

58. Relaciones muchos a muchos

Definición

Las relaciones de **muchos a muchos** o **m:n** se dan cuando

Relaciones m:n en el proyecto

- Product ----- Cart.
- Product ----- Order.

Laravel: crear las relaciones

Usando `php artisan tinker` creamos una nueva tabla.

Debe llamarse, por ejemplo para el caso de la relación Product-Cart:

```
php artisan make:migration CreateCartProductTable
```

⚠ Nombre de la tabla

Es necesario colocar los nombres de los modelos ordenados **alfabéticamente** (Cart > Product).

Tinker: probar las relaciones

Creación de las instancias

Creamos tanto un **Product** como un **User** y los guardamos en variables para trabajar con ellos.

```
Microsoft Windows [Versión 10.0.19043.2130]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\llemi>cd C:/Users/llemi/Desktop/Lunk/Laravel/ola

C:\Users\llemi\Desktop\Lunk\Laravel\ola>php artisan tinker
Psy Shell v0.11.8 (PHP 8.0.9 - cli) by Justin Hileman
>>> $product = App\Models\Product::factory()->create()
=> App\Models\Product {#4582
    title: "Culpa incidunt.",
    description: "Architecto iure et neque error.",
    price: 76.76,
    stock: 8,
    status: "unavailable",
    updated_at: "2022-11-11 09:36:17",
    created_at: "2022-11-11 09:36:17",
    id: 53,
}

>>> $user = App\Models\User::factory()->create()
=> App\Models\User {#4585
    name: "Prof. Selena Baumbach",
    email: "burdette33@example.org",
    email_verified_at: "2022-11-11 09:36:27",
    #password: "$2y$10$92IXUNpkj00r0Q5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi",
```

```
#remember_token: "ql3WSuyONX",
admin_since: "now()",
updated_at: "2022-11-11 09:36:27",
created_at: "2022-11-11 09:36:27",
id: 2,
}
```

Creamos una **Order** añadiendo como `customer_id` el id del `$user` que acabamos de crear.

```
>>> $order = App\Models\Order::factory()->create(['customer_id'=> $user->id])
=> App\Models\Order {#4598
  status: "paid",
  customer_id: 2,
  updated_at: "2022-11-11 09:36:34",
  created_at: "2022-11-11 09:36:34",
  id: 2,
}
```

⚠ Segundo parámetro en Model

Importante tener definido el segundo parámetro en **ambos modelos** si no hemos respetado la consistencia en la nomenclatura de las columnas.

En App\Models\Order:

```
...
public function user() {
    return $this->belongsTo(User::class, 'customer_id');
}
...
```

En App\Models\User:

```
...
public function orders() {
    return $this->hasMany(Order::class, 'customer_id');
}
...
```

Intento de acceder a sus componentes

Creamos un **Cart** e intentamos acceder a los productos que contiene (naturalmente aparecerá vacío, ya que no hemos asignado nada).

Lo mismo ocurre con los productos en nuestra **Order**.

```
>>> $cart = App\Models\Cart::factory()->create()
=> App\Models\Cart {#4551
  updated_at: "2022-11-11 09:36:43",
  created_at: "2022-11-11 09:36:43",
  id: 3,
}

>>> $cart->products
=> Illuminate\Database\Eloquent\Collection {#4547
  all: [],
```

```

    }

    >>> $order->products
    => Illuminate\Database\Eloquent\Collection {#4573
        all: [],
    }

```

Uso de attach()

Explicar attach() aquí.

```

    >>> $order->products()->attach([1 => ['quantity' => 4], 2 => ['quantity' => 3], 3 =>
    ['quantity' => 6]])
    => null

```

fresh()

El método `fresh()` actualiza la instancia **desde la base de datos**, no desde `php artisan tinker`. Esta asignación hace que cualquier dato que hayamos modificado se guarde en la propia instancia `$order`, ahora actualizada.

```

    >>> $order = $order->fresh()
    => App\Models\Order {#4550
        id: "2",
        status: "paid",
        customer_id: "2",
        created_at: "2022-11-11 09:36:34",
        updated_at: "2022-11-11 09:36:34",
    }

```

⚠ Importante

Es **muy importante ejecutar este método** o recibiremos un `null` como resultado a la ejecución de `$order->products`, aun cuando sepamos que hemos realizado el `attach()` correctamente.

Acceder a los componentes correctamente

```

    >>> $order->products
    => Illuminate\Database\Eloquent\Collection {#4546
        all: [
            App\Models\Product {#4583
                id: "1",
                title: "Ea ipsam quae excepturi.",
                description: "Incidunt ipsa id sunt sunt non. Pariatur est omnis praesentium molestiae.",
                price: "18.92",
                stock: "3",
                status: "available",
                created_at: "2022-11-11 07:59:39",
                updated_at: "2022-11-11 07:59:39",
                pivot: Illuminate\Database\Eloquent\Relations\Pivot {#4597
                    order_id: "2",
                    product_id: "1",
                    quantity: "4",
                },
            },
        ],
    }

```

```
App\Models\Product {#4580
  id: "2",
  title: "Sunt iusto vitae.",
  description: "Inventore ea ut sed.",
  price: "78.18",
  stock: "3",
  status: "available",
  created_at: "2022-11-11 07:59:39",
  updated_at: "2022-11-11 07:59:39",
  pivot: Illuminate\Database\Eloquent\Relations\Pivot {#4577
    order_id: "2",
    product_id: "2",
    quantity: "3",
  },
},
App\Models\Product {#4604
  id: "3",
  title: "Eius est assumenda asperiores.",
  description: "Quae qui natus aperiam iusto odio magnam ducimus. Autem
necessitatibus consequatur officia aperiam quam sit.",
  price: "47.62",
  stock: "8",
  status: "available",
  created_at: "2022-11-11 07:59:39",
  updated_at: "2022-11-11 07:59:39",
  pivot: Illuminate\Database\Eloquent\Relations\Pivot {#4574
    order_id: "2",
    product_id: "3",
    quantity: "6",
  },
},
],
}
```