

Tarea. Gestión de una Biblioteca Digital con SQLAlchemy y PostgreSQL

Descripción de la tarea:

Se va a desarrollar un sistema de gestión para una **biblioteca digital** utilizando SQLAlchemy y PostgreSQL. El objetivo es crear una aplicación que permita gestionar libros, autores y préstamos de libros por parte de usuarios de la biblioteca. Los alumnos deberán definir las relaciones necesarias entre las entidades y realizar operaciones CRUD para mantener la integridad de los datos.

Requisitos del sistema:

1. Entidades del sistema:

o Libro: Cada libro tiene un título, una fecha de publicación y un número ISBN

único.

o Autor: Cada autor tiene un nombre y una biografía.

o Usuario: Los usuarios tienen un nombre y un correo electrónico.

o Préstamo: Un usuario puede tomar prestado un libro, registrándose la fecha

del préstamo y la fecha límite de devolución.

2. Relaciones entre entidades:

o Un Libro tiene un único Autor y este puede escribir varios libros

o Un Usuario puede tomar prestados varios Libros y estos a lo largo del tiempo han sido prestados a varios usuarios (relación muchos a muchos), para ello, tras una relación entre estas dos, se va a crear la tabla de préstamo

o Se debe registrar qué Libros ha tomado prestados un Usuario y cuándo (relación uno a muchos a través de la tabla de Préstamos).

3. Operaciones a implementar:

o Crear: Poder registrar libros, autores y usuarios en el sistema.

- **Leer:** Consultar todos los libros prestados por un usuario, y todos los libros disponibles en la biblioteca.
- **Actualizar:** Poder actualizar la información de los libros y los autores.
- **Eliminar:** Eliminar libros del sistema y gestionar préstamos de libros que ya fueron devueltos.

Esquema de las tablas:

1. **Libro:** id, titulo, fecha_publicacion, isbn, autor_id

2. **Autor:** id, nombre, biografía

3. **Usuario:** id, nombre, email

4. **Préstamo:** id, libro_id, usuario_id, fecha_prestamo, fecha_limite_devolucion

Tareas:

1. Definir los **modelos** para las entidades Libro, Autor, Usuario, Préstamo y la tabla intermedia autor_libro para la relación muchos a muchos entre libros y autores.
2. Establecer las **relaciones** necesarias entre las tablas usando ForeignKey y relationship().
3. Crear una sesión utilizando sessionmaker() y realizar las operaciones CRUD solicitadas.
4. Implementar las siguientes **consultas**:
 - Obtener todos los libros prestados por un usuario, mostrando la fecha de préstamo.
 - Consultar los libros disponibles (los que no han sido prestados).
 - Actualizar la biografía de un autor.
 - Eliminar un préstamo cuando un usuario devuelve un libro.

Criterios de evaluación:

El ejercicio se calificará sobre 10 puntos, distribuidos de la siguiente

manera: 1. **Definición de modelos (2 puntos):**

- Los modelos deben estar bien definidos con las columnas y tipos de datos correctos.
- Uso adecuado de claves primarias y foráneas.

2. Creación de relaciones (2 puntos):

- Las relaciones entre Libro, Autor, Usuario y Préstamo deben estar correctamente implementadas utilizando relationship() y ForeignKey.

3. Implementación de operaciones CRUD (2 puntos):

- Los alumnos deben implementar correctamente las operaciones de creación, lectura, actualización y eliminación.

4. Consultas avanzadas (2 puntos):

- Las consultas solicitadas deben funcionar correctamente, mostrando los datos según los requisitos.

5. Uso adecuado de la sesión y transacciones (2 puntos):

- La gestión de la sesión debe ser correcta, incluyendo el uso de commit() y rollback() en las operaciones, si es necesario.