

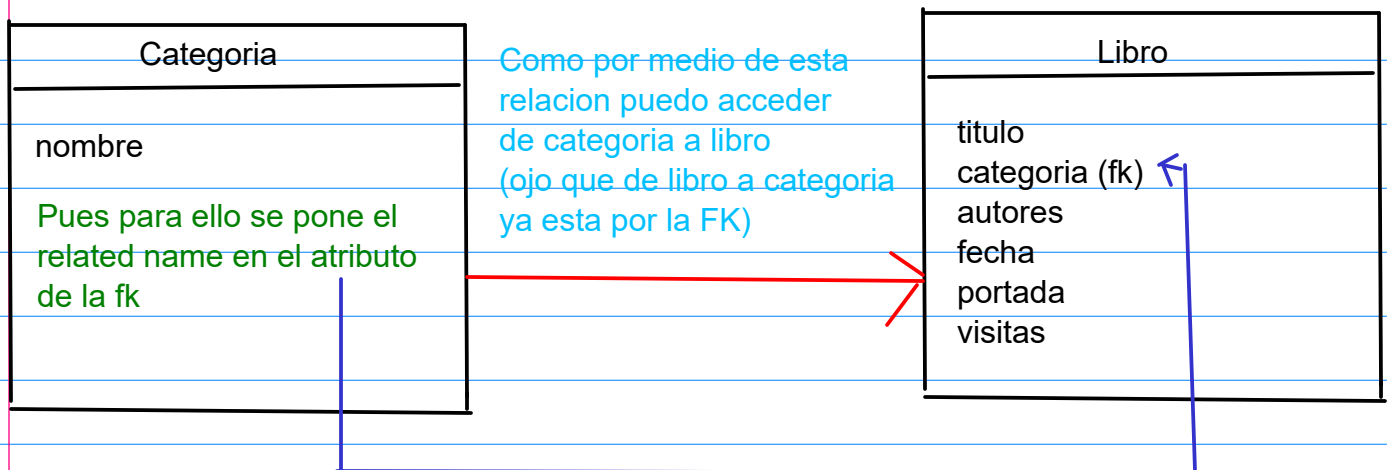
Requerimiento:

Listar todas las categorías de un autor

Los modelos de categoría y autor no están relacionados directamente, entonces por medio de la relación que existe entre libro y categoría se pueden relacionar el autor con la categoría.

Para ello usamos el `related_name` que hace referencia a la relación inversa de la clave foránea, en pocas palabras nos sirve para acceder desde categoría a libro

Si se quiere acceder a libro desde categoría se puede usar este `related_name`



Y por medio de ese `related name` se puede acceder a Libro desde Categoria es decir al libro asociado a esa categoria

`categoria_libro` hace referencia a la relacion de libro con su fk de categoria respectiva y una vez unidos estos modelos accedo al otro atributo de la clase libro para hacer la pregunta... `categoria_libro_autores_id = autor`

Resumiendo el modelo libro le jala a toda la clase Categoria para al momento de hacer la pregunta retornar la categoria del libro

Clase diferencia entre Annotate y Aggregate

Aggregate: Devuelve un diccionario de valores agregados (promedios, sumas, etc.) calculados sobre el QuerySet. Cada argumento para aggregate() especifica un valor que se incluirá en el diccionario que se devuelve.

Se plantea contar cuantas veces fue prestado un libro y se intentará hacerlo usando el aggregate

De igual forma en el modelo prestamo se añade el related_name para poder acceder a la cantidad de relaciones y poder contarlas.

OJO que el Aggregate me retorna un DICCIONARIO a diferencia del Annotate que me retorna un QuerySet

¿En qué caso uso el Annotate y el Aggregate?

Para el caso del conteo que queremos en una consulta o en un queryset, obligatoriamente utilizaremos el annotate, pero si solamente necesitamos una operación aritmética para encontrar un valor, vamos a utilizar el aggregate

Se plantea que de un determinado libro del modelo libro, por ejemplo del libro A se calcule el promedio de la edad de los lectores que se prestan ese libro A