

Morgado-Samagaio Jonathan

2A

**PWS / TP5 PHP**

## Table des matières

1.	Introduction .....	3
2.	Création de la base de données.....	3
3.	Modifier les pages existantes.....	3
4.	Conclusion.....	5

## 1. Introduction

Dans ce TP, nous allons voir comment relier une base de données MySQL à notre site php.

## 2. Création de la base de données

Nous allons donc commencer par compléter le fichier SQL fournis :

```
You, 5 days ago | 1 author (You)
1 DROP TABLE IF EXISTS Produits;          You, 5 days ago • Ajout 19/11
2 DROP TABLE IF EXISTS Categories;
3
4 CREATE TABLE Categories (
5     idCategorie VARCHAR(3),
6     nomCategorie VARCHAR(25),
7     PRIMARY KEY (idCategorie)
8 );
9
10 CREATE TABLE Produits (
11     idProduit VARCHAR(3),
12     idCategorie VARCHAR(3) NOT NULL,
13     nomProduit VARCHAR(30),
14     prixProduit mediumint(5),
15     PRIMARY KEY (idProduit),
16     FOREIGN KEY (idCategorie) references Categories(idCategorie)
17 );
18
19
20 INSERT INTO Categories (idCategorie, nomCategorie) VALUES ('100', 'Téléviseur');
21 INSERT INTO Categories VALUES ('200', 'Camescope');
22 INSERT INTO Categories(idCategorie, nomCategorie) VALUES ('300', 'Ordinateur');
23
24 INSERT INTO Produits (idProduit, idCategorie, nomProduit, prixProduit) VALUES ('115', '100', 'TV Thomson 214i', 1020);
25 INSERT INTO Produits (idProduit, idCategorie, nomProduit, prixProduit) VALUES ('198', '100', 'TV Samsung Highcolor', 980);
26 INSERT INTO Produits (idProduit, idCategorie, nomProduit, prixProduit) VALUES ('231', '200', 'Camescope Hightrack', 754);
27 INSERT INTO Produits (idProduit, idCategorie, nomProduit, prixProduit) VALUES ('302', '300', 'PC HP 5980', 299);
28 INSERT INTO Produits (idProduit, idCategorie, nomProduit, prixProduit) VALUES ('357', '300', 'PC Server Pro', 799);
```

Lorsque nous exécutons ce code sous MySQL, nous obtenons :

```
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.1472 seconds.)
DROP TABLE IF EXISTS Produits;
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.1051 seconds.)
DROP TABLE IF EXISTS Categories;
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.1591 seconds.)
CREATE TABLE Categories ( idCategorie VARCHAR(3), nomCategorie VARCHAR(25), PRIMARY KEY (idCategorie) );
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.1713 seconds.)
CREATE TABLE Produits ( idProduit VARCHAR(3), idCategorie VARCHAR(3) NOT NULL, nomProduit VARCHAR(30), prixProduit mediumint(5), PRIMARY KEY (idProduit), FOREIGN KEY (idCategorie) references Categories(idCategorie) );
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ 1 row inserted. (Query took 0.0135 seconds.)
INSERT INTO Categories (idCategorie, nomCategorie) VALUES ('100', 'Téléviseur');
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ 1 row inserted. (Query took 0.0133 seconds.)
INSERT INTO Categories VALUES ('200', 'Camescope');
[ Edit inline ] [ Edit ] [ Create PHP code ]

✓ 1 row inserted. (Query took 0.0229 seconds.)
INSERT INTO Categories(idCategorie, nomCategorie) VALUES ('300', 'Ordinateur');
```

## 3. Modifier les pages existantes

Nous commençons par créer le fichier connect.inc.php qui va permettre de faire la connection à la base de données :

```

<?php
    $host = "localhost";
    $name = 'myPws2038';
    $user = "root";
    $pass = "admin";
    $dsn = 'mysql:host='.$host.';dbname='.$name.';charset=utf8';

    try
    {
        $conn = new PDO($dsn, $user, $pass);
        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    } catch (PDOException $e) {echo $e->getMessage();}
?>

```

La base de donnée utilisé étant local, le nom d'utilisateur et le mot de passe ne sont pas ceux du serveur de l'IUT.

Nous allons modifier les fichiers ConsultPrix et ConsultCat pour intégrer les requêtes SQL. Voici le code de ConsultPrix :

```

$tmp = array();
$reqp = 'SELECT idProduit, idCategorie, nomProduit, prixProduit
        FROM Produits
        WHERE prixProduit >= 500;';
$reqm = 'SELECT idProduit, idCategorie, nomProduit, prixProduit
        FROM Produits
        WHERE prixProduit < 500;';
if ($prix == 1)
{
    $res = $conn->prepare($reqp);
    $res->execute();
}
else if ($prix == 0)
{
    $res = $conn->prepare($reqm);
    $res->execute();
}

foreach ($res->fetchAll(PDO::FETCH_ASSOC) as $elm)
{
    array_push($tmp, $elm);
}
echo '<BR><BR>';
include("Fonctions.php");
afficheTabHtml($tabProd, $tmp);

```

Nous pouvons donc voir les chaînes de caractères qui correspondent à la requête, le `conn->prepare` qui permet de préparer la requête et le `res->execute` qui permet d'exécuter la requête. On récupère ensuite le résultat grâce à `fetchAll` pour le parcourir et les afficher.

Le résultat reste le même qu'au tp4.

Voici le code de ConsultCat :

```
require_once("connect.inc.php");
$reqp = 'SELECT idCategorie, nomCategorie
        FROM Categories';
$res = $conn->prepare($reqp);
$res->execute();
$res->setFetchMode(PDO::FETCH_ASSOC);
$tabCat = $res->fetchAll();
```

On récupère toutes les catégories disponibles pour pouvoir créer le formulaire. La création du formulaire est la même qu'au tp4. On affiche ensuite le tableau suivant le résultat de la requête :

```
$tmp = array();
$req = 'SELECT * FROM Produits
        WHERE idCategorie = :id';
$res = $conn->prepare($req);
$res->execute(array('id' => $_POST['cat']));
foreach ($res->fetchAll(PDO::FETCH_ASSOC) as $elm)
{
    array_push($tmp, $elm);
}
echo '<br><br>';
afficheTabHtml($tmp, $tmp);
```

## 4. Conclusion

Dans ce TP, nous avons vu comment créer une base de donnée en mySQL avec un script SQL. Nous avons ensuite vu comment la relier à nos pages PHP pour pouvoir préparer et exécuter des requêtes et en utiliser les résultats.