

Solution TD N° 2 : Django - Modèles

Partie 1: Création des modèles

```
# file library/models.py

from django.db import models

class Publisher(models.Model):
    name = models.CharField(max_length=200)
    city = models.CharField(max_length=100)
    website = models.URLField()

    def __str__(self):
        return self.name

class Category(models.Model):
    name = models.CharField(max_length=100)

    def __str__(self):
        return self.name

    def count_books_in_category(self):
        return self.book_set.count()

class Book(models.Model):
    title = models.CharField(max_length=200)
    author = models.CharField(max_length=100)
    publication_date = models.DateField()
    isbn = models.CharField(max_length=13, unique=True)
    publisher = models.ForeignKey(Publisher, on_delete=models.CASCADE)
    categories = models.ManyToManyField(Category)

    def __str__(self):
        return self.title

    @staticmethod
    def get_books_by_author(author_name):
        return Book.objects.filter(author=author_name)
```

Partie 2: Interagir avec les modèles via le shell Django

1. Lancer le shell interactif Django

```
(.venv) PS D:\libraray> python manage.py shell
```

2. Importer les modèles

```
>>> from library.models import Book, Publisher, Category
```

3. Créer des instances

a. Création des éditeurs :

```
>>> publisher1 = Publisher(name="Penguin Random House", city="New York", website="https://www.penguinrandomhouse.com")
>>> publisher1.save()
```

```
>>> publisher2 = Publisher.objects.create(name="HarperCollins", city="London", website="https://www.harpercollins.com")
```

b. Création des Catégories :

```
>>> category1 = Category.objects.create(name="Science Fiction")
```

```
>>> category2 = Category.objects.create(name="Fantasy")
```

```
>>> category3 = Category.objects.create(name="History")
```

c. Création des livres :

```
>>> book1 = Book(title="Dune", author="F. Herbert", publication_date="1965-08-01", isbn="9780441013593", publisher=publisher1)
>>> book1.save()
```

```
>>> book2 = Book(title="The Hobbit", author="J. Tolk", publication_date="1937-09-21", isbn="9780547928227", publisher=publisher2)
>>> book2.save()
```

```
>>> book3 = Book(title="Sapiens", author="N. Harari", publication_date="2011-06-01", isbn="9780062316097", publisher=publisher1)
>>> book3.save()
```

d. Associer des catégories aux livres :

```
>>> book1.categories.add(category1, category2)

>>> book2.categories.add(category2)

>>> book3.categories.add(category3)
```

Remarque : Les relations `ManyToMany` permettent de relier plusieurs objets d'un modèle à plusieurs objets d'un autre modèle. Les méthodes qu'on peut utiliser sont : `add()`, `remove()`, `clear()` et `all()`

4. Récupérer tous les livres de l'auteur "F. Herbert" :

```
>>> books_by_herbert = Book.get_books_by_author("F. Herbert")
>>> print(books_by_herbert)
```

5. Compter les livres dans une catégorie :

```
>>> sci_fi_books = category1.count_books_in_category()
>>> print(f"Nombre de livres en Science Fiction : {sci_fi_books}")
```

6. Lister les livres associés à une catégorie (on utilise le suffixe `__set` pour gérer les relations inverses dans `ForeignKey` ou `ManyToManyField`) :

```
>>> books=publisher1.book_set.all()
>>> print(books)
```

7. Mettre à jour des objets

a. Modifier le titre d'un livre :

```
>>> book1.title = "Dune (Updated Edition)"
>>> book1.save()
```

b. Supprimer un livre :

```
>>> book2.delete()
```