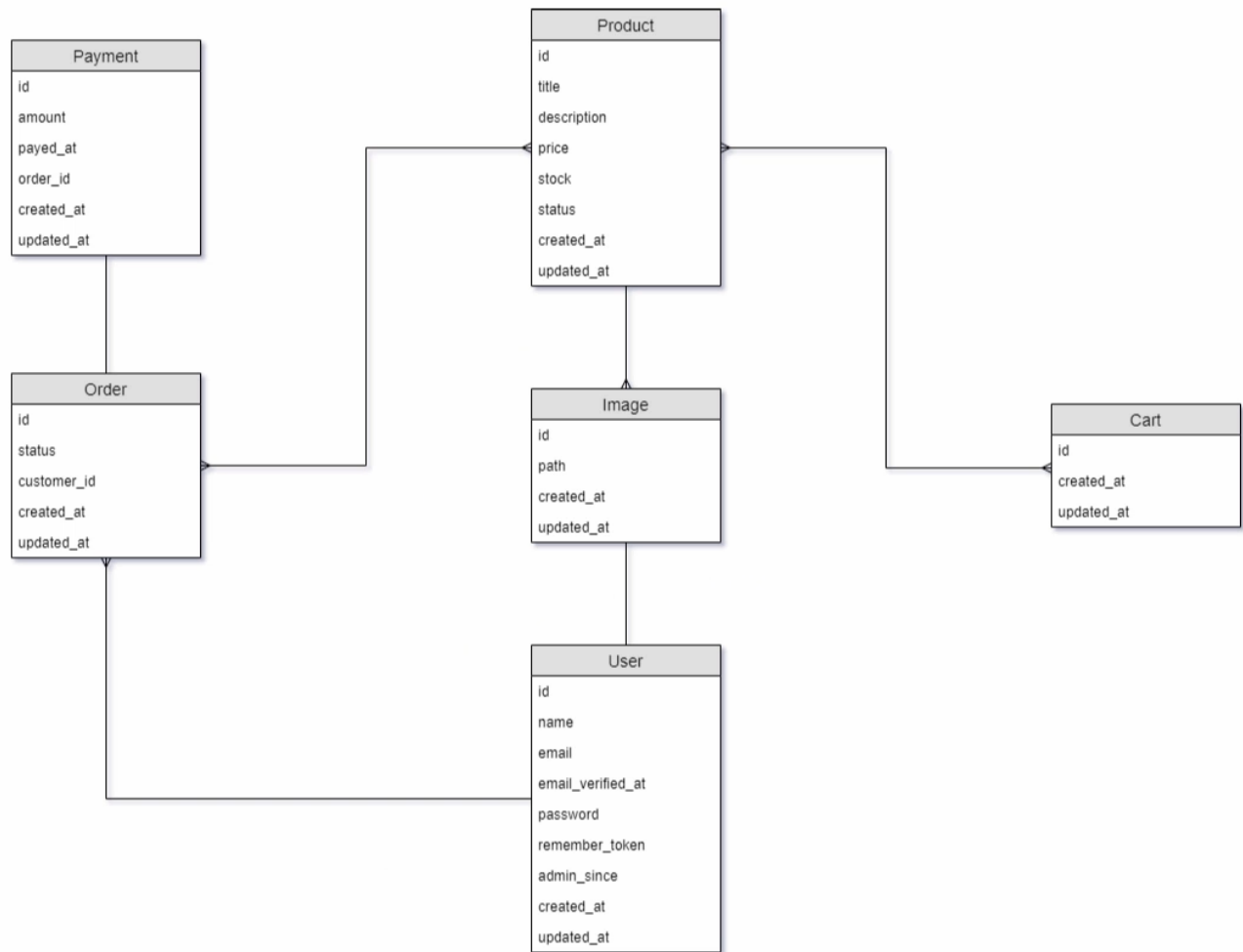


60. Domina las relaciones polimórficas uno a uno

Definición

Podemos ver en el diagrama de clases que una **Image** puede pertenecer a un solo **User** o un solo **Product**.



Podríamos pensar que **Image** debe tener dos claves foráneas: una para **user_id** y otra para **product_id**, e ir definiendo cada una cuando nos sea necesario.

Sin embargo, las claves foráneas no pueden ser null, y deberíamos tener dos relaciones casi iguales.

Esto se arregla incluyendo en **Image** un atributo llamado **imageable_id** y otro llamado **imageable_type**. Este último se encarga de definir el tipo de elemento que es **imageable** (en este caso, tanto **User** como **Product**).

② ¿En qué modelo implemento la relación polimórfica?

Como la ambigüedad está del lado de la **Image** (que puede pertenecer a un **Product** o a un **User**) es en ese modelo donde se tiene que implementar la relación polimórfica, para "pivotar" entre ambos. Esto será un método llamado

```
public function imageable() {  
    ...  
}
```

implementado en la clase ambigua **Image**.

Relaciones polimórficas uno a uno en el proyecto

- Image ----- User.

Laravel: crear las relaciones

En la migración de **Image**:

```
public function up()  
{  
    Schema::create('images', function (Blueprint $table) {  
        $table->id();  
        $table->string('path');  
        $table->timestamps();  
        // Método de Blueprint.php, descripción:  
        // Add the proper columns for a polymorphic table  
        // Añade el []_type y el []_id (siendo aquí [] = imageable).  
        $table->morphs('imageable');  
    });  
}
```

Ahora ejecutamos `php artisan migrate:fresh --seed`, para actualizar.

En **User**:

```
public function image() {  
    return $this->morphOne(Image::class, 'imageable');  
}
```

③ ¿Cómo se realiza esta consulta?

Laravel, por medio de **Eloquent**, va a la tabla de **Image** y busca un elemento cuyo `imageable_type` sea **User** y cuyo `imageable_id` sea el `id` del usuario.

⚠ Importar la clase Image

```
use App\Models\Image;
```

```
class User extends Authenticatable {  
    ...  
}
```

En **Image**:

```
public function imageable() {  
    // No podemos indicar tipo de elemento porque no lo sabemos:  
    // la función morphTo() se encarga de ello automáticamente.  
    return $this->morphTo();  
}
```

Tinker: probar las relaciones

Bastante autoexplicativo.

Creamos un **User**, una **Image** a partir del método de ese user, la guardamos en una variable y accedemos a su tipo de **Model** gracias al método `imageable()`.

```
C:\Users\llemi\Desktop\Lunk\Laravel\ola>php artisan tinker  
Psy Shell v0.11.8 (PHP 8.0.9 - cli) by Justin Hileman  
>>> $user = App\Models\User::factory()->create()  
=> App\Models\User {#4582  
    name: "Orlo Haag Jr.",  
    email: "shanny18@example.com",  
    email_verified_at: "2022-11-12 11:29:48",  
    #password: "$2y$10$92IXUNpkj00r0Q5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi",  
    #remember_token: "Q52pLFuznr",  
    admin_since: "now()",  
    updated_at: "2022-11-12 11:29:48",  
    created_at: "2022-11-12 11:29:48",  
    id: 1,  
}
```

```
>>> $user->image()->save(App\Models\Image::factory()->make())  
=> App\Models\Image {#4596  
    path: "img/products/5.jpg",  
    imageable_id: 1,  
    imageable_type: "App\Models\User",  
    updated_at: "2022-11-12 11:30:40",  
    created_at: "2022-11-12 11:30:40",  
    id: 1,  
}
```

```
>>> $user->image  
=> App\Models\Image {#4589  
    id: "1",  
    path: "img/products/5.jpg",  
    created_at: "2022-11-12 11:30:40",  
    updated_at: "2022-11-12 11:30:40",  
    imageable_type: "App\Models\User",  
    imageable_id: "1",  
}
```

```
>>> $image = $user->image  
=> App\Models\Image {#4589  
    id: "1",  
    path: "img/products/5.jpg",  
    created_at: "2022-11-12 11:30:40",
```

```
    updated_at: "2022-11-12 11:30:40",
    imageable_type: "App\Models\User",
    imageable_id: "1",
  }

>>> // Llamamos a $image→imageable sin indicar el tipo:
>>> // automáticamente se detectará por morphTo()
>>> $image→imageable
⇒ App\Models\User {#4545
  id: "1",
  name: "Orlo Haag Jr.",
  email: "shanny18@example.com",
  email_verified_at: "2022-11-12 11:29:48",
  #password: "$2y$10$92IXUNpkj00r0Q5byMi.Ye4oKoEa3Ro9llC/.og/at2.uhewG/igi",
  admin_since: "now()",
  #remember_token: "Q52pLFuznr",
  created_at: "2022-11-12 11:29:48",
  updated_at: "2022-11-12 11:29:48",
}
```