OBJECTIFS : On se propose dans ce Tp :

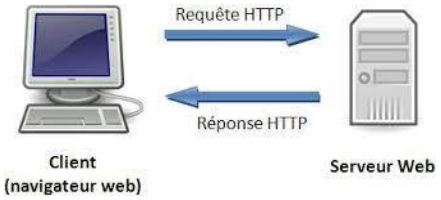
- De constituer et d'utiliser une base de données contenant des informations sur le championnat de ligue 1 de football.

- De créer une page web qui demande de saisir le numéro d'une journée de championnat et le nom d'une équipe et qui affiche en retour le résultat du match correspondant.

Cet objectif sera atteint en 4 parties :

- *Partie 1* : Constitution de la bdd en écrivant les requêtes Sql dans *phpMyAdmin* :
  - création des tables,
  - insertion « manuelle » d'un petit nombre d'enregistrements,
  - tests de l'efficacité du modèle relationnel utilisé.

- *Partie 2* : On recommence le travail de la partie 1 en utilisant à présent une communication http locale entre le navigateur web et le serveur sur lequel se trouve la bdd et les fichiers Php.



- **_Partie 3_ : On constitue le fichier Php qui permettra d'afficher la page web présentée ci-dessus. On teste l'ensemble en local (le serveur Web virtuel Apache MySQL Php crée avec le logiciel UwAmp se trouve sur votre ordinateur).**

- *Partie 4* : On télécharge son travail sur le serveur Apache d'un site clone de *nsibranly.fr* afin de tester l'ensemble sur un serveur distant. On en profite pour analyser les requêtes http échangées entre le navigateur sur votre ordinateur et le serveur distant (serveur Ovh sur la région Lilloise) .

1. Mise en route et objectifs :

⇨ Télécharger le dossier partie3.zip à partir de https://nsibranly.fr , le décompresser et copier les fichiers qu'il comprend, **directement dans le dossier UwAmp/UwAmp/www** avec les autres fichiers de la partie 2 précédente de ce travail. Ces fichiers sont :

- *ligue1.php* que l'on manipulera tout au long de ce tp,

- des fichiers images que l'on pourra éventuellement utiliser.

Le fichier *ligue1.php* est à finaliser. Comme toute cette partie liée au *php* et est ainsi en limite du programme de NSI, on ne peut pas vous demander d'y consacrer trop de temps. Ainsi, pour atteindre l'objectif final fixé dans ce travail sur le championnat de ligue 1, on part d'un fichier *php* qui n'est pas vierge.

⇨ Ouvrir le fichier *ligue1.php* avec Visual Studio Code. Ce fichier comprend :

- un formulaire avec comme action liée au clic sur le bouton, une seconde exécution du fichier *ligue1.php*. Ainsi en cliquant sur le bouton *Valider*, le script de ce fichier est à nouveau exécuté.
  Ce formulaire est composé :
  - d'une balise input avec name = ' *num_journee* '
  - d'une balise input avec name = *'ville'*
  - d'un bouton avec name = '*bouton*'

  Les valeurs saisies par l'utilisateur sont transmises cette fois-ci au serveur par la méthode POST qui transmet directement les données via le protocole http et non par l'url.
- un bloc qui contient le script php.
- un dernier bloc qui permet d'afficher quelque chose qui ressemble à :


```
<!DOCTYPE html>
<html lang="fr-FR">
    <head>
        <title>sql-essais</title>
        <meta charset="utf-8">
        <link rel="stylesheet" href="style.css">
    </head>

    <body>
        <form class="form" action="ligue1.php" method="POST">
            <div >
                <input type="text" name="num_journee" placeholder="n° de journée" autofocus>
            </div>
            <div >
                <input type="text" name="ville" placeholder="Ville de l'équipe" >
            </div>
            <div class="bouton">
                <input id="bouton" type="submit" value="Valider" name="bouton">
            </div>
        </form>
        <div>
            <?php
                $hote = "";
                $score_hote = "";
                $visiteur = "";
                $score_visiteur = "";
                $buts_hote = [];
                $buts_visiteur = [];
                if (isset($_POST["bouton"])){
                    include "sql_manager.php" ; // fichier contenant les fonctions connectDb() et affiche_tableau()
                    $connexion = connectDb("ligue1"); // connexion à la bdd
                    $num_journee = $_POST["num_journee"]; // récupération des données du formulaire
                    $ville = $_POST["ville"];
                    echo "Donnees transmises : num_journee = $num_journee ; ville = $ville";

                }
            ?>
        </div>
        <div class = "contenant">···
        </div>
    </body>
</html>
```

En indiquant action = 'ligue1.php', on spécifie que le fichier *ligue1.php* sera à nouveau exécuté lorsque le formulaire sera validé.

Ce bloc à l'intérieur **du if (isset($_POST[]) {}** est exécuté uniquement lorsque le formulaire est validé (*quand la variable $_POST[] qui est renvoyée lors de la validation, existe*)

FORMULAIRE

Bloc PHP

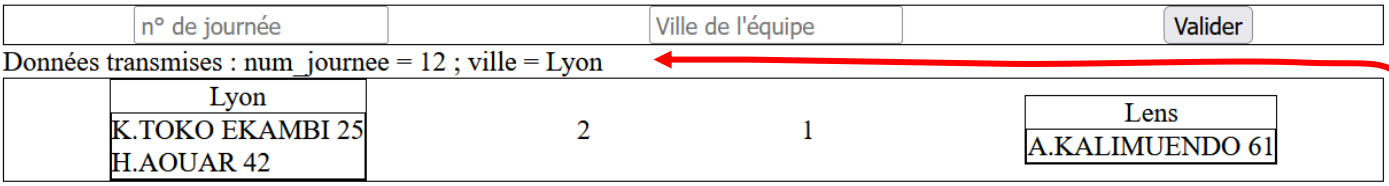
Bloc affichage résultat

⇨ Exécuter ce fichier *ligue1.php* ….. seul le formulaire est affiché sur la page web. La partie à l'intérieur du if *(isset($_POST[]) {}* n'est pas exécutée car la variable *$_POST['bouton']* n'existe pas au premier appel de *ligue1.php* .

⇨ Modifier ce fichier *ligue1.php* , en donnant une valeur aux 6 variables initialisées pour l'instant à des valeurs nulles et qui se trouve au début de la partie php. Pour avoir un résultat qui ressemble à l'objectif que l'on s'est fixé, on doit leur donner les valeurs suivantes :


```
$hote = "Lyon";
$score_hote = "2";
$visiteur = "Lens";
$score_visiteur = "1";
$buts_hote = [["K.TOKO EKAMBI","25"],["H.AOUAR","42"]];
$buts_visiteur = [["A.KALIMUENDO","61"]];
```

⇨ Recharger la page en ayant saisi ***12*** pour le 1er input et ***Lyon*** pour le 2nd. On obtient :



Les valeurs de ces 6 variables sont à présent affichées sur la page web via le script qui a déjà été écrit en fin du fichier de *ligue1.php* .

```php
<?php
    $hote = "Lyon";
    $score_hote = "2";
    $visiteur = "Lens";
    $score_visiteur = "1";
    $buts_hote = [["K.TOKO EKAMBI","25"],["H.AOUAR","42"]];
    $buts_visiteur = [["A.KALIMUENDO","61"]];
    if (isset($_POST["bouton"])){
        include "sql_manager.php" ; // fichier contenant les fonctions connectDb() et affiche_tableau()
        $connexion = connectDb("ligue1"); // connexion à la bdd
        $num_journee = $_POST["num_journee"];  // récupération des données du formulaire
        $ville = $_POST["ville"];
        echo "Données transmises : num_journee = $num_journee ; ville = $ville";
        PARTIE A COMPLETER
    }
?>
```

Ainsi, l'objectif de cette 3ième partie du tp, sera de compléter la partie *php* centrale de *ligue1.php*, afin d'interroger la Bdd pour pouvoir donner les bonnes valeurs aux variables *$hote, $score_hote, $visiteur, $score_visiteur* et aux listes *$buts_hote* et *$buts_visiteur* . Pour cela, le code reçoit de l'utilisateur, uniquement les informations qui sont dans *$num_journee* et *$ville*.

2. On interroge la Bdd pour connaître la ville qui joue contre celle saisie par l'utilisateur :

Cette requête est un peu particulière. On vous la donne donc ….

⇨ Ecrire la requête suivante dans le fichier ***index.php*** utilisé dans la partie 2 précédente de ce tp (BIEN TESTER le requêtes dans ce fichier *index.php* et non pas dans **ligue1.php**) :

```
SELECT eh.ville AS hote , ev.ville AS visiteur FROM matchs AS m
JOIN equipes AS eh ON m.id_equipes_hote = eh.id
JOIN equipes AS ev ON m.id_equipes_visiteur = ev.id
WHERE m.num_journee = 1 AND (eh.ville = 'Lyon' OR ev.ville = 'Lyon')
```

La page renvoyée est :

hote
Lyon

visiteur
Brest

Sur le fichier *index.php* toujours, enlever les // de commentaire sur la fonctior `//affiche_tableau($resultat);`

et recharger la page.

On obtient à présent en plus, l'affichage du tableau (ou liste) nommé *$resultat* et qui est retourné lorsque la requête Sql est exécutée.

```
Array
(
    [0] => Array
        (
            [hote] => Lyon
            [visiteur] => Brest
        )

)
```

$resultat est un tableau indexé qui ne comprend qu'un élément *$resultat[0]*

Cet élément *$resultat[0]* est un tableau associatif (dictionnaire) qui comporte 2 valeurs. Les clés de ces valeurs sont *'hote'* et *'visiteur'*

On voit ainsi que pour récupérer les résultats de la requête et les stocker dans les variables *$hote* et *$visiteur*, on doit exécuter dans le code php :

```
$hote = $resultat[0]["hote"];
$visiteur = $resultat[0]["visiteur"];
```

Cela donne finalement :

```
if (isset($_POST["bouton"])){
    include "sql_manager.php" ; // fichier contenant les fonctions connectDb() et affiche_tableau(
    $connexion = connectDb("ligue1"); // connexion à la bdd
    $num_journee = $_POST["num_journee"];  // récupération des données du formulaire
    $ville = $_POST["ville"];
    // PREMIERE REQUETE SQL
    try {
        $ligne_sql =
            "
            SELECT eh.ville AS hote , ev.ville AS visiteur FROM matchs AS m
            JOIN equipes AS eh ON m.id_equipes_hote = eh.id
            JOIN equipes AS ev ON m.id_equipes_visiteur = ev.id
            WHERE m.num_journee = $num_journee AND (eh.ville = '$ville' OR ev.ville = '$ville')

            ";
        $requete = $connexion->prepare($ligne_sql);
        $requete->execute();
        $resultat = $requete->fetchAll(PDO::FETCH_ASSOC);
        $hote = $resultat[0]["hote"];
        $visiteur = $resultat[0]["visiteur"];
    }catch(PDOException $e){
        die('Erreur : ' . $e->getMessage());
    }

}
```

On met ici les variables qui stockent le numéro de la journée et la ville saisis par

Les variables *$hote* et *$visiteur* sont actualisées par rapport à celles définies à l'initialisation

⇨ Réinitialiser modifiées avant à les variables à des valeurs nulles :

```
$hote = "";
$score_hote = "";
$visiteur = "";
$score_visiteur = "";
$buts_hote = [];
$buts_visiteur = [];
```

⇨ Compléter le fichier *ligue1.php* avec le script ci-dessus et tester le bon fonctionnement en saisissant des noms de villes et numéros de journée différents. Vérifier sur le web que l'équipe adverse est la bonne.

3. On complète le fichier ligue1.php pour connaitre la liste des buteurs de l'équipe hôte

On connait à présent la ville de l'équipe hôte qui est dans *$hote*.

⇨ Compléter la partie php du fichier *ligue1.php* afin de pouvoir mettre les bonnes valeurs dans *$score_hote* qui *$buts_hote* qui est une liste de liste. On donne le script à trou qui suit, qu'il s'agit de reprendre pour compléter *ligue1.php*.

Bien tester au préalable la requête Sql sur le fichier *index.php* afin de vérifier son résultat et aussi afin d'analyser la structure du résultat retourné.

```
try {
    $ligne_sql =
            "
                
                
                
                
                ";
        $requete = $connexion->prepare($ligne_sql);
        $requete->execute();
        $resultat = $requete->fetchAll(PDO::FETCH_ASSOC);
        $score_hote = count($resultat);
        foreach ($resultat as $elt){
            $buteur = $elt["nom"];
            $temps = $elt["temps"];
            array_push($buts_hote,[$buteur , $temps]) ;
        }
    }catch(PDOException $e){
        die('Erreur : ' . $e->getMessage());
    }
```

count( ) en php est l'équivalent de len( ) en python. C'est une fonction qui retourne la taille de la liste en argument

array_push( ) en php serait en python : $buts_hote.append([$buteur , $temps])

⇨ Tester le bon fonctionnement.

4. On complète le fichier ligue1.php pour connaitre la liste des buteurs de l'équipe visiteur

⇨ Compléter de la même façon le script php du fichier ligue1.php afin de pouvoir mettre les bonnes valeurs dans *$score_visiteur* qui *$buts_visiteur* qui est une liste de liste.

⇨ Tester le bon fonctionnement.

5. Finitions

⇨ Améliorer l'ensemble afin d'obtenir un meilleur rendu ou davantage d'informations intéressantes.

Cette troisième partie est terminée. Ses objectifs étaient :

- de toujours pratiquer le Sql en tapant au clavier des lignes de commande d'insertion, de mise à jour et de sélection,
- d'instaurer un dialogue entre l'utilisateur (Client) et le serveur en utilisant une méthode POST
- de réaliser un script php qui exécute plusieurs requêtes sur la Bdd dans l'objectif d'afficher sur la page web visitée les données demandées par l'internaute.

On s'oriente à présent vers la quatrième partie qui consiste à mettre en ligne son travail sur un serveur distant et à analyser le procédé utilisé pour réaliser le dialogue Client – Serveur.