(DJANGO~1) C:\Users\jinal\Desktop folders\Desktop\DojoAssignments\Python\Django\_ORM\Likes\_Books>python manage.py shell

Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32

Type "help", "copyright", "credits" or "license" for more information.

(InteractiveConsole)

>>> from apps.likes\_books.models import \*

## Create 3 different user accounts

>>> User.objects.create(first\_name="Balaguru", last\_name="swami", email="balas@gmail.com")

<User: User object>

>>> User.objects.create(first\_name="steven", last\_name="ring", email="stevens@gmail.com")

<User: User object>

>>> User.objects.create(first\_name="richard", last\_name="john", email="richard@gmail.com")

<User: User object>

## Create different books

>>> Book.objects.create(name="book1", desc="one by one", uploader=User.objects.get(id=1))

<Book: Book object>

>>> Book.objects.create(name="book2", desc="one by two", uploader=User.objects.get(id=2))

<Book: Book object>

>>> Book.objects.create(name="book3", desc="third book", uploader=User.objects.get(id=2))

<Book: Book object>

>>> Book.objects.create(name="book4", desc="forth", uploader=User.objects.get(id=1))

<Book: Book object>

>>> Book.objects.create(name="book5", desc="fifth", uploader=User.objects.get(id=3))

<Book: Book object>

## **Book.objects.first().uploader - this should return the user who uploaded the book**

>>> Book.objects.first().uploader

<User: User object>

## User.objects.first().uploaded\_books - this should return all the books that are uploaded by the first user

>>> User.objects.first().uploaded\_books.all()

<QuerySet [<Book: Book object>, <Book: Book object>]>

## Book.objects.first().liked\_users - this should return all the users who liked the first book

>>> user = User.objects.get(id=1)

>>> user2 = User.objects.get(id=2)

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

>>> book.likers.add(user)

>>> book.likers.add(user2)

>>> **book.likers.all()**

<QuerySet [<User: User object>, <User: User object>]>

## User.objects.first().liked\_books - this should return all the books that were liked by the first user

>>> **user.liked\_books.all()**

<QuerySet [<Book: Book object>]>

>>> User.objects.get(id=2).uploaded\_books.all()

<QuerySet [<Book: Book object>, <Book: Book object>]>

>>> User.objects.get(id=1).uploaded\_books.all()

<QuerySet [<Book: Book object>, <Book: Book object>]>

>>> User.objects.get(id=3).uploaded\_books.all()

<QuerySet [<Book: Book object>]>

>>> Book.objects.create(name="book6", desc="6th book by user 3", uploader=User.objects.get(id=3))

<Book: Book object>

>>> User.objects.get(id=3).uploaded\_books.all()

<QuerySet [<Book: Book object>, <Book: Book object>]>

## Have the first user like the last book and the first book

>>> Book.objects.last().likers.add(User.objects.first())

## Have the second user like the first book and the third book

>>> Book.objects.get(id=3).likers.add(User.objects.get(id=2))

## Have the third user like all books

>>> User.objects.get(id=3).liked\_books.add(\*Book.objects.all())

## Display all users who like the first book

>>> Book.objects.get(id=1).likers.all()

<QuerySet [<User: User object>, <User: User object>, <User: User object>]>

## Display the user who uploaded the first book

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

<User: User object>

## Display all users who like the second book

>>> Book.objects.get(id=2).likers.all()

<QuerySet [<User: User object>]>

# Display the user who uploaded the second book

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

<User: User object>