Master ICONE 1ère année


# Base de données avancées
# *ORM - Doctrine*

*Réaliser par :*
Mohammed BENAOU

## Liste des exercices

# 1 Introduction

Après la création du dossier qui contient notre projet il faut créer aussi un fichier .json avec le contenu suivant :

```
{
    "require": {
        "doctrine/orm": "2.4.*",
        "symfony/yaml": "2.*"
    },
    "autoload": {
        "psr-0": {"": "src/"}
    }
}
```

maintenant on doit installer Doctrine en utilisant la commande *composer install* cette commande va installer des packages Doctrine tel que Doctrine DBAL, Doctrine ORM, Symfony YAML and Symfony Console dans le dossier *Vendor*. Pour générer le schéma de la base de données il faut ajouter ce bout de code dans le fichier*cli-config.php*

```
<?php
// cli-config.php
require_once "bootstrap.php";
return \Doctrine\ORM\Tools\Console\ConsoleRunner::createHelperSet(
    $entityManager);
```

par la suite on fait appel à la commande *vendor/bin/doctrine orm :schema-tool :update –force*

# 2 Exercices

## 2.1 Exercice 1

La commande bin/person-create permettant de créer une personne :

```
#!/usr/bin/php -Cq
<?php
require_once "bootstrap.php";

$personne = new Personne();
$first=$argv[1]; // le premier argument
$last=$argv[2]; // le deuxième argument

$personne->setFirstName($first);
$personne->setLastName($last);

$entityManager->persist($personne);
$entityManager->flush();

echo "Onsuccess : ".$first." ".$last." est ajouté \n";
echo "          => Id: " . $personne->getId() . "\n";
```

```
[MacBook-Pro-de-Mohammed:TP_Doctrine_MB Mohammed$ bin/person-create Mohammed BENAOU  ]
Onsuccess : Mohammed BENAOU est ajouté
          => Id: 10
MacBook-Pro-de-Mohammed:TP_Doctrine_MB Mohammed$
```

## 2.2 Exercice 2

La commande bin/person-remove permettant d'effacer une personne :

```php
#!/usr/bin/php -Cq
<?php

require_once "bootstrap.php";

$PersonID = $argv[1];
$person = $entityManager->find("Personne", $PersonID);

if ($person) {
    $entityManager->remove($person);
    $entityManager->flush();
    echo "Onsuccess : " . $person->getFirstName() . " " . $person->
    getLastName() . " est supprimé "."\n";
}
```

```
[MacBook-Pro-de-Mohammed:TP_Doctrine_MB Mohammed$ bin/person-remove 3                ]
Onsuccess : Momo ben est supprimé
```

## 2.3 Exercice 3

La commande person-list permettant de lister l'ensemble des personnes :

```php
#!/usr/bin/php -Cq
<?php

require_once "bootstrap.php";

$personReq = $entityManager->getRepository("Personne");

$persons = $personReq->findAll();
    echo " ----------------------------------------\n";
    echo "|  ID   |    First name    |    Last name    |\n";
    echo "|-------|------------------|-----------------|\n";

foreach ($persons as $i) {

    echo "    ".$i->getId()."          ".$i->getFirstName()."
       ".$i->getLastName()."\n";

    echo "|-------|------------------|-----------------|\n";
}
```

```
[MacBook-Pro-de-Mohammed:TP_Doctrine_MB Mohammed$ bin/person-list
 ----------------------------------------
|  ID   |   First name   |   Last name    |
|-------|----------------|----------------|
   6        momo             rasfa
|-------|----------------|----------------|
   7        Mohammed            benaou
|-------|----------------|----------------|
   8        Anas             ABDELLAOUI
|-------|----------------|----------------|
   9        Walid             Cherkkaoui
|-------|----------------|----------------|
   10        Mohammed             BENAOU
|-------|----------------|----------------|
```

## 2.4 Exercice 4

La commande bin/person-phone-add permettant d'ajouter un numéro de téléphone à une personne :

```php
#!/usr/bin/php -Cq
<?php

require_once "bootstrap.php";

$telephone = new Telephone;

$type = new Type;

$telephone->setNumero($argv[2]);
$type->setType($argv[3]);

$type->addPhone($telephone);
$telephone->setType($type);

$PersonID = $argv[1];

$personne = $entityManager->find("Personne", $PersonID);

$personne->addPhone($telephone);
$telephone->setPersonne($personne);

$entityManager->persist($personne);
$entityManager->persist($telephone);
$entityManager->persist($type);
$entityManager->flush();

echo"-Le type de numéro de téléphone : " . $telephone->getType()->
    getType() ."\n-Le numéro de téléphone : " . $telephone->
    getNumero() . "\n-Le propriétaire du némero : " . $personne->
    getFirstName() . " " . $personne->getLastName(). "\n";
```

```
[MacBook-Pro-de-Mohammed:TP_Doctrine_MB Mohammed$ bin/person-phone-add 10 0621652705 ]
personel
-Le type de numéro de téléphone : personel
-Le numéro de téléphone : 0621652705
-Le propriétaire du némero : Mohammed BENAOU
```

## 2.5 Exercice 5

La commande bin/phone-remove permettant d'effacer un numéro de téléphone :

```
1  #!/usr/bin/php -Cq
2  <?php
3
4  require_once 'bootstrap.php';
5
6  $num = $argv[1];
7
8
9  $phone = $entityManager->getRepository("Telephone")->findOneBy(
       array("numero" => $num));
10
11 if ($phone) {
12
13     $entityManager->remove($phone);
14     $entityManager->flush();
15     echo "Onsuccess \n";
16     echo "=> Le numéro de Télephone ". $phone->getNumero() . " est
       supprimé \n";
17 }
```

```
[MacBook-Pro-de-Mohammed:TP_Doctrine_MB Mohammed$ bin/person-phone-remove 0621652705
Onsuccess
=> Le numéro de Télephone 0621652705 est supprimé
```

## 2.6 Exercice 6

La commande bin/person-phone-list permettant de lister les numéros de téléphone associés à une personne :

```
1  #!/usr/bin/php -Cq
2  <?php
3
4  require_once "bootstrap.php";
5
6  $person = $entityManager->find("Personne", $argv[1]);
7
8  if ($person) {
9      echo "-La personne est : ".$person->getFirstName(). " " .
       $person->getLastName() . "\n";
10     foreach($person->getPhones() as $i) {
11         echo "-Le numéro de Téléphone : " . $i->getNumero() . "\n";
12         echo "-Le type de numéro de téléphone : " . $i->getType()->
       getType()."\n";
13     }
```

```
14  }
```

```
[MacBook-Pro-de-Mohammed:TP_Doctrine_MB Mohammed$ bin/person-phone-list 10
-La personne est : Mohammed BENAOU
-Le numéro de Téléphone : 0621652705
-Le type de numéro de téléphone : personel
```

## 2.7 Exercice 7

La commande bin/type-list permettant de lister les types de numéros de téléphone :

```
1  #!/usr/bin/php -Cq
2  <?php
3
4  require_once "bootstrap.php";
5
6  $typeRepo = $entityManager->getRepository("Type");
7  $types = $typeRepo->findAll();
8      echo " ------------------------\n";
9      echo "|  ID   |      Type      |\n";
10     echo "|-------|----------------|\n";
11 foreach ($types as $i) {
12     echo "    " .$i->getId() . "        " . $i->getType() . "\n";
13     echo "|-------|----------------|\n";
14 }
```

```
[MacBook-Pro-de-Mohammed:TP_Doctrine_MB Mohammed$ bin/type-list
 ------------------------
|  ID   |      Type      |
|-------|----------------|
    1        personnel
|-------|----------------|
    2        domicile
|-------|----------------|
    3        domicile
|-------|----------------|
    4        domicile
|-------|----------------|
    5        domicile
|-------|----------------|
    6        domicile
|-------|----------------|
    7        domicile
|-------|----------------|
    8        domicile
|-------|----------------|
    9        personel
|-------|----------------|
    10        personel
|-------|----------------|
```

## 2.8 Exercice 8

La commande bin/group-create permettant de créer un nouveau groupe :

```php
#!/usr/bin/php -Cq
<?php

require_once "bootstrap.php";

$groupe = new Groupe;
$groupe->setName($argv[1]);


$entityManager->persist($groupe);
$entityManager->flush();
echo "=> Onsuccess le groupe est crée \n";
echo "-ID : ".$groupe->getId()."\n";
echo "-Le nom du groupe : ".$groupe->getName()."\n";
```

```
[MacBook-Pro-de-Mohammed:TP_Doctrine_MB Mohammed$ bin/group-create Master1Icone
=> Onsuccess le groupe est crée
-ID : 6
-Le nom du groupe : Master1Icone
```

## 2.9 Exercice 9

La commande bin/group-remove permettant d'effacer un groupe :

```php
#!/usr/bin/php -Cq
<?php

require_once "bootstrap.php";

$groupID = $argv[1];

$groupe = $entityManager->find("Groupe", $groupID);

if ($groupe) {

    $entityManager->remove($groupe);
    $entityManager->flush();

    echo "Onsuccess :\n=> ".$groupe->getName()." est supprimé \n";
} else {
    echo "Ce ID n'existe pas \n";
}
```

```
[MacBook-Pro-de-Mohammed:TP_Doctrine_MB Mohammed$ bin/group-remove 4
Onsuccess :
=> mohammed est supprimé
```

## 2.10 Exercice 10

La commande bin/group-rename permettant de renommer un groupe :

```php
#!/usr/bin/php -Cq
<?php

require_once "bootstrap.php";

$groupID = $argv[1];

$groupe = $entityManager->find("Groupe", $groupID);

$groupe->setName($argv[2]);

$entityManager->persist($groupe);
$entityManager->flush();
echo "=> Onsuccess le nom du groupe est renommé \n";
echo "-ID du groupe : " . $groupe->getId() . "\n-Le nouveau nom : "
     . $groupe->getName() . "\n";
```

```
[MacBook-Pro-de-Mohammed:TP_Doctrine_MB Mohammed$ bin/group-rename 6 M1Icone2017/2018
=> Onsuccess le nom du groupe est renommé
-ID du groupe : 6
-Le nouveau nom : M1Icone2017/2018
```

## 2.11 Exercice 11

La commande bin/group-list permettant de lister les groupes :

```php
#!/usr/bin/php -Cq
<?php


require_once "bootstrap.php";


$groupeReq = $entityManager->getRepository("Groupe");
$groupe = $groupeReq->findAll();
    echo " ---------------------------\n";
    echo "|  ID   |       Type       |\n";
    echo "|-------|------------------|\n";
foreach ($groupe as $i) {
    echo "     ".$i->getId()."         ".$i->getName()."\n";
    echo "|-------|------------------|\n";
}
```

```
[MacBook-Pro-de-Mohammed:TP_Doctrine_MB Mohammed$ bin/group-list
 ----------------------------
|  ID   |         Type       |
|-------|--------------------|
    1       mohammed
|-------|--------------------|
    2       mohammed
|-------|--------------------|
    3       momo
|-------|--------------------|
    5       mohammed
|-------|--------------------|
    6       M1Icone2017/2018
|-------|--------------------|
```

## 2.12 Exercice 12

La commande bin/person-group-add permettant d'ajouter une personne à un groupe :

```php
#!/usr/bin/php -Cq
<?php

require_once "bootstrap.php";

$personne = $entityManager->find("Personne", $argv[1]);
$groupe = $entityManager->find("Groupe", $argv[2]);


if($personne) {

  if($groupe) {

    $groupe->addPersonne($personne);
    $personne->addGroupe($groupe);

    $entityManager->persist($personne);
    $entityManager->persist($groupe);
    $entityManager->flush();
    echo "Onsuccess \n";
    echo "=> ajouter ". $personne->getFirstName() . " " . $personne
    ->getLastName()."  au groupe " .$groupe->getName(). "\n";
  } else {
    echo "Le groupe n'existe pas"."\n";
  }
} else {
  echo "La personne n'existe pas"."\n";
}
```

```
[MacBook-Pro-de-Mohammed:TP_Doctrine_MB Mohammed$ bin/person-group-add 10 6
Onsuccess
=> ajouter Mohammed BENAOU  au groupe M1Icone2017/2018
```

## 2.13 Exercice 13

La commande bin/person-group-remove permettant d'effacer une personne d'un groupe :

```php
#!/usr/bin/php -Cq
<?php

require_once "bootstrap.php";

$personne = $entityManager->find("Personne", $argv[1]);
$groupe = $entityManager->find("Groupe", $argv[2]);

if($personne) {
  if($groupe) {

    $personne->removeGroupe($groupe);
    $groupe->removePersonne($personne);

    $entityManager->persist($groupe);
    $entityManager->persist($personne);

    $entityManager->flush();
    echo "Onsuccess \n";
    echo "=> ". $personne->getFirstName() . " " . $personne->
    getLastName()."  est supprimé du groupe " .$groupe->getName().
    "\n";
  } else {
    echo "Le groupe n'existe pas"."\n";
  }
} else {
  echo "La personne n'existe pas"."\n";
}
```

```
[MacBook-Pro-de-Mohammed:TP_Doctrine_MB Mohammed$ bin/person-group-remove 10 6
Onsuccess
=> Mohammed BENAOU  est supprimé du groupe M1Icone2017/2018
```

## 2.14 Exercice 14

La commande bin/person-group-list permettant de lister les groupes d'une personne :

```php
#!/usr/bin/php -Cq
<?php

require_once "bootstrap.php";


$personne = $entityManager->find("Personne", $argv[1]);

if($personne) {

  echo "La personne : ".$personne->getFirstName()." ". $personne->
    getLastName()."\n";
  foreach($personne->getGroupe() as $i) {
        echo "   => membre du groupe : ".$i->getName()."\n";
    }

} else {
  echo "La personne n'existe pas"."\n";
```

```
18 }
```

```
[MacBook-Pro-de-Mohammed:TP_Doctrine_MB Mohammed$ bin/person-group-list 7
La personne : Mohammed benaou
```

### 2.15 Exercice 15

La commande bin/group-person-list permettant de lister les personnes d'un groupe :

```
1  #!/usr/bin/php -Cq
2  <?php
3
4  require_once "bootstrap.php";
5
6  $groupe = $entityManager->find("Groupe", $argv[1]);
7
8  if($groupe) {
9    echo "Le Groupe : ".$groupe->getName()."\n";
10
11   foreach($groupe->getPersonnes() as $i) {
12     echo "    => Les membres sont : ".$i->getFirstName()."  ".$i->
       getLastName()."\n";
13
14     }
15
16 } else {
17   echo "Le groupe n'existe pas"."\n";
18 }
```

```
[MacBook-Pro-de-Mohammed:TP_Doctrine_MB Mohammed$ bin/group-person-list 6
Le Groupe : M1Icone2017/2018
    => Les membres sont : Mohammed  BENAOU
    => Les membres sont : momo  rasfa
```

## 3 Code Source

### 3.1 Personne.php

```
1  <?php
2
3  require_once "src/Telephone.php";
4
5  use Doctrine\Common\Collections\ArrayCollection;
6
7  /**
8   * @Entity @Table(name="personne")
9   */
10
11 class Personne {
```

```
12      /**
13       * @var int
14       * @Id @Column(type="integer") @GeneratedValue
15       */
16      protected $id;
17      /**
18       * @var string
19       * @Column(type="string")
20       */
21      protected $firstName;
22      /**
23       * @var string
24       * @Column(type="string")
25       */
26      protected $lastName;
27      /**
28       * @var Telephone[]
29       * @OneToMany(targetEntity="Telephone", mappedBy="personne",
        cascade={"persist", "remove"})
30       */
31      protected $telephones;
32      /**
33     * @ManyToMany(targetEntity="Groupe", inversedBy="personnes")
34     */
35      protected $groupes;
36
37      public function getId() {
38          return $this->id;
39      }
40
41      public function __construct () {
42          $this->telephones = new ArrayCollection();
43          $this->groupes = new ArrayCollection();
44      }
45
46      public function addPhone (Telephone $telephone) {
47          $this->telephones[] = $telephone;
48          $telephone->setPersonne($this);
49      }
50      public function getPhones () {
51      return $this->telephones;
52    }
53      public function getLastName() {
54          return $this->lastName;
55      }
56      public function setLastName($lastName) {
57          return $this->lastName = $lastName;
58      }
59      public function getFirstName() {
60          return $this->firstName;
61      }
62      public function setFirstName($firstname) {
63          return $this->firstName = $firstname;
64      }
65      public function getGroupe() {
66          return $this->groupes;
67      }
68      public function addGroupe(Groupe $groupe) {
69          $this->groupes[] = $groupe;
70      }
71      public function removeGroupe(Groupe $groupe) {
72      $this->groupes->removeElement($groupe);
```

```
73      }
74  }
```

## 3.2 Telephone.php

```
1   <?php
2
3   require_once "src/Personne.php";
4
5   require_once "src/Type.php";
6
7   // src/Telephone.php
8
9   /**
10   * @Entity @Table(name="telephone")
11   */
12
13  class Telephone {
14    /**
15       * @var int
16       * @Id @Column(type="integer") @GeneratedValue
17       */
18    protected $id;
19    /**
20       * @var string
21       * @Column(type="string")
22       */
23    protected $numero;
24    /**
25     * @ManyToOne(targetEntity="Type", inversedBy="telephones")
26     */
27    protected $type;
28
29    /**
30     * @ManyToOne(targetEntity="Personne", inversedBy="telephones")
31     */
32    protected $personne;
33
34    public function getId() {
35          return $this->id;
36      }
37
38    public function getNumero() {
39          return $this->numero;
40      }
41      public function setNumero($numero) {
42          return $this->numero = $numero;
43      }
44    public function getType() {
45      return $this->type;
46    }
47    public function setType($type) {
48      return $this->type = $type;
49    }
50    public function getPersonne() {
51      return $this->personne;
52    }
53    public function setPersonne($personne) {
```

```
54      return $this->personne = $personne;
55    }
56  }
```

### 3.3 Groupe.php

```
1   <?php
2
3   use Doctrine\Common\Collections\ArrayCollection;
4
5   /**
6    * @Entity @Table(name="groupe")
7    */
8
9   class Groupe {
10    /**
11       * @var int
12       * @Id @Column(type="integer") @GeneratedValue
13       */
14    protected $id;
15    /**
16       * @var string
17       * @Column(type="string")
18       */
19    protected $name;
20    /**
21       * @var Personne[]
22       * @ManyToMany(targetEntity="Personne", mappedBy="groupes")
23       */
24    protected $personnes;
25
26    public function getId() {
27          return $this->id;
28      }
29
30    public function __construct () {
31          $this->personnes = new ArrayCollection();
32      }
33
34    public function getName() {
35          return $this->name;
36      }
37    public function setName($name) {
38          return $this->name = $name;
39      }
40    public function getPersonnes() {
41      return $this->personnes;
42    }
43    public function addPersonne(Personne $personne) {
44      $this->personnes[] = $personne;
45    }
46    public function removePersonne(Personne $personne) {
47      $this->personnes->removeElement($personne);
48    }
49  }
```

### 3.4 Type.php

```php
<?php
require_once "src/Telephone.php";

use Doctrine\Common\Collections\ArrayCollection;

// src/Type.php

/**
 * @Entity @Table(name="type")
 */

class Type {
  /**
     * @var int
     * @Id @Column(type="integer") @GeneratedValue
     */
  protected $id;
  /**
     * @var string
     * @Column(type="string")
     */
  protected $type;
  /**
     * @var Telephone[]
     * @OneToMany(targetEntity="Telephone", mappedBy="type")
     */
  protected $telephones;

  public function getId() {
        return $this->id;
    }

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

  public function getType() {
        return $this->type;
    }
  public function setType($type) {
        return $this->type = $type;
    }
  public function addPhone (Telephone $telephone) {
        $this->telephones[] = $telephone;
        $telephone->setType($this);
    }
  public function getPhones () {
    return $this->telephones;
  }
}
```