

# PHP et MySQL : Connexion, requêtes et gestion des bases de données

## Introduction

PHP est un langage de script côté serveur qui s'intègre parfaitement avec MySQL, un système de gestion de bases de données relationnelles. L'association PHP/MySQL permet de créer des applications web dynamiques capables de stocker, récupérer, afficher et manipuler des données en temps réel. Ce chapitre vous guide dans l'utilisation de PHP pour interagir avec une base de données MySQL : connexion, exécution de requêtes, affichage et gestion des résultats.

---

## 1. Présentation de MySQL

### 1.1 Qu'est-ce que MySQL ?

MySQL est un système de gestion de bases de données relationnelles (SGBDR) open source très populaire. Il permet de gérer des bases de données structurées sous forme de tables, lignes et colonnes.

### 1.2 Structure de base

- **Base de données** : conteneur de tables.
  - **Table** : ensemble de lignes contenant des données organisées en colonnes.
  - **Champ** : colonne d'une table.
  - **Enregistrement** : ligne dans une table.
- 

## 2. Connexion à une base de données MySQL en PHP

### 2.1 Utilisation de `mysqli`

```
<?php
$host = "localhost";
$user = "root";
$password = "";
$dbname = "ma_base";

$conn = new mysqli($host, $user, $password, $dbname);

if ($conn->connect_error) {
    die("Connexion échouée : " . $conn->connect_error);
}
```

```
echo "Connexion réussie !";  
?>
```

## 2.2 Utilisation de PDO (PHP Data Objects)

```
<?php  
try {  
    $conn = new PDO("mysql:host=localhost;dbname=ma_base", "root", "");  
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
    echo "Connexion réussie avec PDO !";  
} catch(PDOException $e) {  
    echo "Erreur : " . $e->getMessage();  
}  
?>
```

---

## 3. Exécution de requêtes SQL

### 3.1 Requête SELECT (lecture)

```
$sql = "SELECT * FROM utilisateurs";  
$result = $conn->query($sql);  
  
if ($result->num_rows > 0) {  
    while ($row = $result->fetch_assoc()) {  
        echo "Nom : " . $row["nom"] . " - Email : " . $row["email"] . "<br>";  
    }  
} else {  
    echo "0 résultats";  
}
```

### 3.2 Requête INSERT (ajout)

```
$sql = "INSERT INTO utilisateurs (nom, email) VALUES ('Alice',  
'alice@example.com')";  
  
if ($conn->query($sql) === TRUE) {  
    echo "Nouvel enregistrement ajouté avec succès";  
} else {  
    echo "Erreur : " . $conn->error;  
}
```

---

## 4. Utilisation sécurisée : Requêtes préparées

Pour éviter les injections SQL, il est essentiel d'utiliser des requêtes préparées :

```
$stmt = $conn->prepare("INSERT INTO utilisateurs (nom, email) VALUES (?, ?)");  
$stmt->bind_param("ss", $nom, $email);  
  
$nom = "Bob";
```

```
$email = "bob@example.com";  
$stmt->execute();
```

---

## 5. Requêtes de modification et suppression

### 5.1 Requête UPDATE (modification)

```
$sql = "UPDATE utilisateurs SET email='nouveau@example.com' WHERE  
nom='Alice'";  
$conn->query($sql);
```

### 5.2 Requête DELETE (suppression)

```
$sql = "DELETE FROM utilisateurs WHERE nom='Bob'";  
$conn->query($sql);
```

---

## 6. Affichage des données dans une page HTML

```
$sql = "SELECT * FROM utilisateurs";  
$result = $conn->query($sql);  
  
echo "<table border='1'>";  
echo "<tr><th>Nom</th><th>Email</th></tr>";  
  
while ($row = $result->fetch_assoc()) {  
    echo "<tr><td>".$row['nom']. "</td><td>".$row['email']. "</td></tr>";  
}  
echo "</table>";
```

---

## 7. Gestion des erreurs et messages

Il est important de vérifier systématiquement si une requête a réussi ou échoué, et d'afficher des messages clairs :

```
if ($conn->query($sql) === TRUE) {  
    echo "Opération réussie.";  
} else {  
    echo "Erreur : " . $conn->error;  
}
```

---

## 8. Déconnexion de la base de données

```
$conn->close();
```

---

## Conclusion

L'intégration entre PHP et MySQL constitue un socle fondamental pour les applications web dynamiques. Grâce aux connexions sécurisées, aux requêtes paramétrées, et à la gestion rigoureuse des résultats, vous pouvez créer des systèmes robustes tels que des CMS, des forums ou des systèmes d'authentification. La maîtrise de ces techniques est essentielle pour tout développeur back-end.