

RELACIÓN ENTRE ENTIDADES

TIPOS

- Relación Uno a Uno
- Relación Uno a Muchos
- Relación Muchos a Muchos

RELACIÓN UNO A UNO

- Supongamos que deseamos rastrear la información del autor de un libro
- Esto puede establecerse como una relación uno a uno, ya que cada libro tiene un único autor y cada autor está asociado con un solo libro

RELACIÓN UNO A UNO

- Aquí hay una migración de ejemplo para agregar la información del autor a la tabla "books", antes de esto hemos creado la tabla de autores:

EJEMPLO

```
/* Run the migrations.
 */
public function up(): void
{
    Schema::table('books', function (Blueprint $table) {
        $table->unsignedBigInteger('author_id'); // Clave foránea para el autor
        $table->foreign('author_id')->references('id')->on('authors'); // Definición de la clave foránea
    });
}
```

EJEMPLO

- Por último, hay modificar el modal y se añade lo siguiente, si no está creado se crea con:

```
php artisan make:model <nombreModal>
```

```
class Book extends Model
{
    public function author()
    {
        return $this->belongsTo(Author::class);
    }
}
```

RELACIÓN UNO A MUCHOS

- Para representar una relación uno a muchos, agregaremos la entidad "Préstamo" a nuestra aplicación

RELACIÓN UNO A MUCHOS

- Un libro puede tener múltiples préstamos, pero cada préstamo está asociado con un solo libro. Y añadimos la relación al modal de libros y al de préstamos

Modal Libros

```
public function loans()  
{  
    return $this->hasMany(Loan::class);  
}
```

Modal Préstamos

```
public function book()  
{  
    return $this->belongsTo(Book::class);  
}
```


RELACIÓN MUCHOS A MUCHOS

- Supongamos que deseamos rastrear las categorías a las que pertenece un libro
- Un libro puede estar en múltiples categorías, y una categoría puede tener múltiples libros

RELACIÓN MUCHOS A MUCHOS

- Esto se representa como una relación muchos a muchos. Creamos una tabla “category” y “book_category”. Y añadimos a los 3 modelos la relación. Quedando así:

Modelo libro

```
public function categories()
{
    return $this->belongsToMany(Category::class);
}
```

Modelo Categoría

```
class Category extends Model
{
    public function books()
    {
        return $this->belongsToMany(Book::class);
    }
}
```

Modelo Libro-Categoría

```
class BookCategory extends Model
{
    protected $table = 'book_category'; // Nombre de la tabla intermedia

    // Definición de las relaciones
    public function book()
    {
        return $this->belongsTo(Book::class);
    }

    public function category()
    {
        return $this->belongsTo(Category::class);
    }
}
```