

## Introduction et Configuration

GitHub du projet : [https://github.com/lclerc561/DWM\\_ParadigmeBDD.git](https://github.com/lclerc561/DWM_ParadigmeBDD.git)

L'objectif de ce TD est de maîtriser Doctrine ORM pour interagir avec une base de données sans écrire de SQL brut. Nous avons configuré l'EntityManager dans le fichier bootstrap.php et défini le mappage des entités via des fichiers XML.

Mapping : Les fichiers comme pratilib.domain.entities.Praticien.dcm.xml définissent la correspondance entre les classes PHP et les tables de la base de données.

```
<?xml version="1.0" encoding="UTF-8"?>
<doctrine-mapping xmlns="http://doctrine-project.org/schemas/orm/doctrine-mapping"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://doctrine-project.org/schemas/orm/doctrine-mapping
    https://doctrine-project.org/schemas/orm/doctrine-mapping.xsd">

  <entity name="pratilib\domain\entities\Praticien" table="praticien">

    <id name="id" column="id" type="uuid">
      <generator strategy="IDENTITY"/>
    </id>

    <field name="nom" column="nom" type="string" />
    <field name="rpps_id" column="rpps_id" type="string" />
    <field name="prenom" column="prenom" type="string" />
    <field name="titre" column="titre" type="string" />
    <field name="ville" column="ville" type="string" />
    <field name="email" column="email" type="string" />
    <field name="telephone" column="telephone" type="string" />
    <field name="organisation" column="organisation" type="boolean" />
    <field name="accepte_nouveau_patient" column="accepte_nouveau_patient" type="boolean" />
    <field name="specialite_id" column="specialite_id" type="integer" />
    <field name="structure_id" column="structure_id" type="integer" />

  </entity>
</doctrine-mapping>
```

## Exercice 1 : Utilisation élémentaire

Dans cet exercice, nous manipulons les entités via l'EntityManager pour effectuer des opérations CRUD (Create, Read, Update, Delete) et naviguer dans les relations.

**1. Afficher la spécialité d'identifiant 1 :** Nous utilisons la méthode find() sur le repository de l'entité Specialite.

```
$specialiteID1 = $specialiteRepository->find(id: 1);

echo "Exercice 1 <br>";
echo "1) <br>";
if ($specialiteID1) {
    echo "ID: " . $specialiteID1->getId() . "<br>";
    echo "Libellé: " . $specialiteID1->getLibelle() . "<br>";
    echo "Description: " . $specialiteID1->getDescription() . "<br>";
} else {
    echo "Spécialité avec l'ID 1 non trouvée.";
}
```

```
Exercice 1
1)
ID: 1
Libellé: médecine générale
Description: Médecine Générale
```

**2. Afficher le praticien (ID spécifique) :** Nous utilisons find() avec l'UUID fourni.

```
$praticienID1 = $praticienRepository->find(id: "8ae1400f-d46d-3b50-b356-269f776be532");
echo "2) <br>";
if ($praticienID1) {
    echo "ID: " . $praticienID1->getId() . "<br>";
    echo "Nom: " . $praticienID1->getNom() . "<br>";
    echo "Prénom: " . $praticienID1->getPrenom() . "<br>";
    echo "Ville: " . $praticienID1->getVille() . "<br>";
    echo "Email: " . $praticienID1->getEmail() . "<br>";
    echo "Téléphone: " . $praticienID1->getTelephone() . "<br>";
} else {
    echo "Praticien avec l'ID 8ae1400f-d46d-3b50-b356-269f776be532 non trouvé.";
}
```

```
2)
ID: 8ae1400f-d46d-3b50-b356-269f776be532
Nom: Klein
Prénom: Gabrielle
Ville: Paris
Email: Gabrielle.Klein@live.com
Téléphone: +33 (0)3 90 27 98 80
```

### 3. Compléter avec spécialité et structure :

```
echo "3) <br>";
if ($praticienID1) {
    $specialite = $praticienID1->getSpecialite();
    if ($specialite) {
        echo "Spécialité du praticien:<br>";
        echo "ID: " . $specialite->getId() . "<br>";
        echo "Libellé: " . $specialite->getLibelle() . "<br>";
        echo "Description: " . $specialite->getDescription() . "<br>";
    } else {
        echo "Le praticien n'a pas de spécialité associée.<br>";
    }
    echo "<br>";
    $structure = $praticienID1->getStructure();
    if ($structure) {
        echo "Structure d'un praticien:<br>";
        echo "ID: " . $structure->getId() . "<br>";
        echo "Nom: " . $structure->getNom() . "<br>";
        echo "Adresse: " . $structure->getAdresse() . "<br>";
        echo "Ville: " . $structure->getVille() . "<br>";
        echo "Code Postal: " . $structure->getCodePostal() . "<br>";
        echo "Téléphone: " . $structure->getTelephone() . "<br>";
    }
} else {
    echo "Praticien avec l'ID 8ae1400f-d46d-3b50-b356-269f776be532 non trouvé.";
}
```

4. Afficher la structure (ID spécifique) et ses praticiens : Après avoir récupéré la structure, nous utilisons la méthode getPraticiens().

```
$structure = $structureRepository->find(id: '3444bdd2-8783-3aed-9a5e-4d298d2a2d7c');
echo "<br>-----<br>";
echo "4) <br>";
if ($structure) {
    echo "Structure et praticiens: <br>";
    echo "Structure ID: " . $structure->getId() . "<br>";
    echo "Nom: " . $structure->getNom() . "<br>";
    echo "Adresse: " . $structure->getAdresse() . "<br>";
    echo "Ville: " . $structure->getVille() . "<br>";
    echo "Téléphone: " . $structure->getTelephone() . "<br>";

    echo "<br>Praticiens rattachés :<br>";
    $praticiens = $structure->getPraticiens();

    if (count(value: $praticiens) > 0) {
        foreach ($praticiens as $praticien) {
            echo "<li>" . $praticien->getNom() . " " . $praticien->getPrenom() . " (" . $praticien->getSpecialite()->getLibelle() . ")</li>";
        }
        echo "</ul>";
    } else {
        echo "Aucun praticien rattaché.<br>";
    }
} else {
    echo "Structure avec l'ID 3444bdd2-8783-3aed-9a5e-4d298d2a2d7c non trouvée.<br>";
}
```

```
4)
Structure et praticiens:
Structure ID: 3444bdd2-8783-3aed-9a5e-4d298d2a2d7c
Nom: Cabinet Bigot
Adresse: 63, rue de Mercier
Ville: Paris
Téléphone: 01 02 03 04 05
```

Praticiens rattachés :

- Paul Marine (médecine générale)
- Guichard Laurence (médecine générale)
- Pichon Arnaude (ophtalmologie)
- Klein Gabrielle (médecine générale)
- Gallet Valérie (Imagerie médicale)
- Goncalves Noël (Imagerie médicale)

**5. Afficher la spécialité 1 et les motifs de visite :** De la même manière, nous naviguons de la spécialité vers ses motifs.

```
echo "<br>-----<br>";
echo "5) <br>";
if ($specialiteID1) {
    echo "Motifs de visite pour la spécialité ID 1:<br>";
    $motifs = $specialiteID1->getMotifVisite();

    if (count(value: $motifs) > 0) {
        foreach ($motifs as $motif) {
            echo "<li>" . $motif->getLibelle() . "</li>";
        }
        echo "</ul>";
    } else {
        echo "Aucun motif de visite pour cette spécialité.<br>";
    }
} else {
    echo "Spécialité avec l'ID 1 non trouvée.";
}
```

```
5)
Motifs de visite pour la spécialité ID 1:
• Consultation
• Renouvellement de traitement
• Médecine du sport
• Enfants
```

**6. Praticien (ID spécifique) :** afficher ses motifs de visite Nous accédons à la collection des motifs associés au praticien récupéré précédemment.

```
echo "<br>-----<br>";
echo "6) <br>";
if ($praticienID1) {
    echo "Liste des motifs pour le praticien " . $praticienID1->getNom() . " :<br>";

    $lesMotifs = $praticienID1->getMotifs();

    if (count(value: $lesMotifs) > 0) {
        echo "<ul>";
        foreach ($lesMotifs as $motif) {
            echo "<li>" . $motif->getLibelle() . "</li>";
        }
        echo "</ul>";
    } else {
        echo "Aucun motif associé à ce praticien.<br>";
    }
} else {
    echo "Praticien introuvable.";
}
```

6)  
Liste des motifs pour le praticien Klein :

- Consultation
- Renouvellement de traitement
- Médecine du sport
- Enfants

**7. Créer un praticien et le sauvegarder :** Nousinstancions un nouvel objet Praticien, nous paramétrons ses propriétés, nous récupérons la spécialité "Pédiatrie" (ID 5) pour lier l'objet, puis nous demandons la persistance.

```
echo "7) <br>";
echo "<h3> Création d'un nouveau praticien </h3>";

$pediatrie = $specialiteRepository->findOneBy(criteria: ['libelle' => 'pédiatrie']);

$nouveauId = Uuid::uuid4()->toString();
echo "Nouveau ID généré pour le praticien : " . $nouveauId . "<br>";
$nouveauPraticien = new Praticien(
    id: $nouveauId,
    nom: "Martin",
    rpps_id: "1010101010",
    prenom: "Alice",
    titre: "Dr.",
    ville: "Nancy",
    email: "alice.martin@test.fr",
    telephone: "0606060606",
    organisation: "0",
    accepte_nouveau_patient: "1",
    specialite: $pediatrie,
    structure: null
);

$entityManager->persist(object: $nouveauPraticien);

try {
    $entityManager->flush();
    echo "Succès ! Le praticien <b>" . $nouveauPraticien->getNom() . "</b> (ID: $nouveauId) a été enregistré en base.";
} catch (Exception $e) {
    echo "Erreur lors de l'enregistrement : " . $e->getMessage();
}
```

7)

### Création d'un nouveau praticien

Nouveau ID généré pour le praticien : 8bbef670-17d6-4725-8258-f140f70edcf3  
Succès ! Le praticien **Martin** (ID: 8bbef670-17d6-4725-8258-f140f70edcf3) a été enregistré en base.

**8. Modifier ce praticien :** Nous modifions l'objet PHP \$newPraticien (ville, structure, ajout de motif). Doctrine détecte les changements. Le flush() génère l'ordre SQL UPDATE.

```
echo "8) <br>";  
$nouveauPraticien->setVille(ville: "Paris");  
$structure = $structureRepository->findOneBy(criteria: ['nom' => 'Cabinet Bigot']);  
if ($structure) {  
    $nouveauPraticien->AddToStructure(structure: $structure);  
    $entityManager->flush();  
    echo "Modification réussie";  
} else {  
    echo "Structure introuvable";  
}
```

8)

Modification réussie

**9. Supprimer ce praticien :** Nous utilisons la méthode remove() sur l'entité suivie.

```
try {  
    $entityManager->remove(object: $nouveauPraticien);  
    $entityManager->flush();  
    echo "Supression de l'utilisateur réussie";  
} catch (Exception $e) {  
    echo "Erreur lors de la suppression : " . $e->getMessage();  
}
```

9)

Supression de l'utilisateur réussie

## Exercice 2 : Requêtes avec conditions de sélection

Ici, nous utilisons les méthodes de recherche par critères (findOneBy, findBy) et l'API Criteria.

**1. Praticien par mail:** Utilisation de findOneBy car le mail est unique.

```
echo "1) <br>";  
$praticien = $praticienRepository->findOneBy(criteria: ["email" => "Gabrielle.Klein@live.com"]);  
echo "Praticien: " . $praticien->getNom() . " " . $praticien->getPrenom();
```

```
1)
Praticien: Klein Gabrielle
```

**2. Praticien nom Goncalves à Paris :** Utilisation de findBy avec un tableau associatif de critères.

```
echo "2) <br>";
$praticien = $praticienRepository->findOneBy(criteria: ["nom" => "Goncalves" , "ville" => "Paris"]);
echo "Praticien: " . $praticien->getNom() . " " . $praticien->getPrenom();
```

```
2)
Praticien: Goncalves Noël
```

**3. Spécialité 'pédiatrie' et praticiens associés :** Nous récupérons d'abord la spécialité par son libellé, puis nous affichons sa collection de praticiens.

```
echo "3) <br>";
$specialite = $specialiteRepository->findOneBy(criteria: ["libelle" => "pédiatrie"]);
echo "spécialité:" . $specialite->getLibelle();
```

```
3)
spécialité:pédiatrie
```

**4. Types de groupements contenant 'santé' :** Pour filtrer sur une chaîne partielle sans DQL, nous utilisons l'objet Criteria de Doctrine Collections.

```
echo "4) <br>";
echo "Structures contenant 'santé' (via critère) :<br>";

$structure = $structureRepository->matching(criteria: Criteria::create()
->where(expression: Criteria::expr()->contains(field: 'nom', value: 'Santé')));

if (count(value: $structure) > 0) {
    echo "<ul>";
    foreach ($structure as $struct) {
        echo "<li>" . $struct->getNom() . " (" . $struct->getVille() . ")</li>";
    }
    echo "</ul>";
} else {
    echo "Aucune structure ne contient le mot 'Santé' dans son nom.<br>";
}
```

```
4)
Structures contenant 'santé' (via critère) :

• Pichon Santé (Diaz-sur-Boulanger)
• Maison de Santé de Care-Sur_Turpin (Carre-sur-Turpin)
• Maison Santé Lombard (Paris-la-Forêt)
```

**5. Praticiens 'ophtalmologie' à Paris :** Nous récupérons la spécialité, puis nous itérons sur les praticiens en filtrant ceux dont la ville est "Paris".

```

echo "5) <br>";
echo "Praticiens 'ophtalmologie' à Paris :<br>";

$specialite = $specialiteRepository->findOneBy(criteria: ['libelle' => 'ophtalmologie']);

if ($specialite) {
    $praticien = $praticienRepository->matching(criteria: Criteria::create()
        ->Where(expression: Criteria::expr()->eq(field: 'specialite', value: $specialite))
        ->andWhere(expression: Criteria::expr()->eq(field: 'ville', value: 'Paris')));

    if (count(value: $praticien) > 0) {
        echo "<ul>";
        foreach ($praticien as $praticien) {
            echo "<li>Dr. " . $praticien->getNom() . " " . $praticien->getPrenom() . "</li>";
        }
        echo "</ul>";
    } else {
        echo "Aucun ophtalmologue trouvé à Paris.<br>";
    }
} else {
    echo "La spécialité 'ophtalmologie' n'existe pas dans la base.<br>";
}

```

```

5)
Praticiens 'ophtalmologie' à Paris :

• Dr. Pichon Arnaude

```

## Exercice 3 : Repository et DQL

Pour les requêtes complexes, nous avons défini des méthodes personnalisées dans les fichiers Repository en utilisant le DQL (Doctrine Query Language).

### 1. Spécialités contenant un mot clé

Méthode : SpecialiteRepository->findByMotCle

```

echo "<b>1) Spécialités contenant 'médecine' ou 'chirurgie' :</b><br>";
$resultat1 = $specialiteRepository->findByMotCle('médecine');
$resultat2 = $specialiteRepository->findByMotCle('chirurgie');
$resultats = array_merge(arrays: $resultat1, $resultat2);
$resultats = array_unique(array: $resultats, flags: SORT_REGULAR);

if (count(value: $resultats) > 0) {
    foreach ($resultats as $spe) {
        echo "- " . $spe->getLibelle() . "<br>";
    }
} else {
    echo "Aucune spécialité trouvée.<br>";
}

```



**1) Spécialités contenant 'médecine' ou 'chirurgie' :**  
- médecine générale

**2. Praticiens dont la spécialité contient un mot-clé**

Méthode : PraticienRepository->findBySpecialiteMotCle

```
echo "<b>2) Praticiens dont la spécialité contient 'dent' :</b><br>";
$praticiens = $praticienRepository->findBySpecialiteMotCle('dent');

if (count(value: $praticiens) > 0) {
    foreach ($praticiens as $p) {
        $libelleSpe = $p->getSpecialite() ? $p->getSpecialite()->getLibelle() : 'Aucune';
        echo "- Dr. " . $p->getNom() . " (" . $libelleSpe . ")<br>";
    }
} else {
    echo "Aucun praticien trouvé.<br>";
}
```

**2) Praticiens dont la spécialité contient 'dent' :**

- Dr. Roger (Dentiste)
- Dr. Parent (Dentiste)
- Dr. Joly (Dentiste)
- Dr. Legros (Dentiste)
- Dr. Blanc (Dentiste)
- Dr. Pascal (Dentiste)
- Dr. Fournier (Dentiste)
- Dr. Gautier (Dentiste)
- Dr. Perrier (Dentiste)
- Dr. Leclerc (Dentiste)
- Dr. Renard (Dentiste)
- Dr. Fabre (Dentiste)
- Dr. Dias (Dentiste)

**3. Praticiens d'une spécialité acceptant un paiement donné**

Méthode : PraticienRepository->findBySpecialiteAndPaiement

```
echo "<b>3) Praticiens 'médecine générale' acceptant 'chèque' :</b><br>";

$praticiensPaiement = $praticienRepository->findBySpecialiteAndPaiement('médecine générale', 'chèque');

if (count(value: $praticiensPaiement) > 0) {
    foreach ($praticiensPaiement as $p) {
        echo "- Dr. " . $p->getNom() . " " . $p->getPrenom() . "<br>";
    }
} else {
    echo "Aucun praticien trouvé pour ces critères.<br>";
}
```

**3) Praticiens 'médecine générale' acceptant 'chèque' :**

- Dr. Paul Marine
- Dr. Guichard Laurence
- Dr. Klein Gabrielle
- Dr. Meunier Michelle
- Dr. Lucas Gilbert
- Dr. Perrin Gérard