

Partie 5 : Relation entre les Entités

Soit de diagramme de classe suivant :



On se propose de créer les deux entités Voiture et Marque et d'ajouter la relation entre ces deux entités.

1. Créer les entités « Marque » et « voiture »
2. Editez le fichier Voiture.php, puis ajoutez l'attribut marque à la classe Voiture et l'associer avec l'entité Marque:

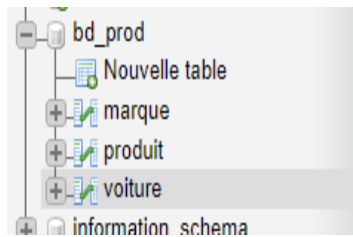
```
/** * @ORM\ManyToOne(targetEntity="Marque") */
private $marque;
```

3. Générez les getters et les setters pour l'attribut « marque » (bouton droit/insert code)

```
public function getMarque()
{
    return $this->marque;
}
public function setMarque($marque) {
    $this->marque = $marque;
}
```

4. Générer, avec Doctrine, les tables Marque et Voiture à partir des entités en tapant : `php bin/console doctrine:schema:update --force --dump-sql`

Vérifier la création des tables dans la base de données BD_PROD



#	Nom	Type	Interclassement	Attributs	Null	Valeur par
1	id	int(11)			Non	Aucun(e)
2	marque_id	int(11)			Oui	NULL
3	numSerie	varchar(50) utf8_unicode_ci			Non	Aucun(e)
4	Date_Mise_Circu	datetime			Non	Aucun(e)

Créer un nouveau Controller : VoitureController.php.

Créer la méthode addMarqueAction(\$nom) qui permet d'ajouter une nouvelle marque.

Le contenu du fichier VoitureController.php est le suivant :

```
class VoitureController extends Controller
{
    /**
     * @Route("/addMarque/{nom}")
     */
    public function addMarqueAction($nom)
    {
        $m= new Marque();
        $m->setNomMarque($nom);
        $em=$this->getDoctrine()->getManager();
        $em->persist($m);
        $em->flush();
        return $this->render('VoitureBundle:Default:addMarque.html.twig',array('marque' => $m));
    }
}
```

- Vérifier l'insertion de la marque dans la table Marque,
- Ajouter au moins 3 marques de votre choix,

Créer un formulaire pour ajouter des voitures

```
namespace VoitureBundle\Controller;
use VoitureBundle\Entity\Marque;
use VoitureBundle\Entity\Voiture;
use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route;
use Symfony\Component\Form\Extension\Core\Type\TextType;
use Symfony\Component\Form\Extension\Core\Type\DateType;
use Symfony\Bridge\Doctrine\Form\Type\EntityType;
use Symfony\Component\Form\Extension\Core\Type\SubmitType;
use Symfony\Component\HttpFoundation\Request;
```

```

/** * @Route("/addVoiture/") */
public function addVoitureAction(Request $request)
{
    $v=new Voiture();
    //générer le formulaire
    $form=$this->createFormBuilder($v)
        ->add('numSerie',TextType::class)
        ->add('dateMiseCircu',DateType::class)
        ->add('marque',EntityType::class,array ('class' => 'VoitureBundle:Marque','choice_label'
        ->add('Add',SubmitType::class)
        ->getForm();
    $form->handleRequest($request);
    //tester si le formulaire est valide
    if($form->isValid()) {
        $em=$this->getDoctrine()->getManager();
        $em->persist($v);
        $em->flush();
    }
    return $this->render('VoitureBundle:Default:formvoiture.html.twig',array('f' => $form->createView()));
}

```

Ajouter, au Contrôleur VoitureController.php, la méthode listeVoitureAction qui affiche la liste des voitures,

```

/** * @Route("/listeVoitures", name="listeV") */
public function listeVoitureAction() {
    $voitures = $this->getDoctrine()->getRepository("VoitureBundle:Voiture")>findAll();

    return $this->render('VoitureBundle:Default:listevoiture.html.twig',array('voitures' => $voitures));
}

```

Ajouter la vue listevoiture.html.twig dans le dossier :

← → ↻ ⓘ localhost/my_project_name/web/app_dev.php/listeVoitures

ID	NOM	PRIX
12345	Audi	01/03/2013

Modifier l'action addVoitureAction et la vue formvoiture.html.twig pour afficher dans la même page le formulaire et la liste des voitures

```

/** * @Route("/addVoiture/") */
public function addVoitureAction(Request $request) {
    $v=new Voiture();
    //générer le formulaire
    $form=$this->createFormBuilder($v)

->add('numSerie',TextType::class)
->add('dateMiseCircu',DateType::class)
->add('marque',EntityType::class,array ('class' => 'VoitureBundle:Marque',
'choice_label' => 'nomMarque','choice_value' =>'id'))
->add('Add',SubmitType::class)->getForm();
    $form->handleRequest($request);
    //tester si le formulaire est valide
    if($form->isValid()) {
        $em=$this->getDoctrine()->getManager();
        $em->persist($v);
        $em->flush();
    }
    $voitures = $this->getDoctrine()->getRepository("VoitureBundle:Voiture")->findAll();
    return $this->render('VoitureBundle:Default:formvoiture.html.twig',array('f' => $form->create
    'voitures'=> $voitures ));
}

```

```

1  {% extends '::base.html.twig' %} {% block body %}
2  {{ form_start(f) }}
3  <table>
4      <tr> <td>{{form_label(f.numSerie)}}</td>
5          <td>{{form_widget(f.numSerie)}}</td>
6          <td>{{form_errors(f.numSerie)}}</td>
7      </tr>
8      <tr>
9          <td>{{form_label(f.dateMiseCircu)}}</td>
10         <td>{{form_widget(f.dateMiseCircu)}}</td>
11         <td>{{form_errors(f.dateMiseCircu)}}</td>
12     </tr>
13     <tr>
14         <td>{{form_label(f.marque)}}</td>
15         <td>{{form_widget(f.marque)}}</td>
16         <td>{{form_errors(f.marque)}}</td>
17     </tr>
18 </table>
19 {{ form_end(f) }}
20 <table border="1">
21     <tr>
22         <th>ID </th> <th>NOM </th> <th>PRIX </th>
23     </tr>
24     {% for v in voitures %}
25     <tr>
26         <td>{{ v.numSerie }}</td>
27         <td>{{ v.marque.nomMarque }}</td>
28         <td>{{ v.dateMiseCircu|date("d/m/Y") }}</td>
29     </tr> {% endfor %} </table> {% endblock %}

```