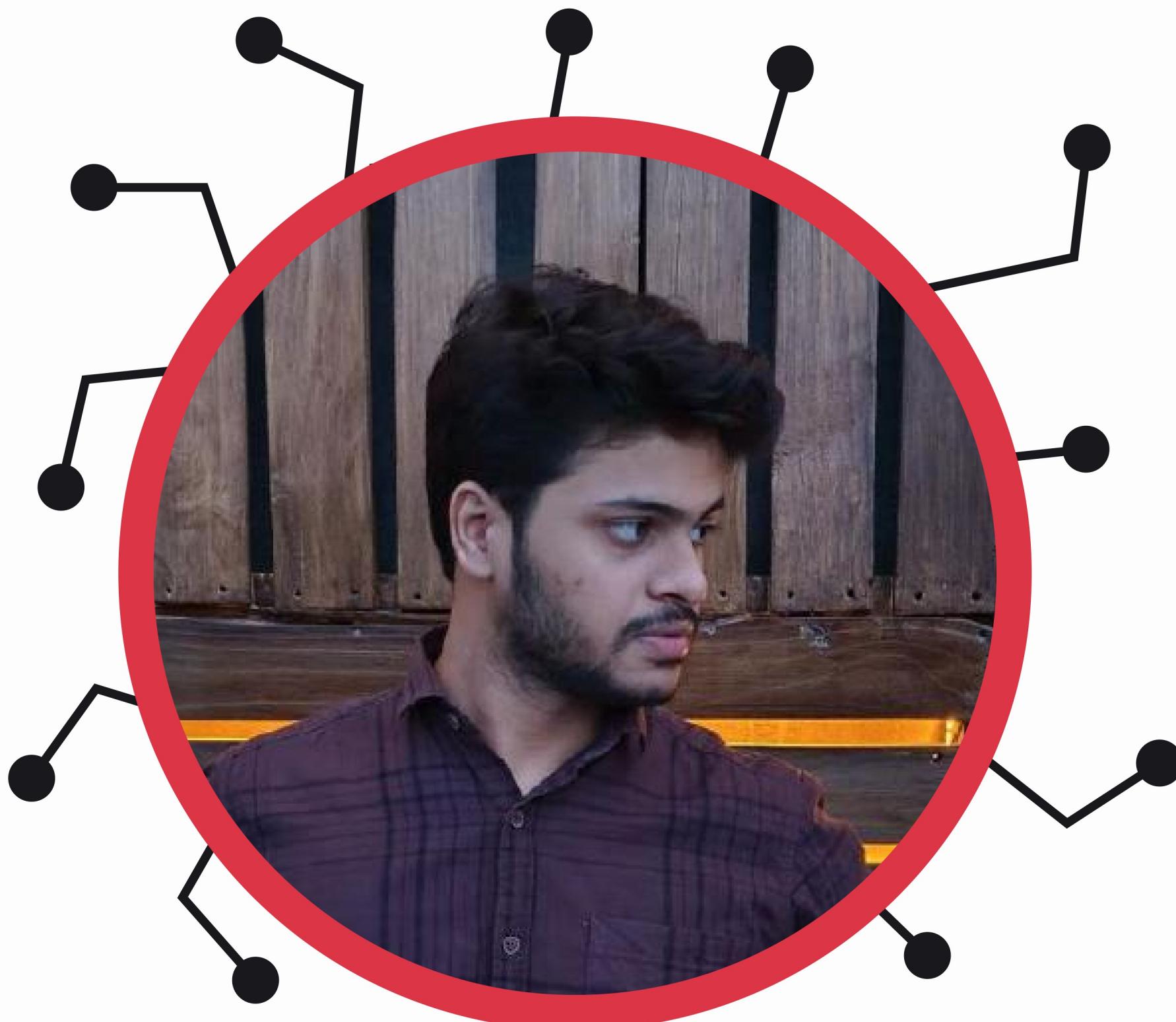
To My Youtube  
Channel



Link in bio

# DID YOU FIND THIS HELPFUL

Let me know in the comment



Aashish Kumar

Software Engineer

Follow for more ❤