from django.db import models

# Create your models here.

class Person(models.Model):

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

email = models.EmailField()

mobile = models.TextField(max\_length=100)

class Adhar(models.Model):

person = models.OneToOneField("Person",on\_delete=models.CASCADE)

signature = models.TextField()

adhar\_no = models.TextField(max\_length=100)

from adhar.models import Person

p1 = Person(name="Joan louji", email="sjlouji10@gmail.com", mobile="123412")

p1.save()

p2 = Person(name="Rion", email="rion@gmail.com", mobile="0987632")

p2.save()

from adhar.models import Adhar

a1 = Adhar(person=p1, signature="asdf234dsafuiq3&^$^GUJHVR", adhar\_no="432198176354")

a1.save()

a2 = Adhar(person=p2, signature="adfpoiwruywqtrgmnb13241)\*&^%$", adhar\_no="2314928376")

a2.save()

p = Person.objects.all()

print(p)

<QuerySet [<Person: Person object (1)>, <Person: Person object (2)>]>

a = Adhar.objects.all()

print(a)

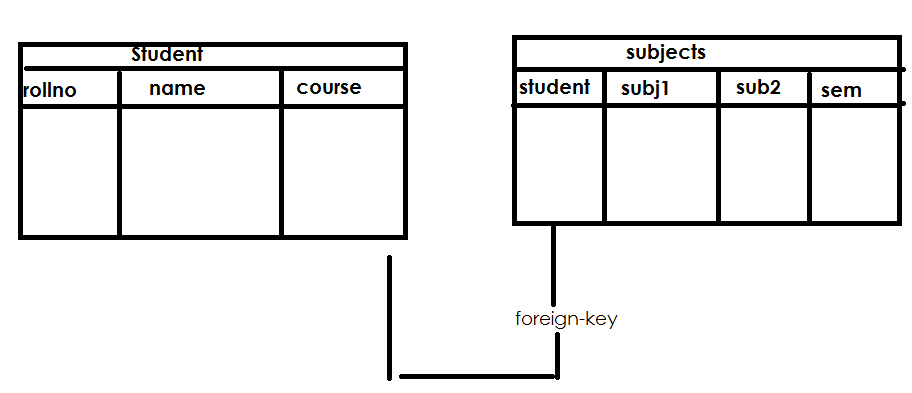
<QuerySet [<Adhar: Adhar object (1)>, <Adhar: Adhar object (2)>]>

a.values()

QuerySet [{'id': 1, 'person\_id': 1, 'signature': 'asdf234dsafuiq3&^$^GUJHVR', 'adhar\_no': '432198176354'}, {'id': 2, 'person\_id': 2, 'signature': 'adfpoiwruywqtrgmnb13241)\*&^%$', 'adhar\_no': '2314928376'}]>

**One-Many or Many -One**

One-Many relationship is done using one-many-field or foreign-key



**Models.py**

class Student(models.Model):

     rollno=models.CharField(max\_length=4)

     name=models.CharField(max\_length=15)

class Marks(models.Model):

     student=models.ForeignKey("Student",on\_delete=models.CASCADE)

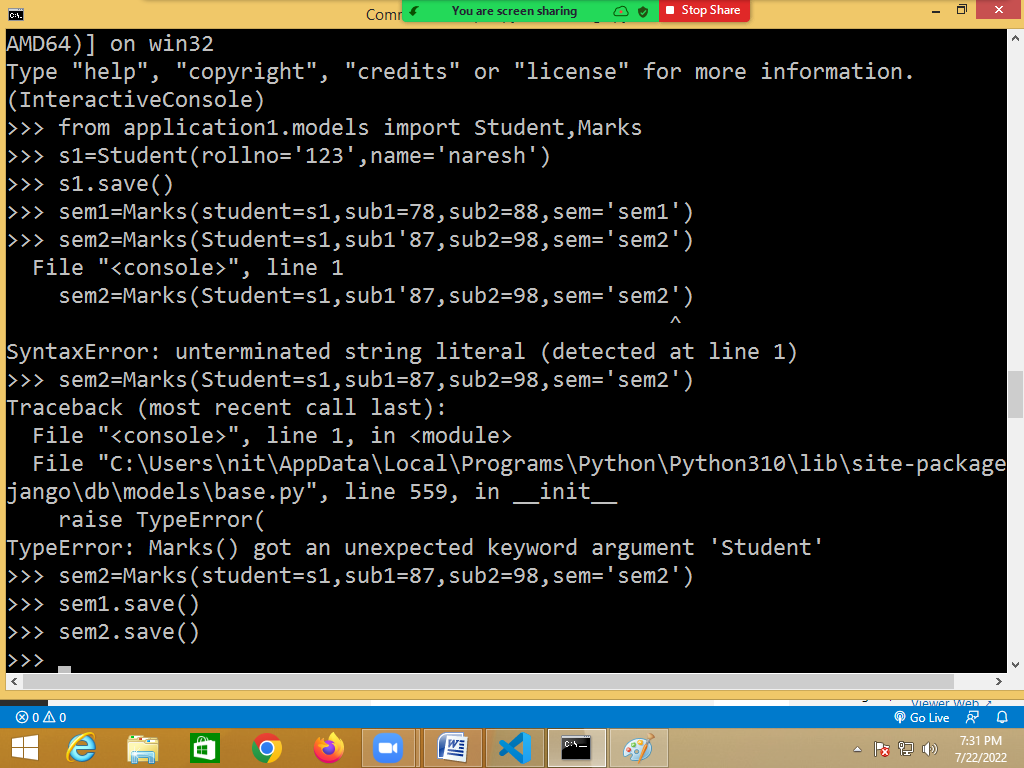
     sub1=models.IntegerField()

     sub2=models.IntegerField()

     sem=models.CharField(max\_length=10)

python manage.py makemigrations

python manage.py migrate



**Many-to-many fields:**

This is used when one record of a model A is related to multiple records of another model B and vice versa. For example – a model **Book** has many-to-many relationship with a model **Author**, i.e. an book can be written by multiple authors and an author can write multiple books. Many-to-many relations are defined using **ManyToManyField** field of **django.db.models**.

from django.db import models

class Author(models.Model):

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

desc = models.TextField(max\_length = 300)

class Book(models.Model):

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

desc = models.TextField(max\_length = 300)

authors = models.ManyToManyField(Author)