Ejercicio: Registro de Modelos en Django y Consultas en el Shell

Objetivo: Practicar la creación y el registro de modelos en la administración de Django, además de realizar consultas desde el shell interactivo de Django.

Crea una aplicación de gestión de biblioteca en Django. La aplicación debe permitir registrar libros, autores y editoriales. También se deben gestionar préstamos de libros. Al finalizar, realizarás varias consultas desde el shell interactivo de Django para practicar.

Pasos:

1. **Crear una nueva aplicación Django:**
    - Crea un proyecto Django llamado `LibrarySystem`.
    - Dentro del proyecto, crea una aplicación llamada `library`.
2. **Definir modelos:**
    - En el archivo `models.py` de la aplicación `library`, define los siguientes modelos:
        - `Author` con campos:
            - `first_name` (CharField)
            - `last_name` (CharField)
            - `birth_date` (DateField)
        - `Publisher` con campos:
            - `name` (CharField)
            - `address` (CharField)
            - `city` (CharField)
            - `state_province` (CharField)
            - `country` (CharField)
            - `website` (URLField)
        - `Book` con campos:
            - `title` (CharField)
            - `authors` (ManyToManyField con `Author`)
            - `publisher` (ForeignKey con `Publisher`)

        - `publication_date` (DateField)
    - `Loan` con campos:
        - `book` (ForeignKey con `Book`)
        - `borrower` (CharField)
        - `loan_date` (DateField)
        - `return_date` (DateField)

3. **Registrar los modelos en el administrador de Django:**
    - Abre el archivo `admin.py` de la aplicación `library`.
    - Registra los modelos `Author`, `Publisher`, `Book` y `Loan` en la administración de Django utilizando `admin.site.register()`.

4. **Realizar migraciones:**
    - Ejecuta los comandos para crear y aplicar las migraciones necesarias para los modelos definidos.

      ```
      python manage.py makemigrations
      python manage.py migrate
      ```

5. **Crear un superusuario:**
    - Crea un superusuario para acceder al panel de administración.

      ```
      python manage.py createsuperuser
      ```

6. **Probar en la administración:**
    - Inicia el servidor de desarrollo de Django.

      ```
      python manage.py runserver
      ```

    - Accede a `http://127.0.0.1:8000/admin/` y prueba las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) para cada modelo.

7. **Realizar consultas en el shell de Django:**
    - Abre el shell interactivo de Django.

      ```
      python manage.py shell
      ```

    - Importa los modelos y realiza las siguientes consultas:
        - **Crear instancias:**

          ```
          from library.models import Author, Publisher, Book, Loan
          author = Author.objects.create(first_name="Gabriel",
          last_name="García Márquez", birth_date="1927-03-06")
          publisher = Publisher.objects.create(name="Editorial
          Sudamericana", address="Calle Falsa 123", city="Buenos
          Aires", state_province="Buenos Aires",
          ```

```
country="Argentina",
website="http://editorialsudamericana.com")
book = Book.objects.create(title="Cien años de soledad",
publisher=publisher, publication_date="1967-06-05")
book.authors.add(author)
```

- **Consultar libros por autor:**

```
books_by_author =
Book.objects.filter(authors__first_name="Gabriel")
```

- **Consultar libros publicados por una editorial específica:**

```
books_by_publisher =
Book.objects.filter(publisher__name="Editorial
Sudamericana")
```

- **Consultar préstamos activos (sin fecha de devolución):**

```
active_loans =
Loan.objects.filter(return_date__isnull=True)
```

- **Actualizar la fecha de devolución de un préstamo:**

```
loan = Loan.objects.first()
loan.return_date = "2025-01-10"
loan.save()
```

- **Eliminar un libro:**

```
book_to_delete = Book.objects.get(title="Cien años de
soledad")
book_to_delete.delete()
```

Este ejercicio adicional te ayudará a practicar el uso del ORM de Django para realizar consultas en el shell, lo que es fundamental para manipular y consultar datos en tu base de datos de forma efectiva.