

Requête PHP MySQL SELECT

Dans ce chapitre, vous apprendrez à sélectionner des enregistrements dans une table MySQL à l'aide de PHP.

Sélection de données à partir de tables de base de données

Jusqu'à présent, vous avez appris à créer une base de données et une table ainsi qu'à insérer des données. Il est maintenant temps de récupérer les données que vous avez insérées dans le chapitre précédent. L'instruction **SQL SELECT** est utilisée pour sélectionner les enregistrements dans les tables de la base de données. Sa syntaxe de base est la suivante :

```
SELECT column1_name, column2_name, columnN_name FROM table_name;
```

Faisons une requête SQL en utilisant l'instruction **SELECT**, après quoi nous exécuterons cette requête SQL en la passant à la fonction PHP **mysqli_query()** pour récupérer les données de la table.

Considérez que notre table de base de données de personnes contient les enregistrements suivants :

+---+-----+-----+-----+			
id	first_name	last_name	email
+---+-----+-----+-----+			
1	Peter	Parker	peterparker@mail.com
2	John	Rambo	johnrambo@mail.com
3	Clark	Kent	clarkkent@mail.com
4	John	Carter	johncarter@mail.com
5	Harry	Potter	harrypotter@mail.com
+---+-----+-----+-----+			

Le code PHP dans l'exemple suivant sélectionne toutes les données stockées dans la table persons (l'utilisation du caractère astérisque (*) à la place du nom de la colonne sélectionne toutes les données de la table).

Code un peu long (le voir directement sur vsCode)

nom du fichier : mysql-select-query.php

Explication du code

Dans l'exemple ci-dessus, les données renvoyées par la fonction `mysqli_query()` sont stockées dans la variable `$result`. Chaque fois que `mysqli_fetch_array()` est invoquée, elle renvoie la ligne suivante de l'ensemble de résultats sous forme de tableau. La boucle `while` est utilisée pour parcourir en boucle toutes les lignes de l'ensemble de résultats. Enfin, il est possible d'accéder à la valeur d'un champ individuel à partir de la ligne en passant l'index du champ ou le nom du champ à la variable `$row` comme `$row['id']` ou `$row[0]`, `$row['first_name']` ou `$row[1]`, `$row['last_name']` ou `$row[2]`, et `$row['email']` ou `$row[3]`.

Si vous souhaitez utiliser la boucle `for`, vous pouvez obtenir la valeur du compteur de boucle ou le nombre de lignes retournées par la requête en passant la variable `$result` à la fonction `mysqli_num_rows()`. Cette valeur du compteur de boucle détermine combien de fois la boucle doit être exécutée.