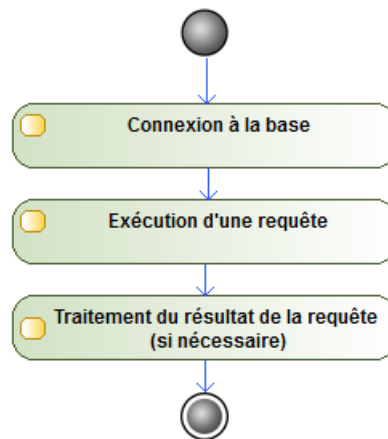


PHP – Accès à une base de données

L'accès à une base de données en PHP se fait en 3 temps :



Il existe différentes classes PHP permettant l'accès à une base de données. Nous utiliserons PDO qui est la plus générique. Référence : <https://www.php.net/manual/fr/book.pdo.php>

Nous utiliserons en guise d'exemple une base de données mysql dont le schéma est le suivant :

villes_villes_france_free	
ville_id	: mediumint(8) unsigned
ville_departement	: varchar(3)
ville_slug	: varchar(255)
ville_nom	: varchar(45)
ville_nom_simple	: varchar(45)
ville_nom_reel	: varchar(45)
ville_nom_soundex	: varchar(20)
ville_nom_metaphone	: varchar(22)
ville_code_postal	: varchar(255)
ville_commune	: varchar(3)
ville_code_commune	: varchar(5)
ville_arrondissement	: smallint(3) unsigned
ville_canton	: varchar(4)
ville_amdi	: smallint(5) unsigned
ville_population_2010	: mediumint(11) unsigned
ville_population_1999	: mediumint(11) unsigned
ville_population_2012	: mediumint(10) unsigned
ville_densite_2010	: int(11)
ville_surface	: float
ville_longitude_deg	: float
ville_latitude_deg	: float
ville_longitude_grd	: varchar(9)
ville_latitude_grd	: varchar(8)
ville_longitude_dms	: varchar(9)
ville_latitude_dms	: varchar(8)
ville_zmin	: mediumint(4)
ville_zmax	: mediumint(4)

Source des données : <https://sql.sh/736-base-donnees-villes-francaises>

1. Connexion à la base

Une bonne pratique est de stocker les paramètres de connexion à la base dans un fichier .php séparé.

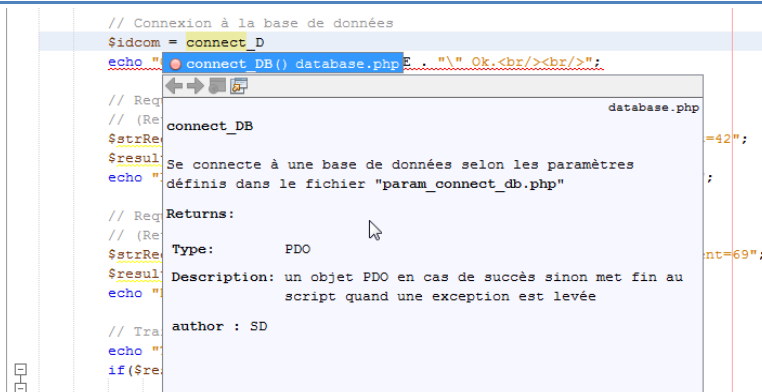
Fichier des paramètres de connexion à la base de données (param_connect_db.php) :

```
<?php
define("HOST","localhost");
define("USER","root");
define("PASS","");
define("BASE","villes");
?>
```

Une autre bonne pratique est de créer une fonction de connexion à la base :

Fichier des fonctions relatives à la base de données (database.php) :

```
<?php
/**
 * Se connecte à une base de données selon les paramètres
 * définis dans le fichier <b>"param_connect_db.php"</b>
 *
 * @author : SD
 * @return PDO un objet PDO en cas de succès sinon met fin au script quand
 * une exception est levée
 */
function connect_DB() {
    // Inclusion des paramètres de connexion
    include_once("param_connect_db.php");
    $dsn="mysql:host=" . HOST . ";dbname=" . BASE;
    try {
        // Tentative de connexion au serveur
        $idcom = new PDO($dsn,USER,PASS);
        // Pour les caractères accentués
        $idcom->exec("SET NAMES UTF8");
        // Renvoie l'objet en cas de réussite
        return $idcom;
    }
    catch(PDOException $except) {
        // Renvoie false en cas d'échec
        echo "Échec de la connexion",$except->getMessage();
        exit();
    }
}
```



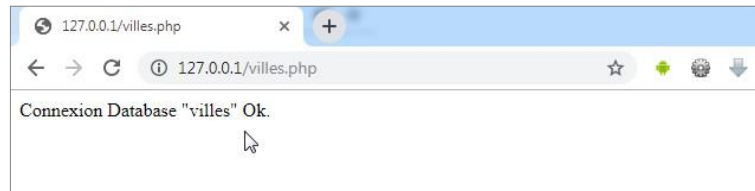
Programme de test :

```
<?php
    include_once("database.php");

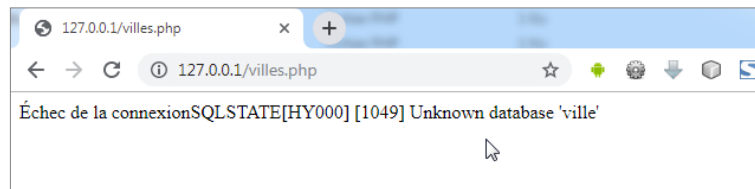
    // Connexion à la base de données
    $idcom = connect_DB();

    echo "Connexion Database \" . BASE . "\" Ok.";

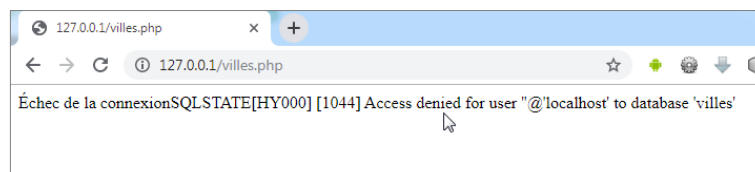
?>
```

Tests :

Connexion réussie



Échec de connexion à cause d'un mauvais nom de base de données



Échec de connexion à cause d'un mauvais nom d'utilisateur

2. Exécution d'une requête

Pour exécuter une requête nous disposons de deux méthodes :

- **exec** : Utilisé pour les requêtes qui ne renvoient pas une liste d'enregistrements comme par exemple UPDATE, DELETE, INSERT,... Retourne le nombre de lignes affectées.
- **query** : Utilisé pour les requêtes qui renvoient une liste d'enregistrements comme SELECT. Retourne un jeu de résultats en tant qu'objet PDOStatement

Programme de test :

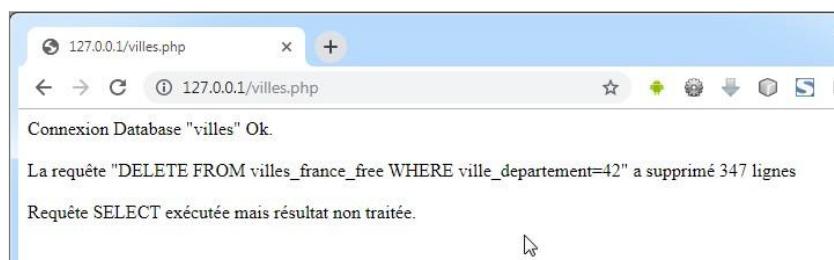
```
<?php
include_once("database.php");

// Connexion à la base de données
$idcom = connect_DB();
echo "Connexion Database \"\" . BASE . "\"" Ok.";

// Requête SQL sur la base choisie avec la méthode "exec"
// (Retourne le nombre de lignes affectées)
$strRequete = "DELETE FROM villes_france_free WHERE ville_departement=42";
$result = $idcom->exec($strRequete);
echo "La requête \"$strRequete\" a supprimé $result lignes<br/><br/>";

// Requête SQL sur la base choisie avec la méthode "query"
// (Retourne un jeu de résultats en tant qu'objet PDOStatement)
$strRequete = "SELECT * FROM villes_france_free WHERE ville_departement=69";
$result = $idcom->query($strRequete);
echo "Requête SELECT exécutée mais résultat non traitée.<br/><br/>";
?>
```

Tests :



3. Traitement des résultats

Il faut parcourir le jeu de résultats renvoyé par la méthode « query » avec la méthode « fetch ».

Programme de test :

```
<?php
    include_once("database.php");

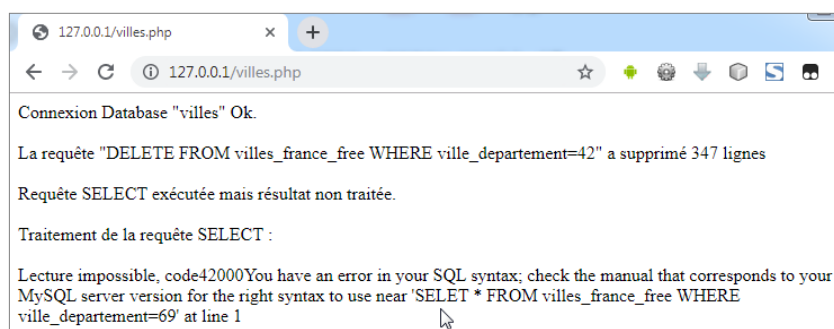
    // Connexion à la base de données
    $idcom = connect_DB();
    echo "Connexion Database \"\" . BASE . "\" Ok.";

    // Requête SQL sur la base choisie avec la méthode "exec"
    // (Retourne le nombre de lignes affectées)
    $strRequete = "DELETE FROM villes_france_free WHERE ville_departement=42";
    $result = $idcom->exec($strRequete);
    echo "La requête \"$strRequete\" a supprimé $result lignes<br/><br/>";

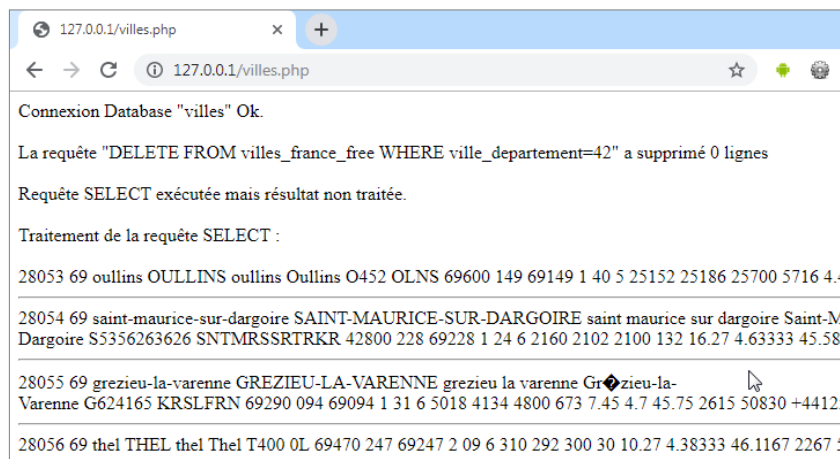
    // Requête SQL sur la base choisie avec la méthode "query"
    // (Retourne un jeu de résultats en tant qu'objet PDOStatement)
    $strRequete = "SELECT * FROM villes_france_free WHERE ville_departement=69";
    $result = $idcom->query($strRequete);
    echo "Requête SELECT exécutée mais résultat non traitée.<br/><br/>";

    // Traitement des résultats
    echo "Traitement de la requête SELECT :<br/><br/>";
    if($result) {
        while ($row = $result->fetch(PDO::FETCH_NUM)) {
            foreach($row as $donn) {
                echo $donn,"&nbsp;";
            }
            echo "<br/>";
        }
        $result->closeCursor();
    } else {
        $mes_erreur=$idcom->errorInfo();
        echo "Lecture impossible, code", $idcom->errorCode(),$mes_erreur[2];
    }
?>
```

Tests :



Avec une erreur de syntaxe dans la requête SELECT



Exécution sans erreur

4. Création de fonctions : afficheTable et afficheRequete

Ces fonctions seront placées dans le fichier « database.php ».

4.1 afficheRequete

Cette fonction utilise deux fonctions afficheEnTeteColonnes et afficheLigne détaillées dans la suite.

```
function afficheRequete($idcom, $requete) {
    $result=$idcom->query($requete);
    if($result) {
        $nbLignes=$result->rowCount();
        echo "Il y a $nbLignes enregistrements.";

        echo "<style>tr:nth-child(odd) {background-color:#FDE4D0;
        color:#984806;} tr:nth-child(even) {background-color:white;
        color:#984806;} </style><table>";

        if($ligne=$result->fetch(PDO::FETCH_ASSOC)) {
            // Affichage des noms des colonnes ou des alias
            afficheEnTeteColonnes(array_keys($ligne));
            // Affichage de la première ligne
            afficheLigne($ligne);
        }

        // Affichage des autres lignes
        while($ligne=$result->fetch(PDO::FETCH_NUM)){
            afficheLigne($ligne);
        }
        echo "</table>";

        $result->closeCursor();
    } else {
        $mes_erreur=$idcom->errorInfo();
        echo "Lecture impossible, code", $idcom->errorCode(),$mes_erreur[2];
    }
}
```

4.2 Fonctions *afficheLigne* et *afficheEnTeteColonnes*

```
function afficheLigne($ligne) {
    echo "<tr>";
    foreach($ligne as $valeur) {
        echo "<td> " . htmlentities($valeur) . " </td>";
    }
    echo "</tr>";
}

function afficheEnTeteColonnes($titres) {
    echo "<tr style=\"background-color:#F79646; color:white;\">";
    foreach($titres as $nomcol) {
        echo "<th>", htmlentities($nomcol) , "</th>";
    }
    echo "</tr>";
}
```

4.3 *afficheTable*

La fonction *afficheTable* appellera la fonction *afficheRequete* :

```
function afficheTable($idcom, $nomTable) {
    $requete = "SELECT * FROM " . $nomTable;
    afficheRequete($idcom, $requete);
}
```

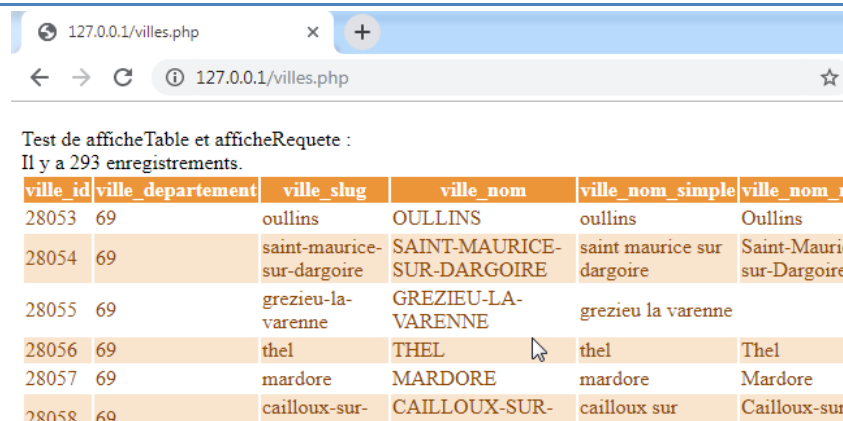
Programme de test et test

```
<?php
    include_once("database.php");

    // Connexion à la base de données
    $idcom = connect_DB();
    echo "Connexion Database \" . BASE . "\" Ok.<br/><br/>";

    echo "<br/><br/>Test de afficheTable et afficheRequete :<br/>";
    afficheRequete($idcom, "SELECT * FROM villes_france_free WHERE
ville_departement=69");

    //Fermeture de la connexion
    $idcom=NULL;
?>
```



Test de afficheTable et afficheRequete :
Il y a 293 enregistrements.

ville_id	ville_departement	ville_slug	ville_nom	ville_nom_simple	ville_nom_t
28053	69	oullins	OULLINS	oullins	Oullins
28054	69	saint-maurice-sur-dargoire	SAINT-AURICE-SUR-DARGOIRE	saint maurice sur dargoire	Saint-Mauri sur-Dargoire
28055	69	grezieu-la-varenne	GREZIEU-LA-VARENNE	grezieu la varenne	
28056	69	thel	THEL	thel	Thel
28057	69	mardore	MARDORE	mardore	Mardore
28058	69	cailloux-sur-	CAILLOUX-SUR-	cailloux sur	Cailloux-sur