

## One-to-One and Many-to-One relationship

1. **Create Virtual Environment**
  - `python -m venv env`
2. **Activate Virtual Environment**
  - `env\Scripts\activat`
3. **Install Django**
  - `pip install Django`
4. **Create Django Project**
  - `django-admin startproject project`
  - `cd project`
5. **Create App**
  - `python manage.py startapp studentapp`
6. Add app in **settings.py**

```
# project/settings.py
INSTALLED_APPS = [
    ...
    ...
    'app',
]
```

### 7. Source Code of app/models.py (Copied Text for Easy Reference)

Below is the same code shown in the screenshot.

You can copy and paste it directly in your models.py file.

Code >>>>>>

```
from django.db import models
```

```
# Create your models here.
```

```
# This example demonstrates a one-to-one relationship between Student and Adhar models.
```

```
class Adhar(models.Model):
    adhar = models.IntegerField(unique=True)
```

```
class Student(models.Model):
    name = models.CharField(max_length=50)
    email = models.EmailField(unique=True)
    city = models.CharField(max_length=50)
    adhar = models.OneToOneField(Adhar, on_delete=models.PROTECT, related_name='stu_info')
```

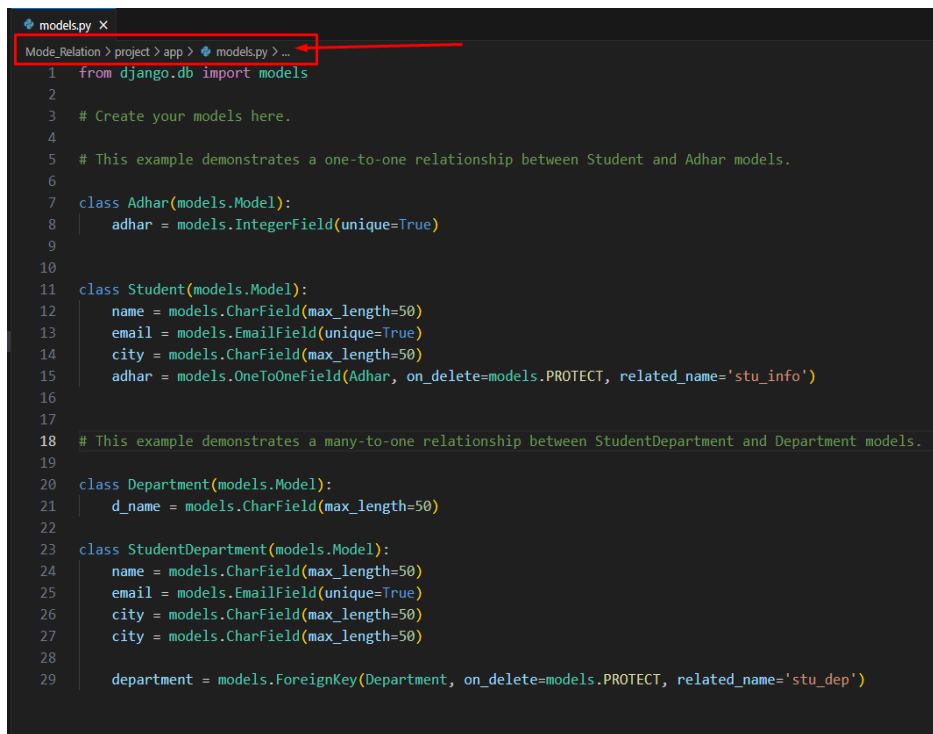
```
# This example demonstrates a many-to-one relationship between StudentDepartment and Department models.
```

```
class Department(models.Model):
    d_name = models.CharField(max_length=50)
```

```
class StudentDepartment(models.Model):
    name = models.CharField(max_length=50)
    email = models.EmailField(unique=True)
```

```
city = models.CharField(max_length=50)
city = models.CharField(max_length=50)
```

```
department = models.ForeignKey(Department, on_delete=models.PROTECT, related_name='stu_dep')
```



```
models.py
1 from django.db import models
2
3 # Create your models here.
4
5 # This example demonstrates a one-to-one relationship between Student and Adhar models.
6
7 class Adhar(models.Model):
8     adhar = models.IntegerField(unique=True)
9
10
11 class Student(models.Model):
12     name = models.CharField(max_length=50)
13     email = models.EmailField(unique=True)
14     city = models.CharField(max_length=50)
15     adhar = models.OneToOneField(Adhar, on_delete=models.PROTECT, related_name='stu_info')
16
17
18 # This example demonstrates a many-to-one relationship between StudentDepartment and Department models.
19
20 class Department(models.Model):
21     d_name = models.CharField(max_length=50)
22
23 class StudentDepartment(models.Model):
24     name = models.CharField(max_length=50)
25     email = models.EmailField(unique=True)
26     city = models.CharField(max_length=50)
27     city = models.CharField(max_length=50)
28
29     department = models.ForeignKey(Department, on_delete=models.PROTECT, related_name='stu_dep')
```

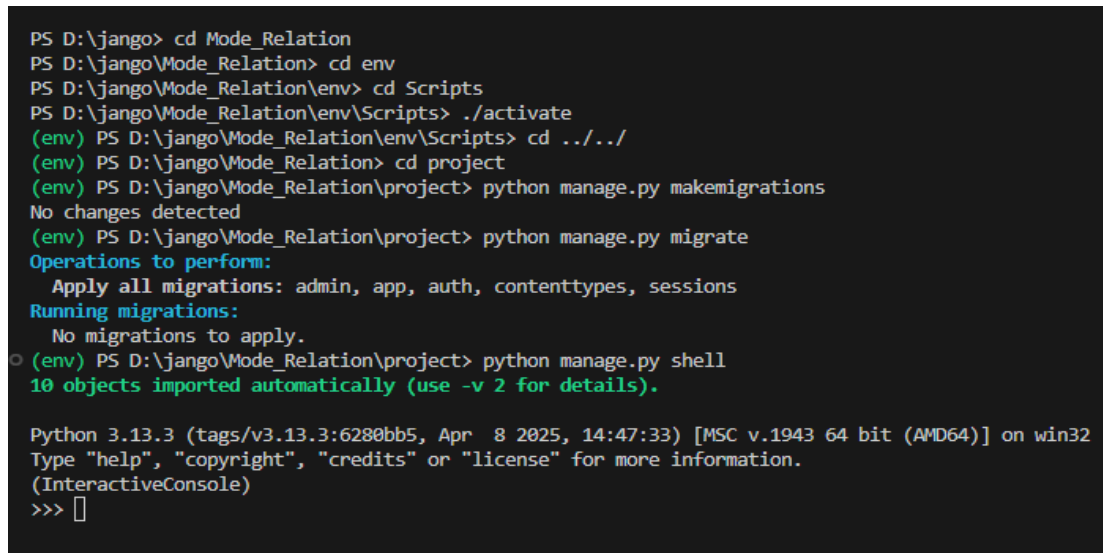
## 8. Database Migration Commands (After Writing models.py)

- python manage.py makemigrations
- python manage.py migrate

### Next Steps: Insert Data into Tables Using Django Shell

Shell open karne ke liye command:

- python manage.py shell
- commands till shell in below image

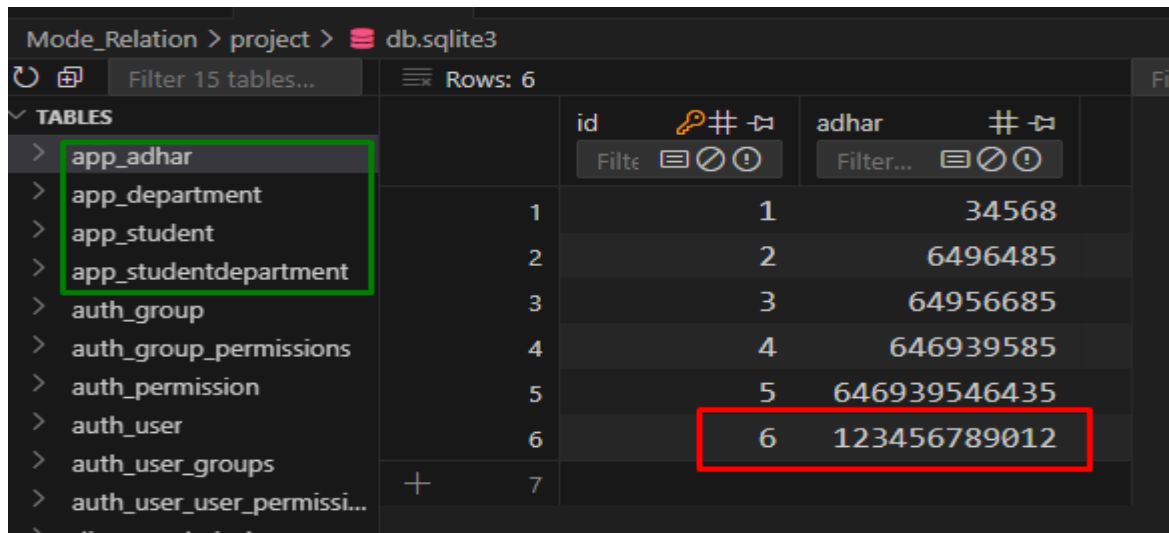


```
PS D:\jango> cd Mode_Relation
PS D:\jango\Mode_Relation> cd env
PS D:\jango\Mode_Relation\env> cd Scripts
PS D:\jango\Mode_Relation\env\Scripts> ./activate
(env) PS D:\jango\Mode_Relation\env\Scripts> cd ../../
(env) PS D:\jango\Mode_Relation> cd project
(env) PS D:\jango\Mode_Relation\project> python manage.py makemigrations
No changes detected
(env) PS D:\jango\Mode_Relation\project> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, app, auth, contenttypes, sessions
Running migrations:
  No migrations to apply.
(env) PS D:\jango\Mode_Relation\project> python manage.py shell
10 objects imported automatically (use -v 2 for details).

Python 3.13.3 (tags/v3.13.3:6280bb5, Apr 8 2025, 14:47:33) [MSC v.1943 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> []
```

## 1. One-to-One Relationship: Student & Adhar

- from app.models import Student, Adhar
- # Adhar object create karo
- a1 = Adhar.objects.create(adhar=123456789012)

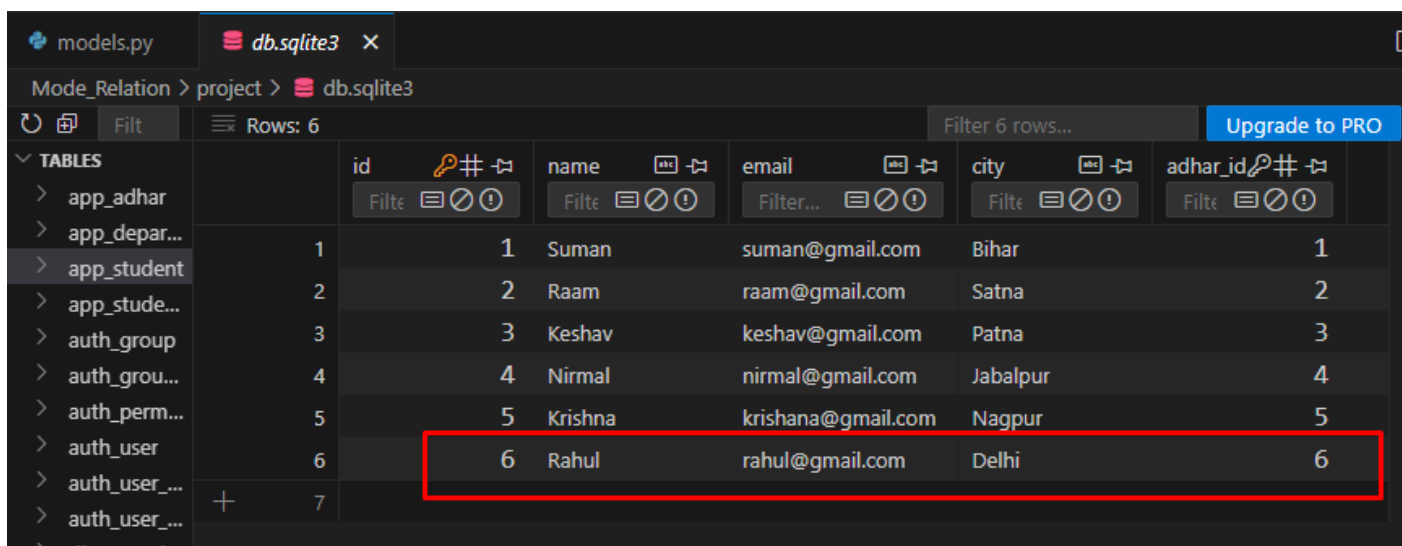


Mode\_Relation > project > db.sqlite3

Filter 15 tables... Rows: 6

TABLES		id	adhar
> app_adhar		1	34568
> app_department		2	6496485
> app_student		3	64956685
> app_studentdepartment		4	646939585
> auth_group		5	646939546435
> auth_group_permissions		6	123456789012
> auth_permission		7	
> auth_user			
> auth_user_groups			
> auth_user_user_permissions			

- # Student object create karo aur Adhar assign karo
- s1 = Student.objects.create(name="Rahul", email="rahul@gmail.com", city="Delhi", adhar=a1)



models.py db.sqlite3

Mode\_Relation > project > db.sqlite3

Filter Rows: 6 Filter 6 rows... Upgrade to PRO

TABLES		id	name	email	city	adhar_id
> app_adhar		1	Suman	suman@gmail.com	Bihar	1
> app_depar...		2	Raam	raam@gmail.com	Satna	2
> app_student		3	Keshav	keshav@gmail.com	Patna	3
> app_stude...		4	Nirmal	nirmal@gmail.com	Jabalpur	4
> auth_group		5	Krishna	krishana@gmail.com	Nagpur	5
> auth_grou...		6	Rahul	rahul@gmail.com	Delhi	6
> auth_perm...		7				
> auth_user						
> auth_user_...						
> auth_user_...						

```
No migrations to apply.
(env) PS D:\jango\Mode_Relation\project> python manage.py shell
10 objects imported automatically (use -v 2 for details).

Python 3.13.3 (tags/v3.13.3:6280bb5, Apr 8 2025, 14:47:33) [MSC v.1943 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from app.models import Student, Adhar
>>> a1 = Adhar.objects.create(adhar=123456789012)
>>> s1 = Student.objects.create(name="Rahul", email="rahul@gmail.com", city="Delhi", adhar=a1)
>>>
```

- # Access One-to-One data
- `print(s1.adhar.adhar)`
- `print(a1.stu_info.name)` # using `related_name`

```
>>> from app.models import Student, Adhar
>>> a1 = Adhar.objects.create(adhar=123456789012)
>>> s1 = Student.objects.create(name="Rahul", email="rahul@gmail.com")
>>> print(s1.adhar.adhar)
123456789012
>>> print(a1.stu_info.name)
Rahul
>>> print(a1.stu_info.city)
Delhi
>>> []
```

## 2. Many-to-One Relationship: StudentDepartment & Department

- `from app.models import Department, StudentDepartment`
- # Department create karo
- `d1 = Department.objects.create(d_name="Computer Science")`

```
(InteractiveConsole)
>>> from app.models import Student, Adhar
>>> a1 = Adhar.objects.create(adhar=123456789012)
>>> s1 = Student.objects.create(name="Rahul", email="rahul@gmail.com")
>>> print(s1.adhar.adhar)
123456789012
>>> print(a1.stu_info.name)
Rahul
>>> print(a1.stu_info.city)
Delhi
>>> from app.models import Department, StudentDepartment
>>> d1 = Department.objects.create(d_name="Computer Science")
>>> []
```

Refresh the data and check it

Mode\_Relation > project > db.sqlite3

Filter Rows: 5

TABLES		id	d_name
> app_adhar			
> app_depar...			
> app_student	1	1	CSE
> app_stude...	2	2	ME
> auth_group	3	3	EE
> auth_grou...	4	4	IT
> auth_perm...	5	5	Computer Science
> auth_user	+	6	
> auth_user_...			

- # StudentDepartment create karo aur department assign karo
- `s2 = StudentDepartment.objects.create(name="Aman", email="aman@gmail.com", city="Lucknow", department=d1)`
- `s3 = StudentDepartment.objects.create(name="Nikki", email="nikki@gmail.com", city="Kanpur", department=d1)`

```
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from app.models import Student, Adhar
>>> a1 = Adhar.objects.create(adhar=123456789012)
>>> s1 = Student.objects.create(name="Rahul", email="rahul@gmail.com", city="Delhi", adhar=a1)
>>> print(s1.adhar.adhar)
123456789012
>>> print(a1.stu_info.name)
Rahul
>>> print(a1.stu_info.city)
Delhi
>>> from app.models import Department, StudentDepartment
>>> d1 = Department.objects.create(d_name="Computer Science")
>>> s2 = StudentDepartment.objects.create(name="Aman", email="aman@gmail.com", city="Lucknow", department=d1)
>>> s3 = StudentDepartment.objects.create(name="Nikki", email="nikki@gmail.com", city="Kanpur", department=d1)
>>> []
```

Refresh the studentdepartment data and check it

Mode\_Relation > project > db.sqlite3

Filter 15 Rows: 7 Filter 7 rows... Upgrade to PRO

TABLES			name	email	city	depart...
> app_adhar			Filter...	Filter...	Filter...	Filter...
> app_department						
> app_student	1	1	Ramesh	ramesh@gmail.com	Bhopal	1
> app_studentdep...	2	2	Kumar	kumar@gmail.com	Bhopal	2
> auth_group	3	3	Vicky	vicky@gmail.com	Sarna	3
> auth_group_per...	4	4	Suryabhan	suryabhan@gmail.com	Sarna	3
> auth_permission	5	5	Raam	ram@gmail.com	Jabalpur	2
> auth_user	6	6	Aman	aman@gmail.com	Lucknow	5
> auth_user_groups	7	7	Nikki	nikki@gmail.com	Kanpur	5
> auth_user_user_...	+	8				
> django_admin_l...						
> django_content_...						

- # Access Many-to-One data
- `print(s2.department.d_name)`

```
Delhi
>>> from app.models import Department, StudentDepartment
>>> d1 = Department.objects.create(d_name="Computer Science")
>>> s2 = StudentDepartment.objects.create(name="Aman", email="aman@gmail.com", city="Lucknow", department=d1)
>>> s3 = StudentDepartment.objects.create(name="Nikki", email="nikki@gmail.com", city="Kanpur", department=d1)
>>> print(s2.department.d_name)
Computer Science
```