

# PHP MySQL Insérer des données

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

# Insertion de données dans une table de base de données MySQL

Maintenant que vous avez compris comment créer une base de données et des tables dans MySQL. Dans ce chapitre, vous apprendrez à exécuter une requête SQL pour insérer des enregistrements dans une table.

L'instruction **INSERT INTO** est utilisée pour insérer de nouvelles lignes dans une table de base de données.

Faisons une requête SQL en utilisant l'instruction **INSERT INTO** avec les valeurs appropriées, après quoi nous exécuterons cette requête d'insertion en la passant à la fonction PHP **mysqli\_query()** pour insérer des données dans la table. Voici un exemple, qui insère une nouvelle ligne dans la table **persons** en spécifiant des valeurs pour les champs `first_name` , `last_name` et `email` .

```
<?php
$link = mysqli_connect("localhost", "root", "", "demo");

// Vérifiez connexion
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}
$sql = "INSERT INTO persons (id, first_name, last_name, email)
        VALUES (1, 'Peter', 'Parker', 'peterparker@mail.com)";
if(mysqli_query($link, $sql)){
    echo "Records inserted successfully.";
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}
mysqli_close($link);
?>
```

Si vous vous souvenez du chapitre précédent , le champ id était marqué avec AUTO\_INCREMENT. Ce modificateur indique à MySQL d'attribuer automatiquement une valeur à ce champ s'il n'est pas spécifié, en incrémentant la valeur précédente de 1.

# Insertion de plusieurs lignes dans un tableau

Vous pouvez également insérer plusieurs lignes dans une table avec une seule requête d'insertion à la fois. Pour ce faire, incluez plusieurs listes de valeurs de colonne dans l'instruction **INSERT INTO**, où les valeurs de colonne de chaque ligne doivent être placées entre parenthèses et séparées par une virgule.

Insérons quelques lignes supplémentaires dans la table **persons** , comme ceci :

```
<?php
$link = mysqli_connect("localhost", "root", "", "demo");
if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}
$sql = "INSERT INTO persons (first_name, last_name, email) VALUES
        ('John', 'Rambo', 'johnrambo@mail.com'),
        ('Clark', 'Kent', 'clarkkent@mail.com'),
        ('John', 'Carter', 'johncarter@mail.com'),
        ('Harry', 'Potter', 'harrypotter@mail.com')";
if(mysqli_query($link, $sql)){
    echo "Records added successfully.";
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}
mysqli_close($link);
?>
```

Maintenant, allez dans phpMyAdmin (<http://localhost/phpmyadmin/>) et consultez les données de la table **persons** dans la base de données demo. Vous constaterez que la valeur de la colonne id est attribuée automatiquement en incrémentant la valeur de l'id précédent de **1**.

# Insérer des données dans une base de données à partir d'un formulaire HTML

Dans la section précédente, nous avons appris à insérer des données dans la base de données à partir d'un script PHP. Maintenant, nous allons voir comment insérer des données dans la base de données obtenues à partir d'un formulaire HTML. Créons un formulaire HTML qui peut être utilisé pour insérer de nouveaux enregistrements dans la table **persons**.

# Étape 1 : Création du formulaire HTML

Voici un formulaire HTML simple qui comporte trois champs <input> de texte et un bouton d'envoi.

```
<form action="insert.php" method="post">
  <p>
    <label for="firstName">Prénom:</label>
    <input type="text" name="first_name" id="firstName">
  </p>
  <p>
    <label for="lastName">Nom de famille:</label>
    <input type="text" name="last_name" id="lastName">
  </p>
  <p>
    <label for="emailAddress">Email:</label>
    <input type="text" name="email" id="emailAddress">
  </p>
  <input type="submit" value="Envoyer">
</form>
```

# Étape 2 : Récupérer et insérer les données du formulaire

Lorsqu'un utilisateur clique sur le bouton d'envoi du formulaire HTML d'ajout d'enregistrement, dans l'exemple ci-dessus, les données du formulaire sont envoyées au fichier 'insert.php'. Le fichier 'insert.php' se connecte au serveur de base de données MySQL, récupère les champs des formulaires à l'aide des variables PHP **\$\_REQUEST** et exécute enfin la requête d'insertion pour ajouter les enregistrements. Voici le code complet de notre fichier 'insert.php'



```
<?php
$link = mysqli_connect("localhost", "root", "", "demo");

if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}

$first_name = mysqli_real_escape_string($link, $_REQUEST['first_name']);
$last_name = mysqli_real_escape_string($link, $_REQUEST['last_name']);
$email = mysqli_real_escape_string($link, $_REQUEST['email']);

$sql = "INSERT INTO persons (first_name, last_name, email) VALUES
('$first_name', '$last_name', '$email')";
if(mysqli_query($link, $sql)){
    echo "Records added successfully.";
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}

mysqli_close($link);
?>
```

**Remarque :** La fonction `mysqli_real_escape_string()` échappe les caractères spéciaux dans une chaîne et crée une chaîne SQL légale pour assurer la sécurité contre l'injection SQL .

Ceci est un exemple très basique d'insertion des données de formulaire dans une table de base de données MySQL. Vous pouvez étendre cet exemple et le rendre plus interactif en ajoutant des validations aux entrées utilisateur avant de l'insérer dans les tables de la base de données. Veuillez consulter le chapitre sur la [validation de formulaire PHP](#) pour en savoir plus sur le nettoyage et la validation des entrées utilisateur à l'aide de PHP.