



TP 6 Web Dynamique

Accès aux bases de données

L'objectif de ce TP est d'interfacer notre application PHP avec une base de données. Pour cela, nous reprenons notre application développée dans le TP précédent (TP5) pour la modifier et la compléter.

Votre tâche dans ce TP consiste à stocker les données de l'application dans une Base de données MySQL (au lieu de tableaux PHP) et de modifier les scripts pour pouvoir chercher, insérer, modifier et supprimer les données directement.

I- Créer la base de données et les tables

- 1- Copier votre travail précédent dans un nouveau répertoire nommé TP6 (vous pouvez télécharger la solution du TP5, pour l'utiliser comme point de départ).
- 2- Utiliser l'outil PHP MyAdmin pour créer une nouvelle Base de données MySQL nommée SMI2020.
- 3- Créer deux tables dans cette base de données (la table **Filiere** (*CodeF varchar(10) primary Key, IntituleF varchar(50) Not Null*) dans laquelle on va stocker les filières et la table **Etudiant** (*CodeE varchar(10) primary Key, Nom varchar (30), Prenom varchar (30), Filiere varchar (10), Note décimal*) dans laquelle on va stocker les étudiants).
- 4- remplir les tables avec des exemples d'enregistrements.

II- Modifier le code de l'application

Les données de notre application sont maintenant dans la base de données (dans les deux tables **Filiere** et **Etudiant**). Pour pouvoir afficher correctement, il faut faire les modifications nécessaires. Par exemple, pour afficher la page **accueil.php**, les données ne sont plus dans le tableau PHP **\$filiere**. La même chose pour les étudiants.

- 1- Modifier le fichier **data.php**.

a- écrire une fonction **ouvrirConnexion()** qui retourne une connexion à la base de données (utiliser PDO).

b- Ecrire une fonction **getListeFiliere()** qui permet de retourner la liste de toutes les filières (elle retourne l'équivalent de l'ancien tableau **\$filiere**).

- c- Modifier l'ancienne fonction ***getListeEtudiants(\$filiere)*** de façon à rechercher ces données à partir de la base de données (le tableau global ***\$etudiants*** n'existe plus).
- d- Modifier la fonction ***getDetailEtudiant(\$code)*** de sorte à ce qu'elle cherche les détails d'un étudiant à partir de la Base de données.
- e- Écrire une fonction ***AjouterEtudiant(\$t)*** qui accepte un tableau en paramètre et insère un enregistrement dans la table ***Etudiant***. (Le tableau ***\$t*** est censé contenir le ***code***, le ***nom***, le ***prénom***, la ***filière*** et la ***note*** d'un étudiant).
- f- Écrire une fonction ***SupprimerEtudiant(\$c)*** qui accepte le code d'un étudiant en paramètre et supprime l'étudiant correspondant.

2- Modifier le script ***accueil.php***

Pour que ce script fonctionne, vous devez affecter à la variable ***\$filiere*** le résultat de retour de la fonction ***getListeFiliere***. Donc, juste après la ligne :

```
include ('data.php');
```

mettez:

```
$filiere=getListeFiliere();
```

3- Faites de sorte que le script ***accueil.php*** affiche les étudiants de la filière sélectionnée (si vous utilisez le script fourni dans la solution du TP5, le script ne nécessite aucune modification, car la modification est déjà faite au niveau de la fonction ***getListeEtudiants*** du fichier ***data.php***).

4- La même chose pour le script ***detail.php*** (remplacez seulement la ligne 12 par:

```
$etudiant= getDetailEtudiant($c);
```

5- Écrire un script ***supprimer.php*** qui permet de supprimer un étudiant dont le code est passé par la méthode ***GET***. Le script utilise pour cela la fonction ***supprimerEtudiant()*** créée plus haut. Ensuite, l'utilisateur est redirigé vers la liste des étudiants.

6- Pour ajouter un étudiant, créer un script ***insert.php*** qui permet de recevoir les données envoyées par le formulaire ***formAjout.php*** et les stocker dans la base de données. L'utilisateur est redirigé ensuite vers la liste des étudiants. (Le script ***receptionPost.php*** sera donc annulé).