

# Compte rendu

## Accès à une base de données

### Table des matières

---

<b>Introduction</b>	Page 1
<b>Exercice 1</b>	Page 2
<b>Exercice 2</b>	Page 3
<b>Exercice 3</b>	Page 4
<b>Exercice 4</b>	Page 4 - 5
<b>Exercice 5</b>	Page 6
<b>Exercice 6</b>	Page 7

---

### Introduction

L'objectif de ce TP est d'exploiter le contenu d'une base de données grâce à des pages en langage PHP et de transférer des données d'une page à l'autre.  
On commence alors le TP en créant la base de données grâce au script fourni au début du TP.

## Exercice 1 :

Ce programme a pour objectif d'afficher la liste de tous les employés.

```
<!DOCTYPE html>
<html>
  <head><title>Liste des employés</title></head>
  <body>
    <h1>Liste des employés</h1>
    <?php
      try {
        $bdd = new PDO("mysql:host=172.28.100.3;dbname=personnel_hmarceau;charset=utf8", "hmarceau",
          "elini01", array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));
        $requete="SELECT * FROM employe";
        $resultat = $bdd->query($requete);
        $donnees=$resultat->fetchAll();
        $nombreDonnees=0;
        echo "<ul>";
        foreach($donnees as $ligne)
        {
          echo "<li>";
          echo $ligne['NomEmpl'];
          echo " ";
          echo $ligne['PrenomEmpl'];
          echo "</li>";
          $nombreDonnees++;
        }
        echo "</ul>";
        echo "Total : ",$nombreDonnees," employé(s)";
      }
      catch (Exception $e) {
        die('Erreur fatale : ' . $e->getMessage());
      }
    ?>
  </body>
</html>
```

*listeEmployes.php*

- ☐ On commence par se connecter à la base de données.
- ☐ Ensuite, On récupère la liste des employés.  
On utilise la méthode **query** afin d'exécuter une requête SQL
- ☐ Cette variable va nous être utile afin de compter le nombre d'employé de la liste (pour l'instant, il y a 12 employés, mais par la suite de ce TP on va créer une page PHP qui nous permettra d'ajouter des employés à la liste).
- ☐ Maintenant, on procède à l'affichage de la liste, grâce à des balises HTML **<ul>** et **<li>**.  
On incrémente la variable **nombreDonnees** de 1 pour chaque employé de la liste.  
La boucle **foreach** permet de parcourir le tableau contenant les informations sur les employés
- ☐ On finit par afficher le nombre total d'employé de la liste
- ☐ Cette instruction sert dans le cas où la connexion à la base de données a échoué, on affiche le détail du message à l'utilisateur.

## Exercice 2

Cet exercice est similaire au précédent. La seule différence est que l'affichage des employés se fera dans un tableau. On affichera donc, en plus de leur nom et de leur prénom, leur matricule. On en a également profité pour centraliser la connexion à la base de données et protéger la page des injection JavaScript grâce à une page PHP externe (**fonction.php**, se référer à l'exercice 3).

```
<!DOCTYPE html>
<html>
  <head><title>Liste des employés</title></head>
  <body>
    <h1>Liste des employés</h1>
    <?php
    include 'fonction.php';
    try {
      $bdd=getBdd();
      $requete="SELECT * FROM employe";
      $resultat =$bdd->query($requete);
      $donnees=$resultat->fetchAll();
      $nombreDonnees=0;
      echo "<table>";
      echo "<thead>";
      echo "<tr>";
      echo "<th>Matricule</th>";
      echo "<th>Nom</th>";
      echo "<th>Prénom</th>";
      echo "</tr>";
      echo "</thead>";
      foreach($donnees as $ligne)
      {
        echo "<tr>";
        echo "<td>",escape($ligne['Matricule']),"</td>";
        echo "<td>",escape($ligne['NomEmpl']),"</td>";
        echo "<td>",escape($ligne['PrenomEmpl']),"</td>";
        echo "</tr>";
        $nombreDonnees++;
      }
      echo "</table>";
      echo "<br>";
      echo "Total : ",$nombreDonnees," employé(s)";
    }
    catch (Exception $e) {
      die('Erreur fatale : ' . $e->getMessage());
    }
  ?>
</body>
</html>
```

On inclut la page **fonction.php**

Récupération de la liste des employés

Initialisation de la variable de Comptage du nombre d'employé

Procédure d'affichage du tableau

Incrémentement de 1 à chaque employé supplémentaire

En cas d'échec de la connexion à la base de données, on affiche le détail de l'erreur à l'utilisateur

*tableauEmployes.php*

### Exercice 3

Cette page contient uniquement deux fonctions que nous allons réutiliser tout au long de ce TP afin de simplifier et centraliser la connexion à la base de données et de protéger nos pages PHP :

- La fonction `escape` permet de protéger la page contre d'éventuels injection JavaScript.
- La fonction `getBdd` permet de se connecter à la base de données.

```
<?php
function escape($valeur) {
    return htmlspecialchars($valeur, ENT_QUOTES, 'UTF-8', false);
}
function getBdd() {
    return new PDO("mysql:host=172.28.100.3;dbname=personnel_hmarceau;charset=utf8", "hmarceau",
        "elini01", array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION));
}
?>
```

*fonction.php*

Il suffit alors d'inclure cette page au début des page PHP afin de pouvoir utiliser les fonctions qu'elle contient.

### Exercice 4

Cet exercice contient deux pages PHP. La première page permet à l'utilisateur de sélectionner un service et la deuxième page permet d'afficher la liste des employés affecté a ce même service.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Choix d'un service</title>
    </head>
    <body>
        <h1>Choix d'un service</h1>
        <?php
            include 'fonction.php';
            $bdd=getBdd();
        ?>
        <form method="POST" action="employeesService.php">
            <?php
                $resultat=$bdd->query('SELECT DesiServ FROM service');
                echo '<div><label for="service">Service : </label><select name="service">';
                while($donnees = $resultat->fetch()){
                    echo '<option value="'. $donnees['DesiServ']. '>'. $donnees['DesiServ']. '</option>';
                }
                echo '</select></div>';
            ?>
            <div>
                <input type="submit" value="Afficher les employés">
            </div>
        </form>
    </body>
</html>
```

*service.php*

- ☐ On inclut la page **fonction.php** afin de se connecter à la base de données.
- ☐ La méthode POST va nous permettre de transmettre le service sélectionné sur la page **employeService.php**.  
On crée donc un menu déroulant avec les balises HTML **<label>** et **<select>**.  
On utilise la méthode **query** afin d'exécuter une requête SQL (récupérer la liste des services).  
On utilise la méthode **fetch()** pour parcourir le résultat de la requête ligne par ligne.
- ☐ Insertion du bouton envoyer.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>liste employés</title>
  </head>
  <body>
    <?php
      include 'fonction.php';
      $bdd = getBdd();
      if(isset($_POST['service'])){
        $service=$_POST['service'];
        echo "<h1>Liste des employés du service $service</h1>";
        $resultat=$bdd->query("SELECT Matricule,NomEmpl,PrenomEmpl FROM employe,service WHERE ServEmpl=CodeServ AND
        DesiServ='".$service."'");
        echo '<table><tr><th>Matricule</th><th>Nom</th><th>Prénom</th></tr>';
        while($donnees = $resultat->fetch()){
          echo '<tr><td>'.$donnees['Matricule'].'</td><td>'.$donnees['NomEmpl'].'</td><td>'.$donnees['PrenomEmpl'].'</td></tr>';
        }
        echo '</table>';
      }
      else{
        echo 'Erreur, veuillez saisir une information.';
      }
    ?>
  </body>
</html>
```

*employesService.php*

- ☐ La méthode **POST** nous permet maintenant de récupérer le service sélectionné sur la page précédente.  
La fonction **isset** de vérifier si un service a bien été sélectionné. Dans le cas où aucun service n'a été choisi, un message d'erreur est renvoyé à l'utilisateur.

L'affichage de la liste des employés se fera sous forme d'un tableau, avec les balises HTML **<table>**, **<tr>** et **<td>**.

## Exercice 5

Cette page va permettre d'afficher les services existants et d'en ajouter un nouveau (avec son code et sa désignation).

La page insère le nouveau service dans la base lorsque l'utilisateur valide sa saisie.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ajout d'un service</title>
  </head>
  <body>
    <h1>liste des services</h1>
    <?php
      include 'fonction.php';
      $bdd=getBdd();
      $resultat=$bdd->query("SELECT * FROM service");
      echo '<table><tr><th>Code</th><th>Désignation</th></tr>';
      while($donnees = $resultat->fetch()){
        echo '<tr><td>'.$donnees['CodeServ'].'</td><td>'.$donnees['DesiServ'].'</td></tr>';
      }
      echo '</table>';
    ?>
    <h1>Ajout d'un service</h1>
    <form method="POST" action="./ajoutService.php">
      <div>
        <label for="CodeServ">Code : </label>
        <input id="CodeServ" name="CodeServ" type="text" required >
      </div>
      <div>
        <label for="desi">Désignation : </label>
        <input id="desi" name="desi" type="text" required >
      </div>
      <div>
        <input type="submit" value="Ajouter le service">
      </div>
    </form>
    <?php
      if(isset($_POST['CodeServ'])){
        $test=(control_doublon('CodeServ','service',$_POST['CodeServ'],$bdd));
        if(!$test){
          if(!($bdd->query("INSERT INTO service VALUES ('".$_POST['CodeServ']."'','".$_POST['desi']."'"))){
            echo "Erreur, ".mysqli_error($bdd);
          }
          else {
            echo "L'ajout a réussi !";
          }
        }
        else{
          echo "Erreur, code déjà utilisé.";
        }
      }
    ?>
  </body>
</html>
```

*ajoutService.php*

- ☐ Connexion à la base de données et récupération des services.
- ☐ Affichage des services déjà existant sous la forme d'un tableau
- ☐ Construction du formulaire d'ajout de service.
- ☐ On commence par vérifier si le code service a bien été renseigné (grâce à la fonction **isset**) et si le service n'existe pas déjà (grâce à la fonction **control\_doublon**). S'il est déjà utilisé, on avertit l'utilisateur. Pour toutes les autres erreurs, on affiche à l'utilisateur le détail de l'erreur SQL. Si tout s'est bien passé, on affiche (sur la même page cette fois ci) « L'ajout a réussi ! ».

## Exercice 6

Cette page va permettre d'ajouter un nouvel employé. Il est important de prendre en compte que toutes les informations sont obligatoires (matricule, nom, prénom, service et éventuellement cadre).

La page insère le nouvel employé dans la base lorsque l'utilisateur valide sa saisie.

```
<?DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>liste des services</title>
</head>
<body>
<h1>liste des services</h1>
<?php
include 'fonction.php';
$dbdd=getDbdd();
?>
<h1>Ajout d'un Employé</h1>
<form method="POST" action="/ajoutEmploye.php">
<div>
<label for="Matricule">Matricule : </label>
<input id="Matricule" name="Matricule" type="text" required >
</div>
<div>
<label for="NomEmpl">Nom : </label>
<input id="NomEmpl" name="NomEmpl" type="text" required >
</div>
<div>
<label for="PrenomEmpl">Prénom : </label>
<input id="PrenomEmpl" name="PrenomEmpl" type="text" required >
</div>
<div>
<label for="CodeCadre">Cadre ? </label>
<input id="CodeCadre" name="CodeCadre" type="checkbox" value="checked">
</div>
<div>
<label for="service">Service : </label>
<select name="service">
<?php
$resultat=$dbdd->query('SELECT * FROM service');
while($donnees = $resultat->fetch()){
echo '<option values="' . $donnees['CodeServ'] . '">' . $donnees['DesiServ'] . '</option>';
}
?>
</select>
</div>
<div>
<input type="submit" value="Ajouter l'employé">
</div>
</form>
<?php
if(isset($_POST['Matricule'])){
$test=control_doublon('Matricule',$_POST['Matricule'],$dbdd);
if(!$test){
if(!$dbdd->query('INSERT INTO employe VALUES (' . $_POST['Matricule'] . ',' . $_POST['NomEmpl'] . ',' . $_POST['PrenomEmpl'] . ',' . $_POST['CodeCadre'] . ',' . $_POST['service'] . ')')){
echo "Erreur, " . mysql_error($dbdd);
}
else {
echo "L'ajout a réussi !";
}
}
else{
echo "Erreur, matricule déjà utilisé.";
}
}
?>
</body>
</html>
```

ajoutEmploye.php

- ☐ Connexion à la base de données
- ☐ Construction du formulaire d'ajout d'un employé (de « Matricule : » jusqu'à « Cadre ? »)
- ☐ Construction du menu déroulant des services existant. On récupère la liste des services existant sur la base de données.
- ☐ Ajout du bouton Envoyer
- ☐ On commence par vérifier si le matricule a bien été renseigné (grâce à la fonction **isset**) et si l'employé n'existe pas déjà (grâce à la fonction **control\_doublon**). S'il est déjà utilisé, on avertit l'utilisateur. Pour toutes les autres erreurs, on affiche à l'utilisateur le détail de l'erreur SQL. Si tout s'est bien passé, on affiche sur la même page « L'ajout a réussi ! ».