1.relaciones:

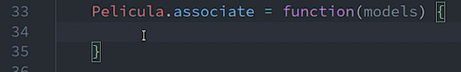
a. 1:N

b. N:M

1.

Relaciones: en MySQL para relacionar las tablas usábamos JOIN, con Sequelize es distinto, tanto las relaciones de una a muchos como de muchos a 1, uno a uno, muchos a muchos.

Para implementarlas en cada modelo primero hay que creas una función donde utilizaremos estos metos (crear un “contexto” de chee estoy creando asociaciones, acá adentro te explico cómo son):



a.

**1:N** ("Un género tiene muchas películas y una películas pertenece a un géneros".)

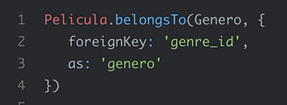
Dos funciones de sequelize para informar las relaciones entre las tablas géneros y películas: **hasMeny()** (tiene muchas) y **belongsTo()** (pertenece a)**.**

ambas son aplicadas a ambas de las tablas a relacionar

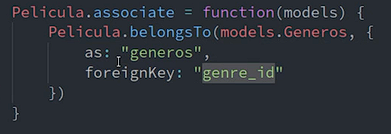
**belongsTo()**:

una película pertenece a un género. En el modelo de película usaremos este método.

A este método le pasamos como parámetro **primero el modelo a asociar**, y como segundo parámetro un **objeto literal** con la propiedad “as” que indica el nombre que le vamos dar a la relación, y “foreinhKey” con el nombre de la clave foránea.

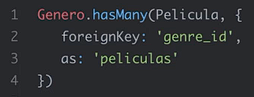


Ejemplo práctico: en el modelo Pelicula



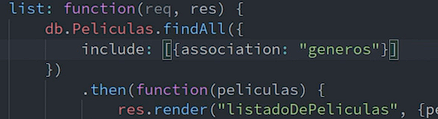
Ahora puedo preguntar por la el genero que tiene una película. Pero no puedo preguntar que películas tiene un género en cuestión.

Para completar la relación en el modelo de géneros utilizaremos el método **hasMany()** en el modelo de Genero

****

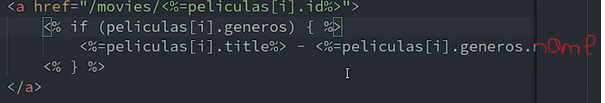
**¿Cómo lo utilizamos?**

Si quisiéramos que en el listado de todas la películas también nos mostrara el género de cada una, debemos ir al controlador, al método que enlista todas la películas y al método findAll() le pasamos una configuración, un **objeto literal** con el atributo **include,** como valor le pasamos el association con generos todo en otro objeto literal



Así le pasamos la relación a la viste, y ahora ¿cómo hacemos para mostrarlo en la vista?

Al lado del título le pedimos que de la película en cuestión nos de el género, ¿por qué “géneros”? porque ese es el nombre que le dimos en la asociación. Y a generos le pedimos name, o cualquier columna que tenga la tabla de géneros



El if es por si hay peliculas que no tengas genero

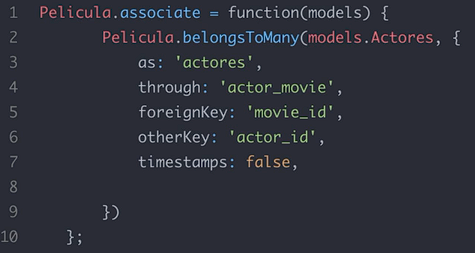
b.

**N:M** En actor trabaja en muchas películas y una película tiene muchos actores.

Utilizaremos la funcion **belongsToMany(). Recibe dos parámetros**: el alias del modelo que queremos relacionar y como segundo un objeto literal con configuracion

En uno de los modelos que queremos asociar, en este caso **Pelicula**, creamos la función de **associate()**. Dentro utilizaremso la funcions **belongsToMany(),** a cual recibe el otro medelo de tabla que queremos relacionar y un objeto literal con configuraciones.

Para completar la relación hay que hacer una función similar, pero en el otro modelo.



as: hacer referencia al nombre de la relación, se utilizará en el controlador para pasarle esa info a las vistas

through: significa a través, hacer referencia a la tabla “pivot” entre los dos modelos.

foreignKey: hace referencia a la FK de la tabla pivot correspondiente al modelo actual

otherKey: hace referencia a la FK de la tabla pivot correspondiente al modelo al que nos queremos relacionar

timestamps: porque la tabla pivot no tiene las columnas created\_at ni updated\_at.

Ejemplo practico:

Las relaciones el los modelos es igual que el ejemplo de arriba.

En el controles, en el método de enlistar, al includes como es una array de asociaciones le pasamos la asocciascion de mucho a mucho que hicismos:



En la vista, debajo del título de cada película enlistaremos los actores que participan en dicha película: 