

Development Web : Back-End

MySql/PHP



Mohammed KABBAJ



Les requêtes SQL

Création de la base de données

La requête SQL "CREATE DATABASE" est utilisée pour créer une nouvelle base de données dans le système de gestion de base de données. La syntaxe de base est la suivante:

```
CREATE DATABASE nom_de_base_de_donnees;
```



Les requêtes SQL

Création de la table

La requête "CREATE TABLE" est utilisée pour créer une nouvelle table dans une base de données SQL. La syntaxe standard de cette requête est la suivante :

```
CREATE TABLE nom_de_table (  
    nom_colonne1 type_de_données(taille),  
    nom_colonne2 type_de_données(taille),  
    ...  
    nom_colonne_n type_de_données(taille),  
    PRIMARY KEY (nom_colonne1)  
);
```



Les requêtes SQL

Insertion dans la table

La requête "INSERT" permet d'ajouter des données à une table existante dans une base de données SQL. La syntaxe générale de cette requête est la suivante:

```
INSERT INTO nom_table (colonne1, colonne2, colonne3, ...)  
VALUES (valeur1, valeur2, valeur3, ...);
```



Les requêtes SQL

Afficher les données

La requête "SELECT" est utilisée pour sélectionner des données d'une table dans une base de données SQL. La syntaxe de base de la requête SELECT est la suivante:

```
SELECT colonne1, colonne2, ...  
FROM nom_de_table;
```



Les requêtes SQL

Modifier les données

La requête "UPDATE" permet de mettre à jour les données dans une table en spécifiant la table, les colonnes à mettre à jour et les nouvelles valeurs. La syntaxe générale est la suivante :

```
UPDATE table_name SET column1 = value1, column2 = value2, ...  
WHERE some_column = some_value;
```



Les requêtes SQL

Supprimer les données

La requête DELETE est utilisée pour supprimer des enregistrements d'une table dans une base de données. Son syntaxe est :

```
DELETE FROM nom_table  
WHERE some_column = some_value;
```



Les Privilèges mySql

Les utilisateurs dans MySQL permettent de contrôler l'accès à une base de données ou à une table spécifique. Chaque utilisateur peut avoir des autorisations différentes, telles que la lecture, l'écriture, la modification et la suppression.



Les Privilèges mySql

Les étapes

Pour créer un utilisateur dans phpMyAdmin, vous devez vous connecter à votre compte administrateur et suivre ces étapes:

1. Sélectionnez la base de données .
2. Cliquez sur le bouton "Privilèges" en haut.
3. Cliquez sur "Ajoutez un compte d'utilisateur".
4. Saisissez le "Nom d'utilisateur" et "Le mot de pass".
5. Cochez les case des privilèges selon le besoin
6. Cliquez sur le bouton "Exécuter" pour enregistrer les modifications.



Connexion MySql <=> PHP

L existe trois méthodes couramment utilisées pour établir une connexion entre PHP et MySQL :

1. MySQLi : cette méthode a été introduite dans PHP 5 et offre une interface orientée objet pour la connexion et l'interaction avec MySQL.
2. PDO (PHP Data Objects) : PDO est une couche d'abstraction de données qui permet une connexion à différents types de bases de données, y compris MySQL. Il offre une interface orientée objet similaire à MySQLi.
3. MySQL Extension : Cette méthode est la plus ancienne et n'est plus recommandée pour de nouveaux développements. Il utilise une interface de fonction traditionnelle pour la connexion et l'interaction avec MySQL.



Connexion MySql <=> PHP

Se connecter à une base de données en PDO

```
<?php
try {
    $pdo = new PDO("mysql:host=hostname;dbname=database_name", "username", "password");
}
catch (PDOException $e) {
    echo "Error : " . $e->getMessage();
    die();
}
?>
```



Connexion MySql <=> PHP

Ajouter à une table en PDO

```
<?php
try {
    // Établissement de la connexion à la base de données
    $conn = new PDO("mysql:host=hostname;dbname=database_name", "username", "password");
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // Préparation de la requête d'insertion
    $stmt = $conn->prepare("INSERT INTO nom_de_la_table (col1, col2, col3, ...) VALUES (:val1, :val2, :val3, ...)");

    // Liaison des valeurs
    $stmt->bindParam(':val1', $value1);
    $stmt->bindParam(':val2', $value2);
    $stmt->bindParam(':val3', $value3);
    ...

    // Exécution de la requête
    $stmt->execute();

    // Affichage d'un message de confirmation
    echo "Données insérées avec succès.";

} catch(PDOException $e) {
    // Affichage de l'erreur
    echo "Erreur : " . $e->getMessage();
}
?>
```



Connexion MySql <=> PHP

Modifier une ligne dans table en PDO

```
<?php
$pdo = new PDO("mysql:host=localhost;dbname=database_name", "username", "password");
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

$requete = "UPDATE nom_de_la_table SET champ1 = :valeur1, champ2 = :valeur2 WHERE champ3 = :valeur3";
$stmt = $pdo->prepare($requete);
$stmt->bindValue(':valeur1', $valeur1);
$stmt->bindValue(':valeur2', $valeur2);
$stmt->bindValue(':valeur3', $valeur3);

$stmt->execute();

?>
```



Connexion MySql <=> PHP

Supprimer une ligne dans table en PDO

```
<?php
$pdo = new PDO("mysql:host=localhost;dbname=database_name", "username", "password");
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$sql = "DELETE FROM nom_de_table WHERE condition";
$stmt = $pdo->prepare($sql);
$stmt->execute();
?>
```



Connexion MySql <=> PHP

Afficher des données en PDO

```
<?php
$pdo = new PDO("mysql:host=localhost;dbname=database_name", "username",
"password");
$pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$sql = "SELECT * FROM table_name";
$stmt = $pdo->prepare($sql);
$stmt->execute();
$result = $stmt->fetchAll();
foreach($result as $row) {
    echo $row['column_name'] . " ";
}
?>
```

