Models.py

from \_\_future\_\_ import unicode\_literals

from django.db import models

*class* book(*models*.*Model*):

name = models.CharField(*max\_length* = 255)

desc = models.TextField(*default* = " ")

created\_at = models.DateTimeField(*auto\_now\_add* = True)

update\_at = models.DateTimeField(*auto\_now* = True)

*class* author(*models*.*Model*):

first\_name = models.CharField(*max\_length* = 255)

last\_name = models.CharField(*max\_length* = 255)

email = models.CharField(*max\_length* = 255)

book = models.ManyToManyField(book, *related\_name* = "authors")

created\_at = models.DateTimeField(*auto\_now\_add* = True)

update\_at = models.DateTimeField(*auto\_now* = True)

>>> from apps.booksauthors.models import \*

>>> book.objects.create(name="C Sharp", desc="Coding book for C#")

<book: book object (1)>

>>> book.objects.create(name="Java", desc="Coding book for Java")

<book: book object (2)>

>>> book.objects.create(name="Python", desc="Coding book for Python")

<book: book object (3)>

>>> book.objects.create(name="PHP", desc="Coding book for PHP")

<book: book object (4)>

>>> book.objects.create(name="Ruby", desc="Coding book for Ruby")

<book: book object (5)>

>>> author.objects.create(first\_name="Mike", last\_name="Williams", email="mw@aol.com")

<author: author object (1)>

>>> author.objects.create(first\_name="Speros", last\_name="Greek", email="sg@aol.com")

<author: author object (2)>

>>> author.objects.create(first\_name="John", last\_name="Troy", email="jt@aol.com")

<author: author object (3)>

>>> author.objects.create(first\_name="Jadee", last\_name="Jay", email="jj@aol.com")

<author: author object (4)>

>>> author.objects.create(first\_name="Jay", last\_name="Blue", email="jb@aol.com")

<author: author object (5)>

from \_\_future\_\_ import unicode\_literals

from django.db import models

*class* book(*models*.*Model*):

name = models.CharField(*max\_length* = 255)

desc = models.TextField(*default* = " ")

created\_at = models.DateTimeField(*auto\_now\_add* = True)

update\_at = models.DateTimeField(*auto\_now* = True)

*class* author(*models*.*Model*):

first\_name = models.CharField(*max\_length* = 255)

last\_name = models.CharField(*max\_length* = 255)

email = models.CharField(*max\_length* = 255)

book = models.ManyToManyField(book, *related\_name* = "authors")

notes = models.TextField(*default* = " ")

created\_at = models.DateTimeField(*auto\_now\_add* = True)

update\_at = models.DateTimeField(*auto\_now* = True)

)

$ python manage.py makemigrations

Migrations for 'booksauthors':

apps\booksauthors\migrations\0002\_author\_notes.py

- Add field notes to author

(DJANGO~1)

User@Osburn-Family-Laptop MINGW64 /i/coding dojo/dojoassignments/python/python\_stack/django/booksauthors (master)

$ python manage.py migrate

Operations to perform:

Apply all migrations: admin, auth, booksauthors, contenttypes, sessions

Running migrations:

Applying booksauthors.0002\_author\_notes... OK

(DJANGO~1)

User@Osburn-Family-Laptop MINGW64 /i/coding dojo/dojoassignments/python/python\_stack/django/booksauthors (master)

$

>>> from apps.booksauthors.models import \*

>>> b = book.objects.get(id=5)

>>> b.name = "C #"

>>> b.save()

>>> a = author.objects.get(id=5)

>>> a.first\_name = "Ketul"

>>> a.save()

>>> myauthor.books.add(mybook)

>>> myauthor = Author.objects.get(id=2)

>>> mybook = Book.objects.get(id=1)

>>> myauthor.books.add(mybook)

>>> mybook = Book.objects.get(id=2)

>>> myauthor.books.add(mybook)

>>> mybook = Book.objects.get(id=3)

>>> myauthor.books.add(mybook)

>>>

>>> myauthor = Author.objects.get(id=3)

>>> mybook = Book.objects.get(id=1)

>>> myauthor.books.add(mybook)

>>> mybook = Book.objects.get(id=2)

>>> myauthor.books.add(mybook)

>>> mybook = Book.objects.get(id=4)

>>> myauthor.books.add(mybook)

>>> Book.objects.get(id=3).authors

<django.db.models.fields.related\_descriptors.create\_forward\_many\_to\_many\_manager.<locals>.ManyRelatedManager obj

ect at 0x000002762BA0AA20>

>>> p = Book.objects.get(id=3).author

Traceback (most recent call last):

File "<console>", line 1, in <module>

AttributeError: 'Book' object has no attribute 'author'

>>> p = Book.objects.get(id=3).authors

>>> print p

File "<console>", line 1

print p

^

SyntaxError: Missing parentheses in call to 'print'

>>> print(p)

booksauthors.Author.None

>>> Author.objects.first().books

<django.db.models.fields.related\_descriptors.create\_forward\_many\_to\_many\_manager.<locals>.ManyRelatedManager obj

ect at 0x000002762BA32BE0>