Relations

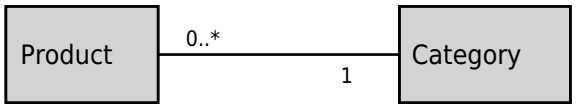
2 grands types de relations existent :

- manyToOne/oneToMany
- manyToMany

manyToOne/oneToMany

Correspond aux associations hiérarchiques de type CIF.
manyToOne et **oneToMany** correspondent au même type d'association, mais vus de chaque côté de la relation.

Création



CREATED WITH YUML

Exemple :

manyToOne

Chaque produit appartient à une catégorie :

```
// ...
class Product{
    // ...

    /**
     * @ORM\ManyToOne(targetEntity="App\Entity\Category", inversedBy="products")
     * @ORM\JoinColumn(nullable=true)
     */
    private $category;

    public function getCategory(): Category
    {
        return $this->category;
    }

    public function setCategory(Category $category)
    {
        $this->category = $category;
    }
}
```

oneToMany

Chaque catégorie peut contenir plusieurs produits :

```
// ...
use Doctrine\Common\Collections\ArrayCollection;
use Doctrine\Common\Collections\Collection;

class Category{
    // ...

    /**
     * @ORM\OneToMany(targetEntity="App\Entity\Product", mappedBy="category")
     */
    private $products;

    public function __construct(){
        $this->products = new ArrayCollection();
    }

    /**
     * @return Collection|Product[]
     */
    public function getProducts(){
        return $this->products;
    }
}
```

Chargement

manyToOne : Récupération de la catégorie d'un produit :

```
use App\Entity\Product;
// ...

/**
 * @Route("/product/{id}", name="product_show")
 */
public function showAction(Product $product){

    $categoryName = $product->getCategory()->getName();

    // ...
}
```

Persistance

Sauvegarde d'un produit et de sa catégorie

```
// ...

use App\Entity\Category;
use App\Entity\Product;
use Symfony\Component\HttpFoundation\Response;

class ProductController extends Controller{
    /**
     * @Route("/product", name="product")
     */
    public function index(){
        $category = new Category();
        $category->setName('Computer Peripherals');

        $product = new Product();
        $product->setName('Keyboard');
        $product->setPrice(19.99);
        $product->setDescription('Ergonomic and stylish!');

        // relates this product to the category
        $product->setCategory($category);

        $em = $this->getDoctrine()->getManager();
        $em->persist($category);
        $em->persist($product);
        $em->flush();

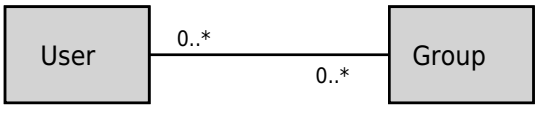
        return new Response(
            'Saved new product with id: '.$product->getId()
            .' and new category with id: '.$category->getId()
        );
    }
}
```

ManyToMany

Correspond aux associations non hiérarchiques de type CIM, non porteuses de données.

Création

Exemple :



CREATED WITH YUML

```
<?php
/** @Entity */
class User
```

```
{
    // ...

    /**
     * Each User has Many Groups.
     * @ManyToMany(targetEntity="Group", inversedBy="users")
     * @JoinTable(name="users_groups")
     */
    private $groups;

    public function __construct() {
        $this->groups = new \Doctrine\Common\Collections\ArrayCollection();
    }

    // ...
}
```

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/framework-web/symfony/models/relations

Last update: **2019/08/31 14:21**