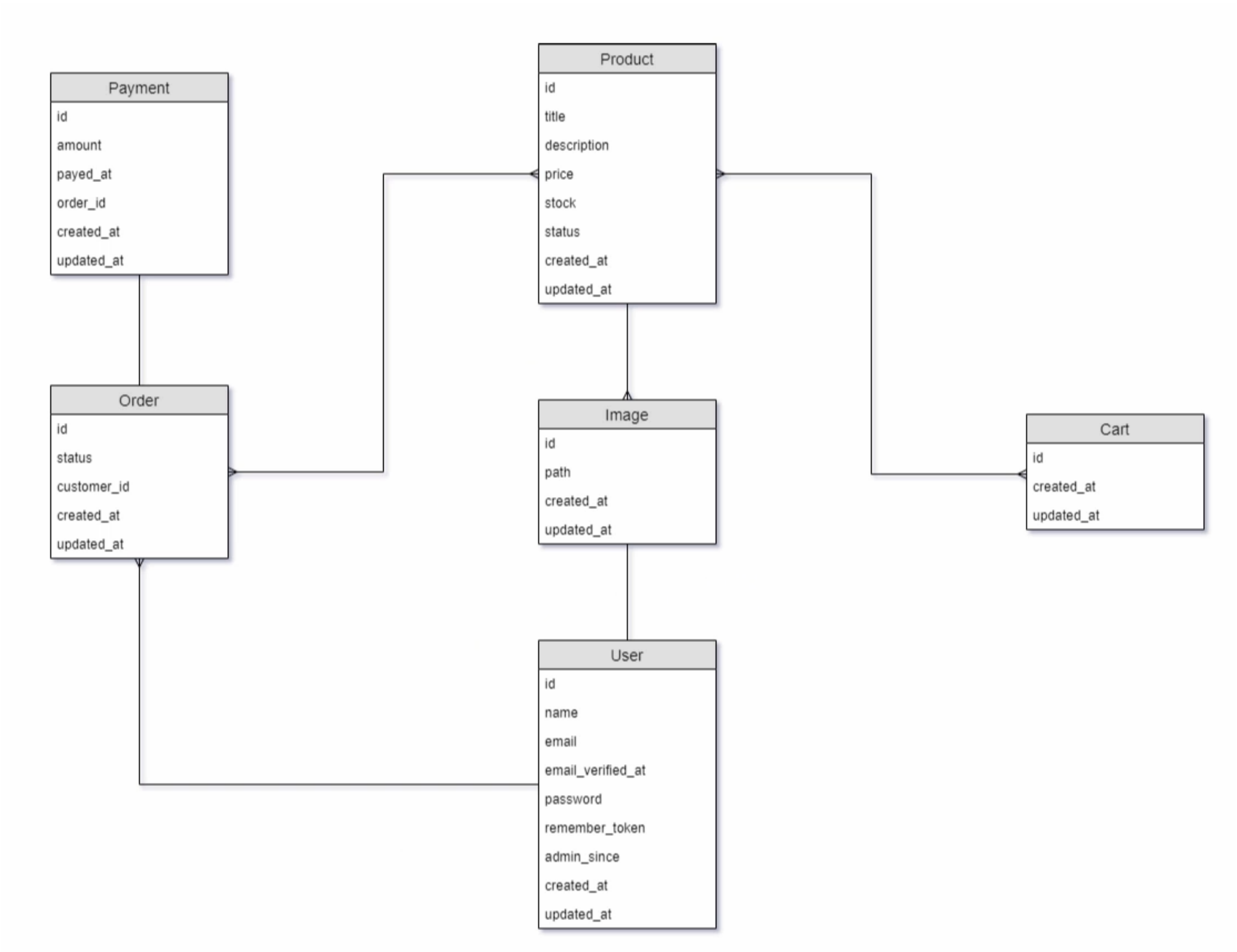
61. Conoce las relaciones polimórficas uno a muchos

Definición

Un **Product** puede tener múltiples **Image**, como vemos en el diagrama:



✓ **Al tener ya preparado el método `imageable()` en Image es mucho más rápido implementarla.**

Relaciones polimórficas 1:n en el proyecto

Image ↔ Product.

Laravel: crear las relaciones

En **Product**:

```
    public function images() {
        // Importante indicar el nombre de la relación polimórfica
        return $this->morphMany(Image::class, 'imageable');
    }
```

Tinker: probar las relaciones

```
C:\Users\llemi\Desktop\Lunk\Laravel\ola>php artisan tinker
Psy Shell v0.11.8 (PHP 8.0.9 — cli) by Justin Hileman

// Obtenemos el producto con id=2 de la lista de products
>>> $product = App\Models\Product::find(2)
=> App\Models\Product {#4558
     id: "2",
     title: "In iusto aspernatur porro.",
     description: "Perspiciatis iusto deserunt qui deleniti velit. Sequi quia debitis
explicabo voluptas.",
     price: "80.74",
     stock: "8",
     status: "available",
     created_at: "2022-11-12 11:15:27",
     updated_at: "2022-11-12 11:15:27",
   }

// Le asignamos una imagen mediante el método morphMany()
// de Product->images()
>>> $product->images()->save(App\Models\Image::factory()->make())
=> App\Models\Image {#4595
     path: "img/products/8.jpg",
     imageable_id: 2,
     imageable_type: "App\Models\Product",
     updated_at: "2022-11-12 12:27:00",
     created_at: "2022-11-12 12:27:00",
     id: 2,
   }

// Buscamos la imagen con ese id (fijémonos en el imageable_type)
>>> $image = App\Models\Image::find(2)
=> App\Models\Image {#4561
     id: "2",
     path: "img/products/8.jpg",
     created_at: "2022-11-12 12:27:00",
     updated_at: "2022-11-12 12:27:00",
     imageable_type: "App\Models\Product",
     imageable_id: "2",
   }

// Imageable sabe, por el imageable_type = Product y el imageable_id,
// que debe mostrar el Product con id = 2
>>> $image->imageable
=> App\Models\Product {#4587
     id: "2",
     title: "In iusto aspernatur porro.",
     description: "Perspiciatis iusto deserunt qui deleniti velit. Sequi quia debitis
explicabo voluptas.",
     price: "80.74",
     stock: "8",
     status: "available",
     created_at: "2022-11-12 11:15:27",
     updated_at: "2022-11-12 11:15:27",
   }
```

Creamos una nueva **Image**, la asignamos y comprobamos que, efectivamente, un **Product** puede contener muchas.

```
>>> $product->images()->save(App\Models\Image::factory()->make())
=> App\Models\Image {#4601
     path: "img/products/5.jpg",
     imageable_id: 2,
     imageable_type: "App\Models\Product",
     updated_at: "2022-11-12 12:34:29",
     created_at: "2022-11-12 12:34:29",
     id: 3,
   }

>>> $product = $product->fresh()
=> App\Models\Product {#4550
     id: "2",
     title: "In iusto aspernatur porro.",
     description: "Perspiciatis iusto deserunt qui deleniti velit. Sequi quia debitis
explicabo voluptas.",
     price: "80.74",
     stock: "8",
     status: "available",
     created_at: "2022-11-12 11:15:27",
     updated_at: "2022-11-12 11:15:27",
   }

>>> $product->images
=> Illuminate\Database\Eloquent\Collection {#4565
     all: [
       App\Models\Image {#4564
         id: "2",
         path: "img/products/8.jpg",
         created_at: "2022-11-12 12:27:00",
         updated_at: "2022-11-12 12:27:00",
         imageable_type: "App\Models\Product",
         imageable_id: "2",
       },
       App\Models\Image {#4590
         id: "3",
         path: "img/products/5.jpg",
         created_at: "2022-11-12 12:34:29",
         updated_at: "2022-11-12 12:34:29",
         imageable_type: "App\Models\Product",
         imageable_id: "2",
       },
     ],
   }
```