

Alimenter une ou plusieurs tables mySQL

Dans cet exercice nous allons voir comment alimenter une ou plusieurs tables avec les données qui proviennent d'un même formulaire.

Alimenter une table

Pour commencer vous allez créer la table **infos_tbl** dans phpMyAdmin comme suit :

requête SQL:

```
CREATE TABLE infos_tbl (id INT (11) not null AUTO_INCREMENT, nom VARCHAR (35) not null , prenom VARCHAR (35) not null , email VARCHAR (70) not null , icq INT (11) null , titre VARCHAR (70) not null , url VARCHAR (255) not null , PRIMARY KEY (id), INDEX (id), UNIQUE (id))
```

Ensuite nous allons utiliser le formulaire ci dessous qui va alimenter la table :

Code HTML	Donne à l'écran
<pre><html> <form method="POST" action="add.php"> <center> <input type="text" name="nom" size="20" value="nom" maxlength="35"> <input type="text" name="prenom" size="20" value="prenom" maxlength="35">
 <input type="text" name="email" size="20" value="email" maxlength="70"> <input type="text" name="icq" size="20" value="icq" maxlength="11">
 <input type="text" name="titre" size="20" value="titre du site" maxlength="70"> <input type="text" name="url" size="20" value="url du site" maxlength="255">
 <input type="submit" value="Envoyer" name="envoyer"> </center> </form> </html></pre>	

Vous noterez les "**maxlength**" dans chacun des champs, ceci permet de brider le nombre de caractères maximum que le visiteur peut entrer dans le champ, bien sûr ici le "maxlength" correspond au nombre de caractères spécifié dans la création de la table **infos_tbl**. Cela a un intérêt, celui d'être sûr que le visiteur ne tapera pas plus de caractères que prévu.

Voyons maintenant le script PHP en lui-même, celui-ci sera contenu dans le fichier **add.php** auquel fait référence le **POST** du formulaire :

Explication :

La requête INSERT INTO permet donc l'insertion des champs du formulaire dans la table. Dans notre cas le premier champ reste vide car il s'agit de l'id (identifiant) qui s'incrémente automatiquement à chaque nouvelle requête INSERT.

Notez que dans le cas ou vous ne voudriez insérer que les champs Nom et Prénom dans la table vous pouvez très bien spécifier dans la requête le nom de chaque champ, comme suit :

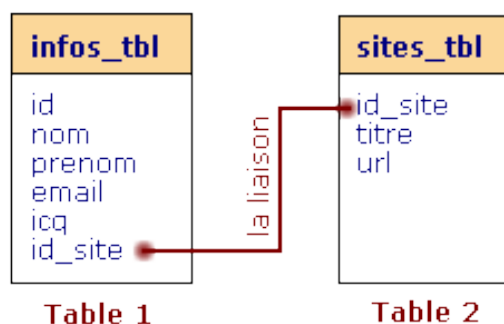
```
$sql = "INSERT INTO infos_tbl(nom,prenom) VALUES(' $nom' , ' $prenom' ) " ;
```

Mais attention dans ce cas les autres champs de la table devront avoir l'attribut **NULL** et non pas **NOT NULL**, Null indique au champ qu'il pourra rester vide. Et si vous essayez d'insérer un nombre de valeurs

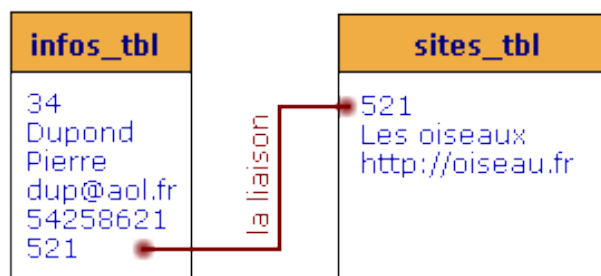
Bien sûr, l'idéal est de tester si l'URL existe dans la table pour éviter les doublons, si celle-ci existe déjà on indique au visiteur qu'il ne peut pas valider son formulaire. J'ai pris l'exemple ici de l'URL mais il va sans dire que vous pouvez vérifier les champs de votre choix, cela dépend des doublons que vous ne souhaitez pas avoir dans votre table. Voici le code :

Alimenter deux tables et créer une liaison

Voyons maintenant comment ces mêmes informations peuvent être enregistrées dans deux tables différentes en gardant une liaison entre les deux. Le principe est en fait tout bête, admettons que nous ayons une première table qui va nous servir à stocker les coordonnées du visiteur (`$nom` , `$prenom`, `$email`, `$icq`) et une seconde dans laquelle ne seront sauvegardées que les informations du site (`$titre` et `$url`). Voyez ce petit schéma qui va vous éclairer sur la manière de créer une liaison entre 2 tables :



Un exemple concret :



Examinons la méthode à employer :

1. On commence par insérer `$titre` et `$url` dans la table 2 (**sites_tbl**).
2. Une fois l'insertion effectuée, on utilise la fonction PHP `mysql_insert_id()` pour connaître l'`id_site` qui a été affecté à notre nouvel enregistrement.
3. On insère le reste du formulaire dans la table 1 (**infos_tbl**), soit : `$nom` , `$prenom`, `$email`, `$icq` et `$id_site`.

Nous avons maintenant une liaison entre la table 1 et 2 via l'`id_site`. Ci-dessous le code PHP de cette manipulation :

Important : L'`id_site` de la table 2 (**sites_tbl**) doit être en **auto_incrément automatique** au même titre que l'`id` de la table 1 (**infos_tbl**). Je ne vous donne "volontairement" pas la requête SQL pour créer les deux tables, et ce pour vous faire travailler un peu tout de même ;).

Conclusion :

Vous pouvez créer autant de liaisons que vous le voulez entre vos tables, cela a pour but de les alléger ce qui permet une plus grande rapidité lorsque vous souhaitez les interroger.

