

Working with multiple models / Relationships

Working with multiple models / Relationships

Faisal Memon

Kinds of relationships

- One to one relationship
- Many to One relationship
- Many to Many relationship

One to One Relationship

One to one relationship is a type of relationship where in one record in a table is related to exactly one more record in another table and vice versa.

Example

Employee and salary account



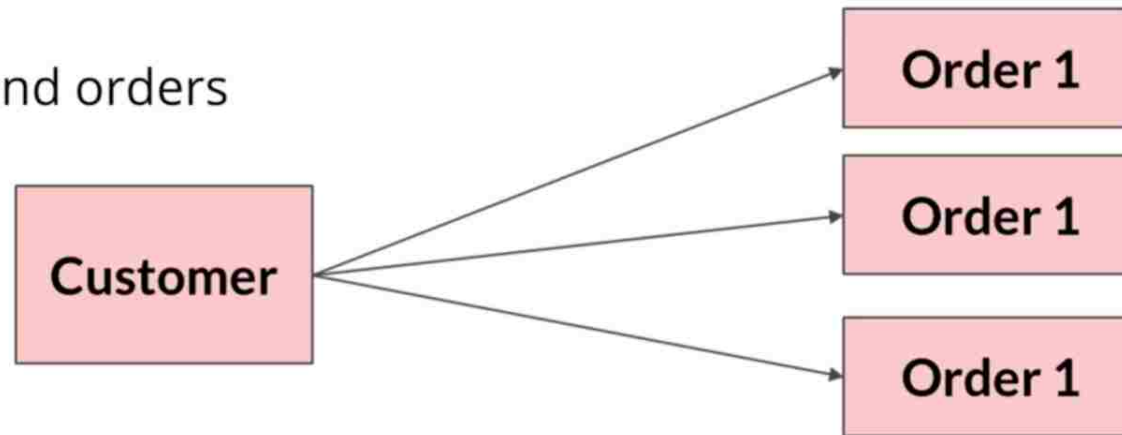
Example 2 → Person and passport

Many to One relationship

A many to one relationship is a type of relationship when one record in one table has one or many related record in another table.

Example

Customer and orders

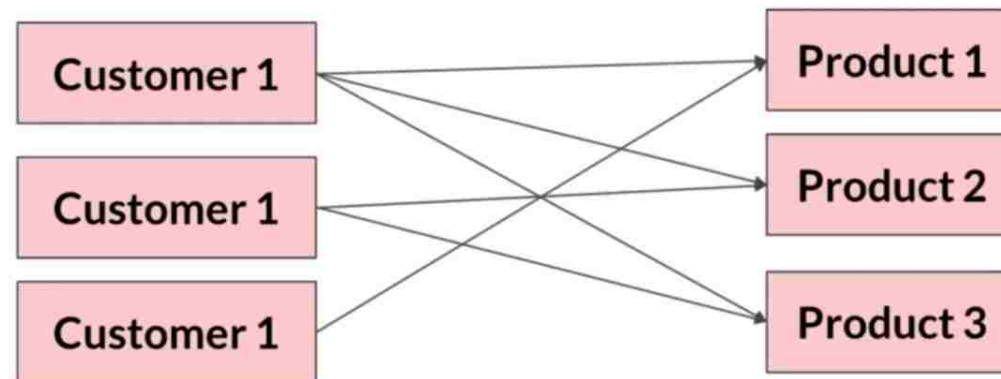


Many to Many relationship

A many to many relationship is a type of relationship when one record in one table has many related record in another table.

Example

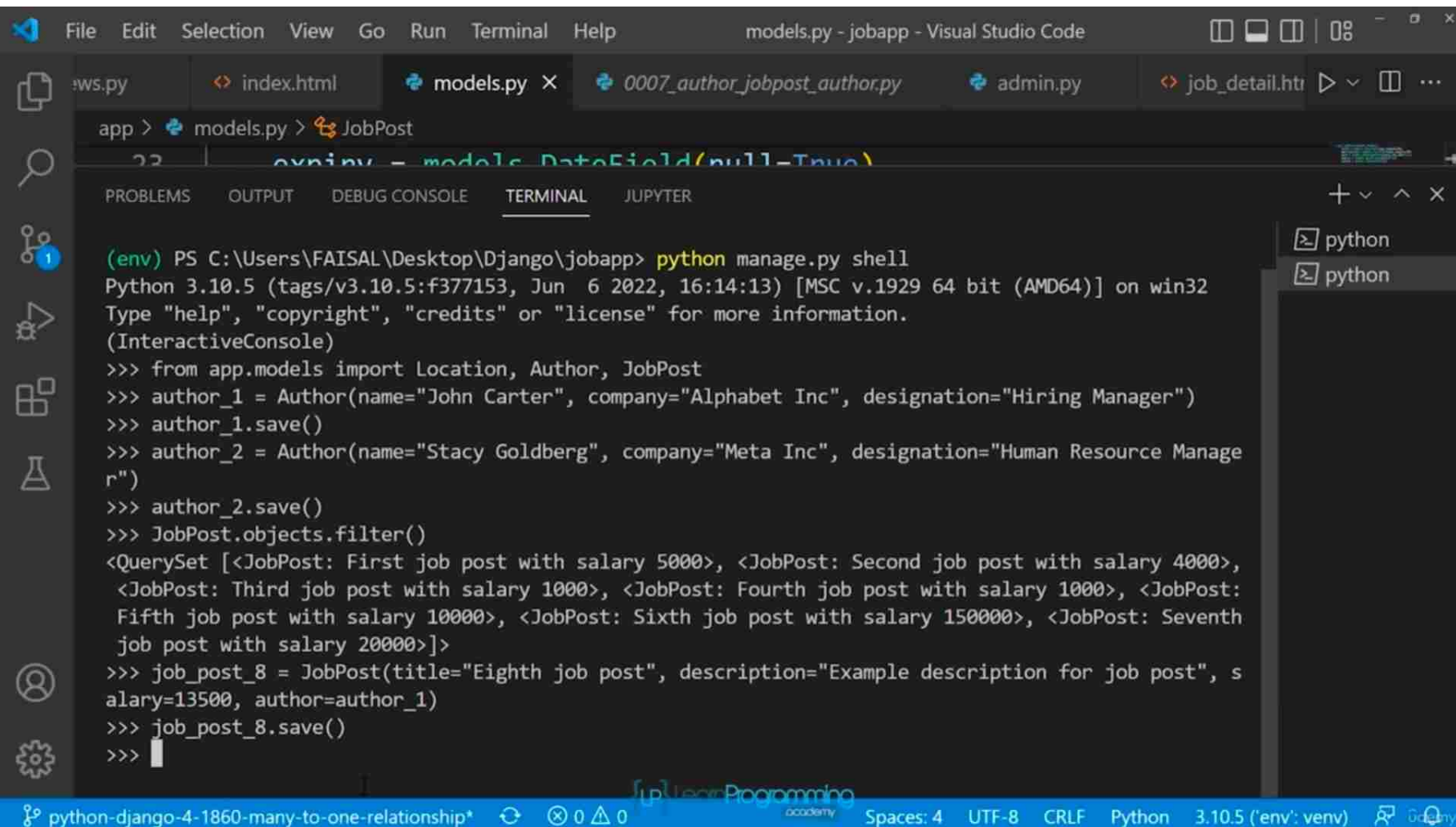
Customer and products




```
File Edit Selection View Go Run Terminal Help 0006_location_jobpost_location.py - jobapp - Visual Studio C...  
models.py 0006_location_jobpost_location.py U x admin.py job  
app > migrations > 0006_location_jobpost_location.py > Migration  
22 ('zip', models.CharField(max_length=200)),  
23 ],  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER  
Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license" for more information.  
(InteractiveConsole)  
>>> from app.models import JobPost, Location  
>>> l1 = Location(street="abc", city="adelaide", state="california", zip="42780")  
>>> l1.save()  
>>> job_post_1=JobPost.objects.all()[0]  
>>> job_post_1  
<JobPost: First job post with salary 5000>  
>>> job_post_1.location  
>>> job_post_1.location=l1  
>>> job_post_1.location  
<Location: Location object (1)>  
>>> job_post_1.save()  
>>> job_post_2=JobPost.objects.all()[1]  
>>> job_post_2  
<JobPost: Second job post with salary 4000>  
>>> job_post_2.location  
>>> job_post_2.location=
```

python-django-4-1740-using-inbuilt-css-classes Spaces: 4 UTF-8 CRLF Python 3.10.5 ('env': venv)

```
File Edit Selection View Go Run Terminal Help 0006_location_jobpost_location.py - jobapp - Visual Studio C...  
models.py 0006_location_jobpost_location.py U x admin.py job  
app > migrations > 0006_location_jobpost_location.py > Migration  
22 ('zip', models.CharField(max_length=200)),  
23 ],  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER  
>>> job_post_2.location=12  
>>> job_post_2.save()  
>>> JobPost.objects.get(title="Second job post")  
<JobPost: Second job post with salary 4000>  
>>> JobPost.objects.get(location=11)  
<JobPost: First job post with salary 5000>  
>>> JobPost.objects.get(location__street="abc")  
<JobPost: First job post with salary 5000>  
>>> JobPost.objects.get(location__street__startswith="a")  
<JobPost: First job post with salary 5000>  
>>> Location.objects.get(jobpost=job_post_1)  
<Location: Location object (1)>  
>>> Location.objects.get(jobpost__location=11)  
<Location: Location object (1)>  
>>> Location.objects.get(jobpost=job_post_2)  
<Location: Location object (2)>  
>>> Location.objects.get(jobpost__title__startswith="First")  
<Location: Location object (1)>  
>>> Location.objects.get(jobpost__location__title__startswith="First")  
python-django-4-1740-using-inbuilt-css-classes 0 0 0 Spaces: 4 UTF-8 CRLF Python 3.10.5 (env: venv)
```



```
models.py - jobapp - Visual Studio Code

models.py x 0007_author_jobpost_author.py admin.py job_detail.ht

app > models.py > JobPost
... = models.DateField(null=True)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

(env) PS C:\Users\FAISAL\Desktop\Django\jobapp> python manage.py shell
Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from app.models import Location, Author, JobPost
>>> author_1 = Author(name="John Carter", company="Alphabet Inc", designation="Hiring Manager")
>>> author_1.save()
>>> author_2 = Author(name="Stacy Goldberg", company="Meta Inc", designation="Human Resource Manage
r")
>>> author_2.save()
>>> JobPost.objects.filter()
<QuerySet [<JobPost: First job post with salary 5000>, <JobPost: Second job post with salary 4000>,
<JobPost: Third job post with salary 1000>, <JobPost: Fourth job post with salary 1000>, <JobPost:
Fifth job post with salary 10000>, <JobPost: Sixth job post with salary 150000>, <JobPost: Seventh
job post with salary 20000>]>
>>> job_post_8 = JobPost(title="Eighth job post", description="Example description for job post", s
alary=13500, author=author_1)
>>> job_post_8.save()
>>>
```

python-django-4-1860-many-to-one-relationship* Spaces: 4 UTF-8 CRLF Python 3.10.5 (env: venv)

```
File Edit Selection View Go Run Terminal Help models.py - jobapp - Visual Studio Code

models.py x 0007_author_jobpost_author.py admin.py job_detail.hti

app > models.py > JobPost
22 | ... = models.DateField(null=True)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

<QuerySet [<JobPost: Sixth job post with salary 150000>, <JobPost: Eighth job post with salary 13500>]>
>>> Author.objects.get(name__contains="John")
<Author: Author object (1)>
>>> Author.objects.get(name__contains="John").jobpost.set_all()
Traceback (most recent call last):
  File "<console>", line 1, in <module>
AttributeError: 'Author' object has no attribute 'jobpost'
>>> Author.objects.get(name__contains="John").jobpost_set.all()
<QuerySet [<JobPost: Sixth job post with salary 150000>, <JobPost: Eighth job post with salary 13500>]>
>>> job_post_9 = author_1.jobpost_set.create(title="Nineth job post",description="Example testing", salary=2500)
>>> job_post_9
<JobPost: Nineth job post with salary 2500>
>>> job_post_1=JobPost.objects.get(title__contains="First")
>>> job_post_1
<JobPost: First job post with salary 5000>
>>> author_1.jobpost_set.add(job_post_1)
>>> JobPost.objects
```

python-django-4-1860-many-to-one-relationship Spaces: 4 UTF-8 CRLF Python 3.10.5 (env: venv)

models.py - jobapp - Visual Studio Code

File Edit Selection View Go Run Terminal Help

models.py x 0007_author_jobpost_author.py admin.py job_detail.ht

app > models.py > JobPost

```
22 class JobPost(models.Model):
    salary = models.IntegerField()
    author = models.ForeignKey(Author, on_delete=models.CASCADE)
    title = models.CharField(max_length=100)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return f'JobPost: {self.title} with salary {self.salary}'

    class Meta:
        ordering = ('salary',)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
<QuerySet [(<JobPost: First job post with salary 5000>, <JobPost: Sixth job post with salary 150000>,
<JobPost: Eighth job post with salary 13500>, <JobPost: Nineth job post with salary 2500>)]>
>>> JobPost.objects.filter(author__in=[1,2])
<QuerySet [(<JobPost: First job post with salary 5000>, <JobPost: Sixth job post with salary 150000>,
<JobPost: Eighth job post with salary 13500>, <JobPost: Nineth job post with salary 2500>)]>
>>> Author.objects.filter(jobpost__id=1)
<QuerySet [(<Author: Author object (1)>)]>
>>> Author.objects.filter(jobpost=1)
<QuerySet [(<Author: Author object (1)>)]>
>>> Author.objects.filter(jobpost__pk=1)
<QuerySet [(<Author: Author object (1)>)]>
>>> Author.objects.filter(jobpost=job_post_1)
<QuerySet [(<Author: Author object (1)>)]>
>>> Author.objects.filter(jobpost__title="First job post")
<QuerySet [(<Author: Author object (1)>)]>
>>> Author.objects.filter(jobpost__title__contains="First")
<QuerySet [(<Author: Author object (1)>)]>
>>> Author.objects.filter(jobpost__title__contains="First").count()
1
>>>
```

python python

python-django-4-1860-many-to-one-relationship* 0 0 0 Spaces: 4 UTF-8 CRLF Python 3.10.5 ('env': venv)

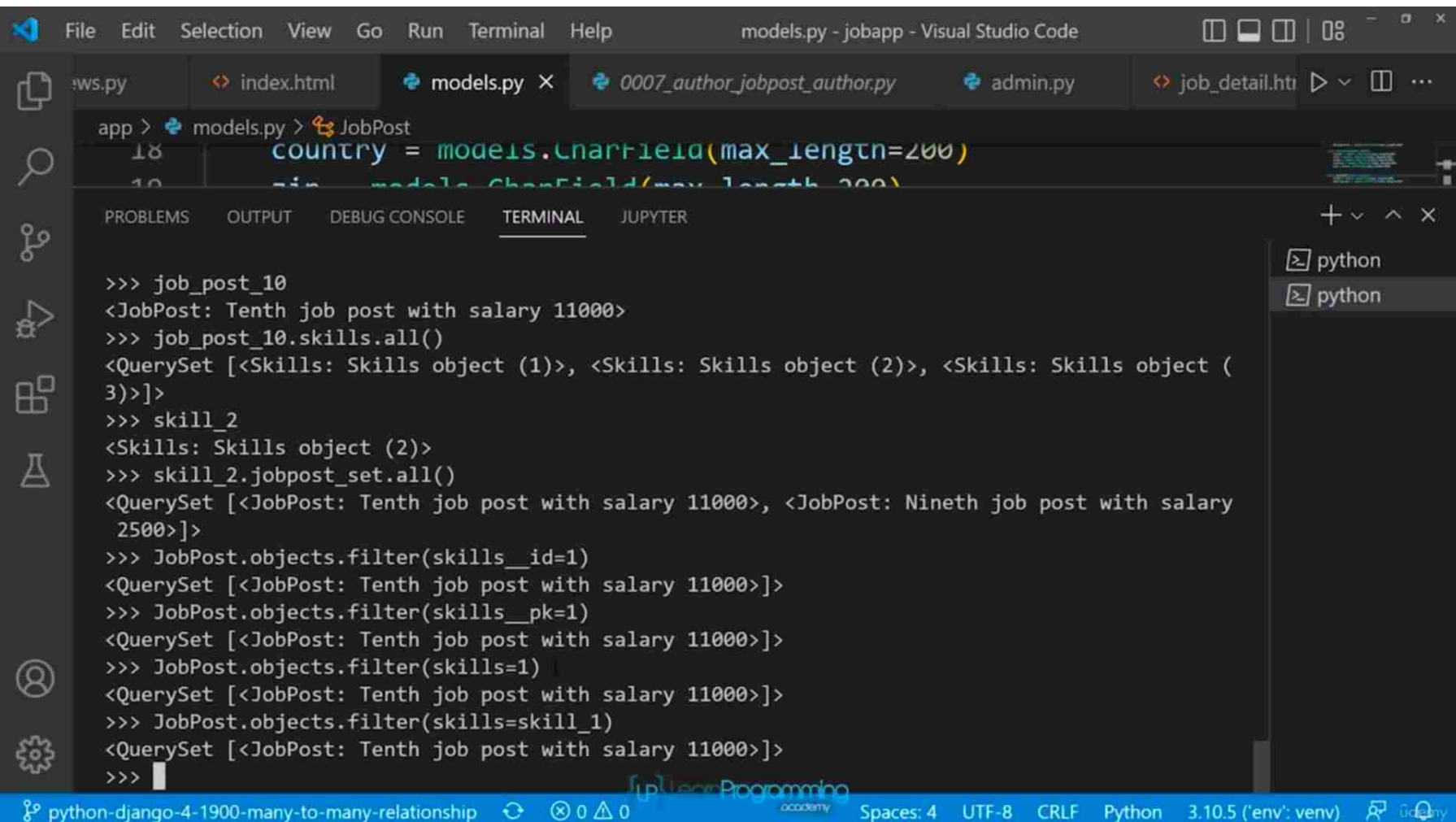

```
File Edit Selection View Go Run Terminal Help models.py - jobapp - Visual Studio Code

models.py M X 0007_author_jobpost_author.py admin.py job_d

app > models.py > JobPost
26 expiry = models.DateField(null=True)
27 salary = models.IntegerField()
28 slug = models.SlugField(null=True, max length=40, unique=True)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Python 3.10.5 (tags/v3.10.5:f377153, Jun 6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from app.models import JobPost, Location, Skills, Author
>>> all_jobs = JobPost.objects.all()
>>> all_jobs
<QuerySet [<JobPost: First job post with salary 5000>, <JobPost: Second job post with salary 400
<JobPost: Third job post with salary 1000>, <JobPost: Fourth job post with salary 1000>, <JobPo
Fifth job post with salary 10000>, <JobPost: Sixth job post with salary 150000>, <JobPost: Seve
job post with salary 20000>, <JobPost: Eighth job post with salary 13500>, <JobPost: Nineth job
st with salary 2500>]>
>>> job_post_10 = JobPost(title="Tenth job post", description="Example description", salary=1100
>>> skill_1 = Skills(name="writing")
>>> skill_1.save()
>>> job_post_10.save()
>>> job_post_10.skills.add(skill_1)
>>>
```



```
models.py - jobapp - Visual Studio Code

File Edit Selection View Go Run Terminal Help

models.py x 0007_author_jobpost_author.py admin.py job_detail.hti

app > models.py > JobPost
18 country = models.CharField(max_length=200)
19 skills = models.CharField(max_length=200)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

>>> job_post_10
<JobPost: Tenth job post with salary 11000>
>>> job_post_10.skills.all()
<QuerySet [<Skills: Skills object (1)>, <Skills: Skills object (2)>, <Skills: Skills object (3)>]>
>>> skill_2
<Skills: Skills object (2)>
>>> skill_2.jobpost_set.all()
<QuerySet [<JobPost: Tenth job post with salary 11000>, <JobPost: Nineth job post with salary 2500>]>
>>> JobPost.objects.filter(skills__id=1)
<QuerySet [<JobPost: Tenth job post with salary 11000>]>
>>> JobPost.objects.filter(skills__pk=1)
<QuerySet [<JobPost: Tenth job post with salary 11000>]>
>>> JobPost.objects.filter(skills=1)
<QuerySet [<JobPost: Tenth job post with salary 11000>]>
>>> JobPost.objects.filter(skills=skill_1)
<QuerySet [<JobPost: Tenth job post with salary 11000>]>
>>>
```

python-django-4-1900-many-to-many-relationship Spaces: 4 UTF-8 CRLF Python 3.10.5 (env: venv)

```
File Edit Selection View Go Run Terminal Help models.py - jobapp - Visual Studio Code

models.py x 0007_author_jobpost_author.py admin.py job_detail.ht

app > models.py > JobPost
18 country = models.CharField(max_length=200)
19

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

<QuerySet [<JobPost: Tenth job post with salary 11000>, <JobPost: Ninth job post with salary
2500>]>
>>> job_post_10
<JobPost: Tenth job post with salary 11000>
>>> job_post_10.skills.all()
<QuerySet [<Skills: Skills object (1)>, <Skills: Skills object (2)>, <Skills: Skills object (
3)>]>
>>> job_post_10.skills.remove(skill_2)
>>> job_post_10.skills.all()
<QuerySet [<Skills: Skills object (1)>, <Skills: Skills object (3)>]>
>>> job_post_10.id
13
>>> skill_2
<Skills: Skills object (2)>
>>> skill_2.jobpost_set.all()
<QuerySet [<JobPost: Ninth job post with salary 2500>]>
>>> skill_2.id
2
>>>
```

python-django-4-1900-many-to-many-relationship* 0 0 0 Spaces: 4 UTF-8 CRLF Python 3.10.5 (env: venv)

The screenshot shows the Visual Studio Code interface with the file `models.py` open. The code defines a `JobPost` model with a `country` field. Below the code, the `TERMINAL` tab is active, displaying a Django shell session. The session includes commands to query the database, create a `job_post_10` instance, and add it to the `jobpost_set` of `skill_2`. A `NameError` is shown for an undefined `skill_3`.

```
models.py - jobapp - Visual Studio Code
```

models.py

```
country = models.CharField(max_length=200)
```

TERMINAL

```
>>> skill_2.jobpost_set.all()
<QuerySet []>
>>> job_post_10.id
13
>>> skill_3
Traceback (most recent call last):
  File "<console>", line 1, in <module>
NameError: name 'skill_3' is not defined
>>> skill_2
<Skills: Skills object (2)>
>>> job_post_10.skills.set([skill_1, skill_2])
>>> job_post_10.skills.set([skill_1])
>>> job_post_10.skills.clear()
>>> job_post_10.skills.all()
<QuerySet []>
>>> skill_2
<Skills: Skills object (2)>
>>> skill_2.jobpost_set.add(job_post_9, job_post_10)
>>>
```

python-django-4-1900-many-to-many-relationship* 0 0 Spaces: 4 UTF-8 CRLF Python 3.10.5 (env: venv)

One to one relationship

- One to one relationship is a type of relationship where in one record in a table is related to exactly one more record in another table and vice versa.

Example

- One employee has a salary account and one salary account belongs to exactly one employee
- One passport belongs to one person and one person has one passport

One to one relationship

```
location = models.OneToOneField(Location,  
on_delete=models.CASCADE, null=True)
```

Many to One relationship

- A many to one relationship is a type of relationship when one record in one table has one or many related record in another table

Example

- One customer can have multiple orders, however one order can belong to a single customer
- One user can have multiple bank accounts, however one bank account can belong to a single account holder

Many to One relationship

```
author = models.ForeignKey(Author,  
on_delete=models.CASCADE, null=True)
```

Many to Many relationship

- A many to many relationship is a type of relationship when one record in one table has many related record in another table

Example

- One customer can have purchase multiple products and products can be purchased by multiple customers as well
- One course can have multiple students, and one student take multiple courses.

Many to Many relationship

```
skills = models.ManyToManyField(Skills)
```