Create a new model called *'Book'* with the information above.

Create a new model called 'Author' with the information above.  Design the models in a way that you could perform the following:

Book.objects.first().authors

Author.objects.first().books

class Book(models.Model):

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

desc = models.TextField(max\_length=1000)

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

updated\_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)

books = models.ManyToManyField(Book, related\_name="authors")

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

updated\_at = models.DateTimeField(auto\_now = True)

Successfully create and run the migration files

python manage.py makemigrations

python manage.py migrate

*python manage.py shell*

*from apps.book\_authors.models import \**

Using the shell...

Create 5 books with the following names: C sharp, Java, Python, PHP, Ruby

>>> from apps.book\_authors.models import \*

>>> b1 = Book(name = 'C sharp', desc = '')

>>> b2 = Book(name = 'Java', desc = '')

>>> b3 = Book(name = 'Python', desc = '')

>>> b4 = Book(name = 'PHP', desc = '')

>>> b5 = Book(name = 'Ruby', desc = '')

>>> b1.save()

>>> b2.save()

>>> b3.save()

>>> b4.save()

>>> b5.save()

Create 5 different authors: Mike, Speros, John, Jadee, Jay

>>> a1 = Author(first\_name = 'Mike', last\_name ='l1', email = '1@1.com')

>>> a2 = Author(first\_name = 'Speros', last\_name ='l2', email = '2@2.com')

>>> a3 = Author(first\_name = 'John', last\_name ='l3', email = '3@4.com')

>>> a4 = Author(first\_name = 'Jadee', last\_name ='l4', email = '4@4.com')

>>> a5 = Author(first\_name = 'Jay', last\_name ='l5, email = '5@5.com')

>>> a5 = Author(first\_name = 'Jay', last\_name ='l5', email = '5@5.com')

>>> a1.save()

>>> a2.save()

>>> a3.save()

>>> a4.save()

>>> a5.save()

Add a new field in the authors table called 'notes'.  Make this a TextField.  Successfully create and run the migration files.

class Author(models.Model):

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

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

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

books = models.ManyToManyField(Book, related\_name="authors")

notes = models.TextField(blank=True, null=True)

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

updated\_at = models.DateTimeField(auto\_now = True)

python manage.py makemigrations

python manage.py migrate

*python manage.py shell*

*from apps.book\_authors.models import \**

Using the shell...

Change the name of the 5th book to C#

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

>>> book.name = 'C#'

>>> book.save()

Change the first\_name of the 5th author to Ketul

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

>>> author.first\_name = 'Ketul'

>>> author.save()

Assign the first author to the first 2 books

this\_book = Book.objects.get(id=1)

this\_author = Author.objects.get(id=1)

this\_author.books.add(this\_book)

this\_author.save()

this\_book2 = Book.objects.get(id = 2)

this\_author.books.add(this\_book2)

this\_author.save()

Assign the second author to the first 3 books

this\_author = Author.objects.get(id=2)

this\_book = Book.objects.get(id=1)

this\_author.books.add(this\_book)

this\_author.save()

this\_author = Author.objects.get(id=2)

this\_book = Book.objects.get(id=2)

this\_author.books.add(this\_book)

this\_author.save()

this\_author = Author.objects.get(id=2)

this\_book = Book.objects.get(id=3)

this\_author.books.add(this\_book)

this\_author.save()

Assign the third author to the first 4 books

this\_author = Author.objects.get(id=3)

this\_book = Book.objects.get(id=1)

this\_author.books.add(this\_book)

this\_author.save()

this\_book = Book.objects.get(id=2)

this\_author.books.add(this\_book)

this\_author.save()

this\_book = Book.objects.get(id=3)

this\_author.books.add(this\_book)

this\_author.save()

this\_book = Book.objects.get(id=4)

this\_author.books.add(this\_book)

this\_author.save()

Assign the fourth author to the first 5 books (or in other words, all the books)

this\_author = Author.objects.get(id=4)

this\_book = Book.objects.get(id=1)

this\_author.books.add(this\_book)

this\_author.save()

this\_book = Book.objects.get(id=2)

this\_author.books.add(this\_book)

this\_author.save()

this\_book = Book.objects.get(id=3)

this\_author.books.add(this\_book)

this\_author.save()

this\_book = Book.objects.get(id=4)

this\_author.books.add(this\_book)

this\_author.save()

this\_book = Book.objects.get(id=5)

this\_author.books.add(this\_book)

this\_author.save()

For the 3rd book, retrieve all the authors

Book.objects.get(id=3).authors.all().values()

For the 3rd book, remove the first author

--the 2nd author is in the 1st 3 books.

??

For the 2nd book, add the 5th author as one of the authors

this\_author = Author.objects.get(id=5)

this\_book = Book.objects.get(id=2)

this\_author.books.add(this\_book)

this\_author.save()

Find all the books that the 3rd author is part of

Author.objects.get(id=3).books.all().values()

Find all the books that the 2nd author is part of

Author.objects.get(id=2).books.all().values()