**Videofile:75**

**Codefile:ch65**

**Django Model Many to One Relationship**

## Django Model Many-to-One Relationship

* A **Many-to-One relationship** means that **multiple objects of one model are related to a single object of another model**.
* In Django, this is created using **ForeignKey**.

### ✅ When to Use?

* When you want to allow **many items to belong to one category or parent object**.
* Example:
  + Many **Students** belong to one **Class**.
  + Many **Orders** belong to one **Customer**.

### 🔑 Key Points:

1. Defined using:
2. models.ForeignKey(OtherModel, on\_delete=models.CASCADE)
3. **Forward Access**: student.classroom → Access parent.
4. **Reverse Access**: classroom.student\_set.all() → Access all related children.
   * (You can change student\_set name using related\_name).
5. on\_delete defines behavior when the parent object is deleted (commonly CASCADE).

👉 Example for notes:

from django.db import models

class Classroom(models.Model):

    name = models.CharField(max\_length=100)

class Student(models.Model):

    name = models.CharField(max\_length=100)

    classroom = models.ForeignKey(Classroom, on\_delete=models.CASCADE, related\_name="students")

# Usage:

# c = Classroom.objects.create(name="10th Grade")

# s1 = Student.objects.create(name="Ali", classroom=c)

# s2 = Student.objects.create(name="Ayesha", classroom=c)

# Forward:

# print(s1.classroom.name)   # Ali's classroom

# Reverse:

# print(c.students.all())    # All students in this classroom

⚡ **Summary for Notes:**  
A Many-to-One relationship in Django is defined with ForeignKey, allowing multiple child records to be linked to one parent record. Example: Many Students can belong to one Classroom.

Self document:

Models.py:

from django.db import models

from django.contrib.auth.models import User

# Create your models here.

class Post(models.Model):

    user = models.ForeignKey(User,on\_delete=models.CASCADE)

    title = models.CharField(max\_length=255)

By using cascade:

in this we can make relation between the user and the post :

1. user can make mulitple post in this
2. if we want to delete a single user so all their posts automatically delete
3. if we can delete the a single post or others. But user exists user cannot be delete

By using protected:

from django.db import models

from django.contrib.auth.models import User

# Create your models here.

class Post(models.Model):

    user = models.ForeignKey(User,on\_delete=models.PROTECT)

    title = models.CharField(max\_length=255)

* BY USING PROTECT ,
* IF WE WANT TO DELET THE USER, IF USER CAN CREATE THE POST SO USER CANOT BE DELETE UNTIL THE POST DLETE’S
* IF POST CANNOT EXIST OF THAT USER SO USER CAN EASILY BE DELETED

If we want both user or post can do independently so we use this :

    user = models.ForeignKey(User,on\_delete=models.SET\_NULL,null=True)

by using this, if we delete user but their post exist so make it null