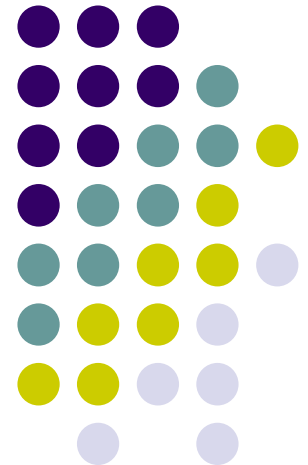
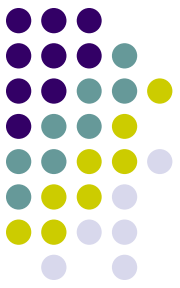




**PHP**

**Les associations**





# ONE-TO-MANY

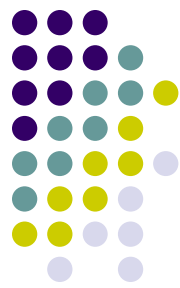


## One-To-Many / Many-To-One

>> bidirectionnelle

Many-To-One





## One-To-Many / Many-To-One

>> bidirectionnelle

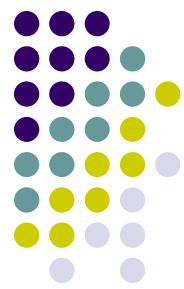


```
class Post
{
    #[ORM\Id]
    #[ORM\Column(type: 'integer')]
    #[ORM\GeneratedValue]
    private ?int $id;
    #[ORM\Column(type: 'string', length: 255)]
    private ?string $titre;
    #[ORM\Column(type: 'text')]
    private ?string $description;

    private array $commentaires;
```

```
class Commentaire
{
    #[ORM\Column(type: 'integer')]
    #[ORM\Id]
    #[ORM\GeneratedValue]
    private int $id;
    #[ORM\Column(type: "text")]
    private string $contenu;

    private Post $post;
```



## One-To-Many / Many-To-One

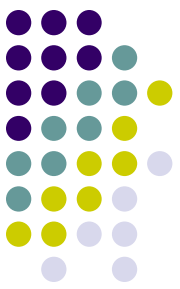
>> **bidirectionnelle**



**Ajouter des attributs (`#[...]`) afin d'indiquer à **Doctrine** comment gérer l'association bidirectionnelle**

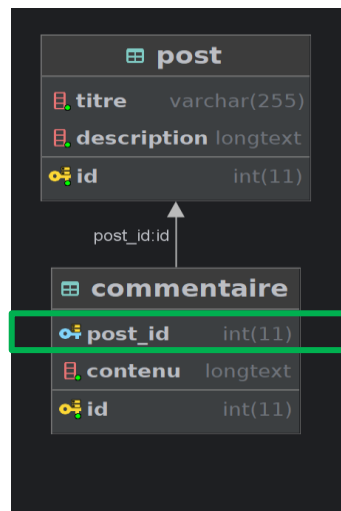
NEED TO KNOW

**DOCTRINE A BESOIN DE CONNAITRE  
LE COTE PROPRIETAIRE DE  
L'ASSOCIATION BIDIRECTIONNELLE**



## One-To-Many / Many-To-One

>> bidirectionnelle



**C'est "moi"  
qui possède la  
clé étrangère**



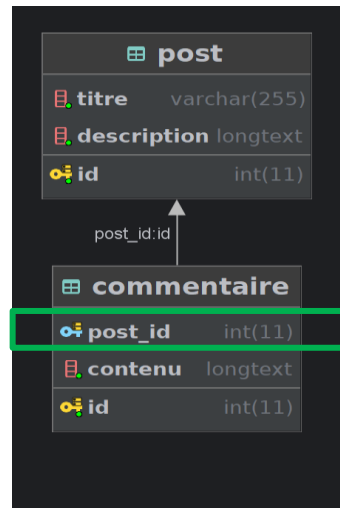


## One-To-Many / Many-To-One

>> bidirectionnelle



**Côté  
inverse**



C'est "moi" qui possède la  
clé étrangère



**Côté  
propriétaire**



## One-To-Many / Many-To-One

>> bidirectionnelle

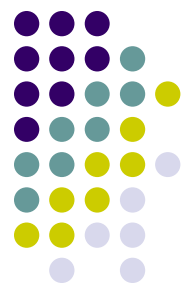


**POUR  
DOCTRINE**



La classe **Commentaire** est responsable  
de la synchronisation de l'association  
dans la base de données





## One-To-Many / Many-To-One

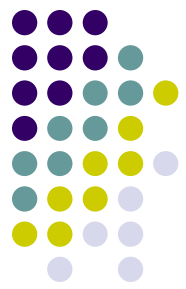
>> bidirectionnelle



**Le côté **propriétaire** est celui qui "gère" l'association**



**Chaque fois qu'on associe un **objet "propriétaire"** à un **objet "inverse"**, **Doctrine** saura **synchroniser** cette association dans la base de données**



## One-To-Many / Many-To-One

>> bidirectionnelle

**\$proprietaire->setInverse(\$inverse)**



On associe un **commentaire** à un **post**

**\$commentaire->setPost(\$post)**

Côté propriétaire



Côté inverse



Création du lien





## One-To-Many / Many-To-One

>> bidirectionnelle



```
class Post
{
    #[ORM\Id]
    #[ORM\Column(type: 'integer')]
    #[ORM\GeneratedValue]
    private ?int $id;
    #[ORM\Column(type: 'string', length: 255)]
    private ?string $titre;
    #[ORM\Column(type: 'text')]
    private ?string $description;

    #[ORM\OneToMany(targetEntity: Commentaire::class,
        mappedBy: 'post')]
    private Collection $commentaires;
}
```

**ONE-TO-MANY**

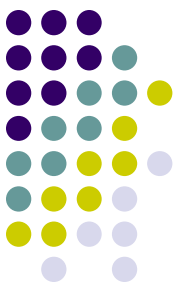
```
class Commentaire
{
    #[ORM\Column(type: 'integer')]
    #[ORM\Id]
    #[ORM\GeneratedValue]
    private int $id;

    #[ORM\Column(type: "text")]
    private string $contenu;

    #[ORM\ManyToOne(targetEntity: Post::class,
        inversedBy: 'commentaires')]
    private Post $post;
}
```

**MANY-TO-ONE**



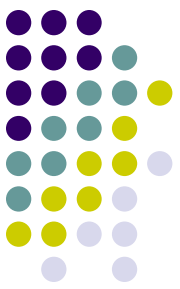


## One-To-Many / Many-To-One

>> bidirectionnelle



**C'est toujours le côté **MANY**  
qui est le côté **propriétaire**  
car il détient la **clé étrangère****

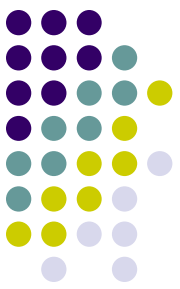


**One-To-Many / Many-To-One**

**>> bidirectionnelle**

**A vos codes !**

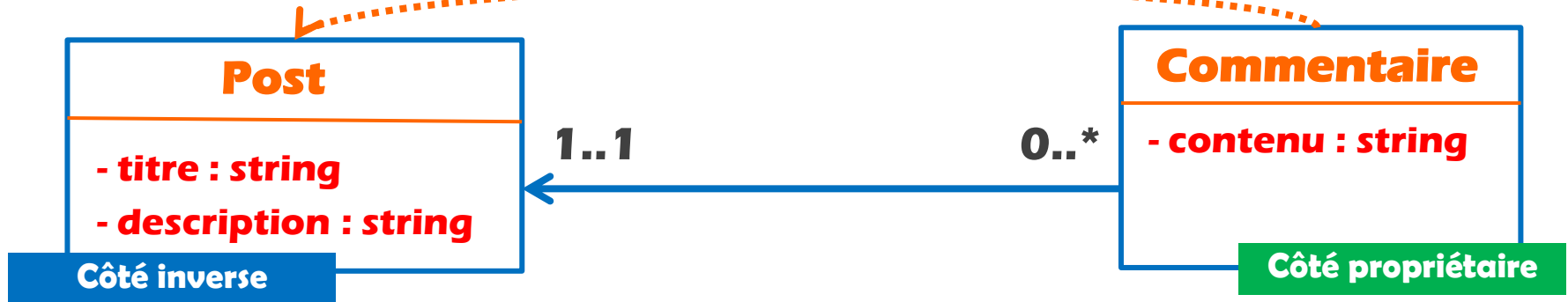




## One-To-Many / Many-To-One

>> unidirectionnelle

Many-To-One



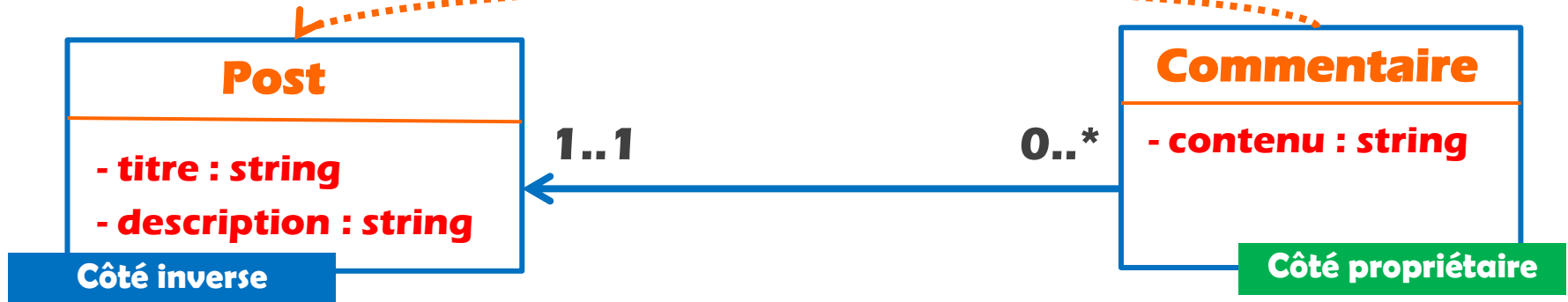
**C'est moi qui possède la clé étrangère**



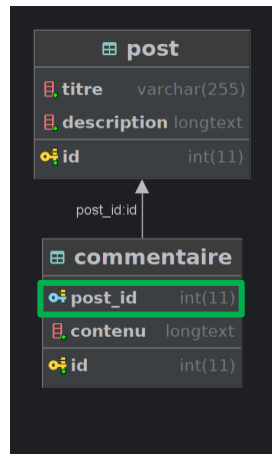
## One-To-Many / Many-To-One

>> unidirectionnelle

Many-To-One



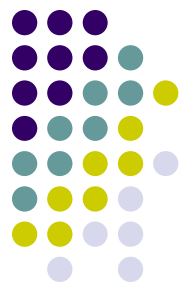
```
class Post
{
    #[ORM\Id]
    #[ORM\Column(type: 'integer')]
    #[ORM\GeneratedValue]
    private ?int $id;
    #[ORM\Column(type: 'string', length: 255)]
    private ?string $titre;
    #[ORM\Column(type: 'text')]
    private ?string $description;
}
```



```
class Commentaire
{
    #[ORM\Column(type: 'integer')]
    #[ORM\Id]
    #[ORM\GeneratedValue]
    private int $id;

    #[ORM\Column(type: "text")]
    private string $contenu;

    #[ORM\ManyToOne(targetEntity: Post::class)]
    private Post $post;
}
```



**One-To-Many / Many-To-One**

**>> unidirectionnelle**

# A vos codes !







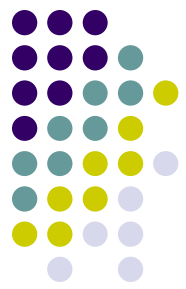
## One-To-Many / Many-To-One

>> unidirectionnelle

### One-To-Many



C'est moi qui possède  
la **clé étrangère**



## One-To-Many / Many-To-One

>> unidirectionnelle

### One-To-Many



```
class Post
{
    #[ORM\Id]
    #[ORM\Column(type: 'integer')]
    #[ORM\GeneratedValue]
    private ?int $id;
    #[ORM\Column(type: 'string', length: 255)]
    private ?string $titre;
    #[ORM\Column(type: 'text')]
    private ?string $description;

    #[ORM\OneToMany(targetEntity: Commentaire::class)]
    private Collection $commentaires;
}
```

**ONE-TO-MANY**

?

```
class Commentaire
{
    #[ORM\Column(type: 'integer')]
    #[ORM\Id]
    #[ORM\GeneratedValue]
    private int $id;

    #[ORM\Column(type: "text")]
    private string $contenu;
}
```



## One-To-Many / Many-To-One

>> unidirectionnelle

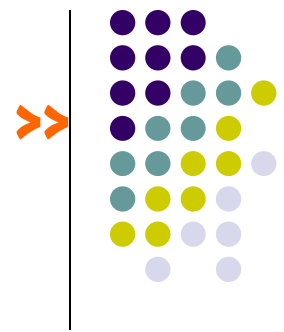
### One-To-Many



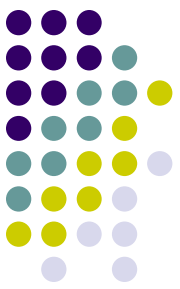
**Doctrine** ne sait pas  
comment gérer  
correctement  
l'association dans la BD



C'est moi qui  
possède la **clé**  
**étrangère**



**MANY-TO-MANY**



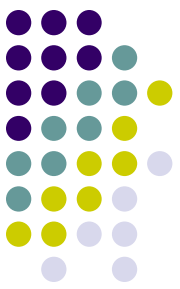
**Many-To-Many**

**>> bidirectionnelle**



**NEED TO KNOW**

**DOCTRINE A BESOIN DE CONNAITRE  
LE COTE PROPRIETAIRE DE  
L'ASSOCIATION BIDIRECTIONNELLE**



**Many-To-Many**

**>> bidirectionnelle**



**C'est vous qui décidez !**

**I'M THE BOSS**



**Many-To-Many**

**>> bidirectionnelle**



**La classe **Etudiant** est responsable de la synchronisation de l'association dans la base de données**



**Many-To-Many**

**>> bidirectionnelle**

**\$proprietaire->ajouterInverse(\$inverse)**



**On associe un étudiant à un cours**

**\$etudiant->ajouterCours(\$cours)**

**Côté propriétaire**

**Côté inverse**



**Création du lien**







**Many-To-Many**

**>> bidirectionnelle**

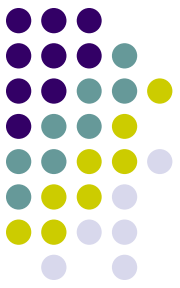


**Comment définir le mapping**

?



**Lit la documentation et la solution tu trouveras !**



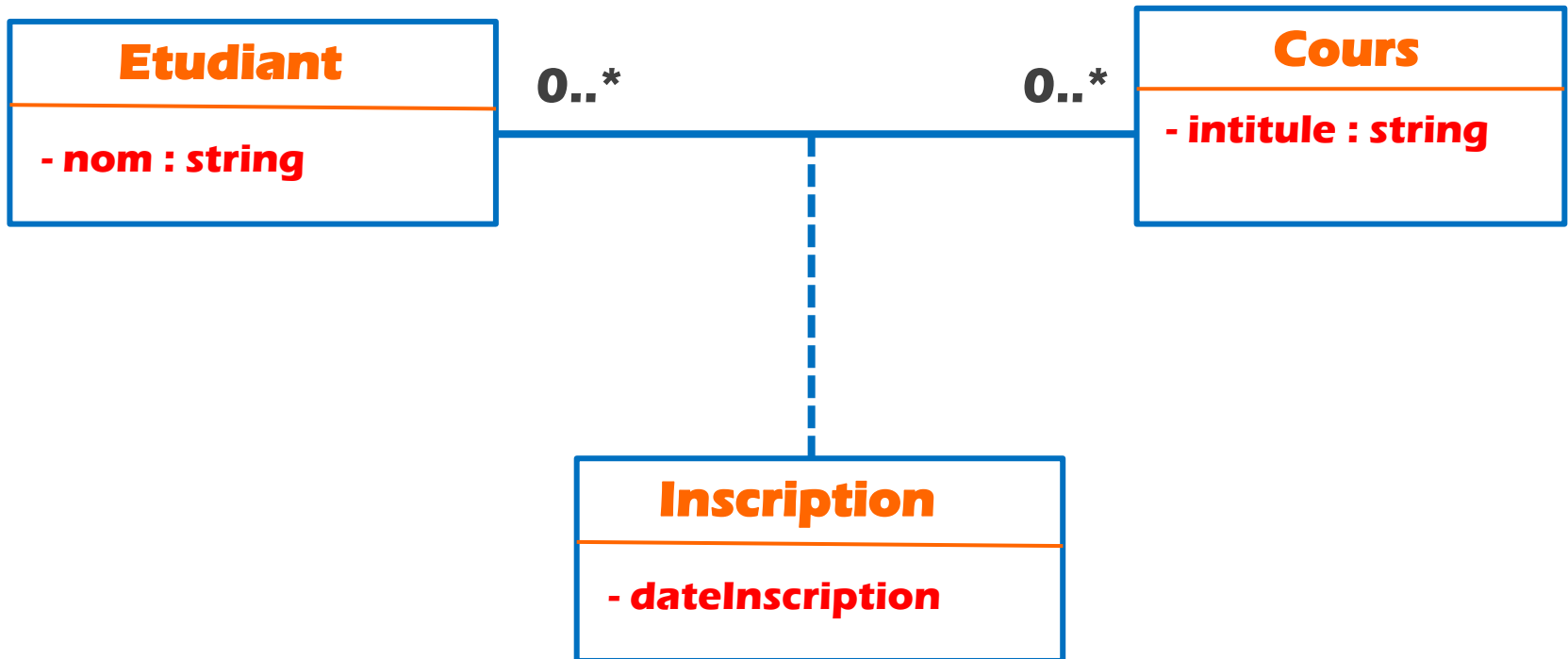
# MANY-TO-MANY

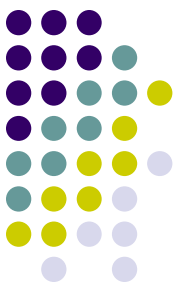
**Porteuse de données**



## Many-To-Many

>> Porteuse de données





**Many-To-Many**

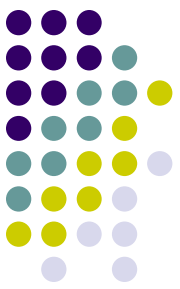
**>> Porteuse de données**



**DOCTRINE NE SAIT PAS  
GERER CE GENRE  
D'ASSOCIATION**

**MAIS...**

**PAS DE  
PANIQUE!**



## Many-To-Many

>> Porteuse de données

