

modifier l'annotation de clé primaire

```
@ORM\GeneratedValue(strategy="NONE")
```

1/b

l'entity manager est l'interface entre le code et la base de données. Il effectue des opérations de persistance, de récupération : exemples `getRepository()` et `persist()`

2-a

les étapes pour la relation :

commande : `php bin/console create:entity Film`

(pour modifier la table existante)

ajouter la propriété `category`

on choisit la relation `OneToMany`.

indiquer la classe `Category` pour la relation

et indiquer oui pour créer l'attribut `films` dans la classe `Category`

2-b

Dans la classe `Film` :

```
/**
 * @ORM\ManyToOne(targetEntity="App\Entity\Category", inversedBy="films")
 * @ORM\JoinColumn(name="category_id", referencedColumnName="id")
 */
private $category;
```

2)b)

[1]

```
ManyToOne(targetEntity: Category::class, mappedBy: 'films')
```

[2]

```
private $category;
```

[3]

```
OneToMany(targetEntity: Film::class)
```

[4]

```
private Collection $films;
```

3-

```
#[Route('/nvFilm', name: nvFilm)]
public function addFilm(Request $request , FilmRepository
$filmRepository)
{
    $film = new Film();

    $form = $this->createForm(FilmType::class, $film);

    $form->handleRequest($request);

    if ($form->isSubmitted() && $form->isValid()) {
        $film->setViews(0);
        $filmRepository->save($film,true);

        return $this->redirectToRoute('film_list');
    }

    return $this->render('film/add.html.twig', [
        'form' => $form->createView(),
    ]);
}
```

4- attribut de type Category cannot be converted to string car le form add le categorie sans type , passe string par default qui il nest pas etre categorye

-b

i faut indiquer le type EntityType

```
->add(category, EntityType::class, [
    'class' => Category::class,
    'choice_label' => name,
]);
```

5.a

```
#[Route('/getFilm/{id}', name: 'getFilm')]
public function listFilms(FilmRepository $filmRepository)
{
    $films = $filmRepository->findAll();

    return $this->render('film/list.html.twig', [
        'films' => $films,
    ]);
}
```

5-b

```
#[Route('/films', name: films)]
```

(5-b

```
[1] : #[Route('/getFilm/{id}', name: 'getFilm')]
public function showFilm(ManagerRegistry $MR, FilmRepository $repo, $id): Response
{
    [2] : $entityManager = $MR->getManager();
        $film = $repo->find($id);

    if ($film) {
        $film->setViews($film->getViews() + 1);
        $entityManager->flush();
    } // [end 2] ////

    return $this->render('film/show.html.twig', [
        'film' => $film,
    ]);
}
```

5)c)      return \$this->redirectToRoute('film\_list');

7-

```
public function getFilmByCat($id)
{
    return $this->createQueryBuilder('f')
        ->leftJoin('f.category', 'c')
        ->andWhere('c.id = :categoryId')
        ->setParameter('categoryId', $id)
        ->getQuery()
        ->getResult();
}
```

8-

```
public function getFilmsnbViews()
{
    return $this->createQueryBuilder('f')
        ->where('f.views > :views')
        ->setParameter('views', 100)
        ->getQuery()
        ->getResult();
}
```